

Computational modeling to design and analyze synthetic metabolic circuits

Mathilde Koch

▶ **To cite this version:**

Mathilde Koch. Computational modeling to design and analyze synthetic metabolic circuits. Quantitative Methods [q-bio.QM]. Université Paris Saclay (COmUE), 2019. English. ⟨NNT : 2019SACLS467⟩. ⟨tel-02417453⟩

HAL Id: tel-02417453

https://theses.hal.science/tel-02417453v1

Submitted on 18 Dec 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

université PARIS-SACLAY

NNT : 2019SACLS467

Thèse de doctorat

UNIVERSITÉ PARIS SUD
Comprendre le monde, construire l'avenir

Computational modeling to design and analyze synthetic metabolic circuits

Thèse de doctorat de l'Université Paris-Saclay
préparée à l'Université Paris-Sud

École doctorale n°577 Structure et Dynamique des Systèmes Vivants (SDSV)
Spécialité de doctorat: Sciences de la Vie et de la Santé

Thèse présentée et soutenue à Jouy-en-Josas, le 28/11/19, par

Mathilde Koch

Composition du Jury :

Joerg Stelling Professeur, ETH Zurich (Department of Biosystems Science and Engineering)	Président et Rapporteur
Daniel Kahn Directeur de Recherche, INRA (UAR 1203, Département Mathématiques et Informatique Appliquées)	Rapporteur
Gregory Batt Directeur de Recherche, INRIA	Examinateur
Olivier Sperandio Directeur de Recherche, Institut Pasteur	Examinateur
Wolfram Liebermeister Chargé de Recherche, INRA (UR1404, MaIAGE, Université Paris-Saclay, Jouy-en-Josas)	Examinateur
Jean-Loup Faulon Directeur de Recherche, INRA (UMR 1319, Micalis, AgroParisTech, Université Paris-Saclay)	Directeur de thèse

Abstract

The aims of this thesis are two-fold, and centered on synthetic metabolic circuits, which perform sensing and computation using enzymes. The first part consisted in developing reinforcement and active learning tools to improve the design of metabolic circuits and optimize biosensing and bioproduction. In order to do this, a novel algorithm (RetroPath3.0) based on similarity-guided Monte Carlo Tree Search to improve the exploration of the search space is presented. This algorithm, combined with data-derived reaction rules and varying levels of enzyme promiscuity, allows to focus exploration toward the most promising compounds and pathways for bio-retrosynthesis. As retrosynthesis-based pathways can be implemented in whole cell or cell-free systems, an active learning method to efficiently explore the combinatorial space of components for rational buffer optimization was also developed, to design the best buffer maximizing cell-free productivity. The second part consisted in developing analysis tools, to generate knowledge from biological data and model biosensor response. First, the effect of plasmid copy number on sensitivity of a transcription-factor based biosensor was modeled. Then, using cell-free systems allowing for broader control over the experimental factors such as DNA concentration, resource usage was modeled to ensure our current knowledge of underlying phenomenons is sufficient to account for circuit behavior, using either empirical models or mechanistic models. Coupled with metabolic circuit design, those models allowed us to develop a new biocomputation approach, called metabolic perceptrons. Overall, this thesis presents tools to design and analyze synthetic metabolic circuits, which are a novel way to perform computation in synthetic biology.

Résumé

Les buts de cette thèse sont doubles, et concernent les circuits métaboliques synthétiques, qui permettent de détecter des composants chimiques par transmission de signal et de faire du calcul en utilisant des enzymes. La première partie a consisté à développer des outils d'apprentissage actif et par renforcement pour améliorer la conception de circuits métaboliques et optimiser la biodétection et la bioproduction. Pour atteindre cet objectif, un nouvel algorithme (RetroPath3.0) fondé sur une recherche arborescente de Monte Carlo guidée par similarité est présenté. Cet algorithme, combiné à des règles de réaction apprises sur des données et des niveaux différents de promiscuité enzymatique, permet de focaliser l'exploration sur les composés et les chemins les plus prometteurs en bio-rétrosynthèse. Les chemins obtenus par rétrosynthèse peuvent être implémentés dans des cellules ou des systèmes acellulaires. Afin de concevoir le meilleur milieu pour optimiser la productivité du système, une méthode d'apprentissage actif qui explore efficacement l'espace combinatoire des composants du milieu a été développée. La deuxième partie a consisté à développer des méthodes d'analyse, pour générer des connaissances à partir de données biologiques, et modéliser les réponses de biocapteurs. Dans un premier temps, l'effet du nombre de copies de plasmides sur la sensibilité d'un biocapteur utilisant un facteur de transcription a été modélisé. Ensuite, en utilisant des systèmes acellulaires qui permettent un meilleur contrôle des variables expérimentales comme la concentration d'ADN, l'utilisation des ressources a été modélisée pour assurer que notre compréhension actuelle des phénomènes sous-jacents est suffisante pour rendre compte du comportement du circuit, en utilisant des modèles empiriques ou mécanistiques. Couplés aux outils de conception de circuits métaboliques, ces modèles ont ensuite permis de développer une nouvelle approche de calcul biologique, appelée perceptrons métaboliques. Dans l'ensemble, cette thèse présente des outils de conception et d'analyse pour les circuits métaboliques synthétiques. Ces outils ont été utilisés pour développer une nouvelle méthode permettant d'effectuer des calculs en biologie synthétique.

Résumé français détaillé

0.0.1 Introduction

La biologie de synthèse est un domaine de recherche interdisciplinaire qui a vu le jour au début des années 2000, grâce à des articles fondateurs comme le repressilateur ou le switch bi-stable, mêlant connaissances en biologie et modélisation mathématique et permettant de programmer des comportements dans des systèmes biologiques. En effet, grâce aux avancées permises entre autres par la baisse du coût de la synthèse d'ADN, les biologistes peuvent désormais construire à façon des gènes, et tester leurs connaissances à une vitesse jusqu'à présent inégalée. L'objectif de la biologie de synthèse est de faire de la biologie une discipline d'ingénieurs, en standardisant et inter-opérant des éléments biologiques comme des gènes, leurs promoteurs, terminateurs et autres éléments de contrôle connus de l'expression génétique découverts par la biologie fondamentale. Depuis les débuts de ce domaine de recherche, beaucoup de travaux ont entrepris la construction de circuits génétiques ayant un comportement prédictible, comme des portes logiques ou des oscillateurs. Fabriquer de tels circuits permet de démontrer une connaissance suffisante de la biologie sous-jacente pour obtenir des comportements prédictibles, ouvrant la voie à de potentielles applications pratiques, notamment en médecine. Les échecs de ces circuits permettent aussi de découvrir les phénomènes biologiques responsables de ces erreurs et donc de progresser en biologie fondamentale. Cependant, un certain nombre de limites, dont la synchronisation temporelle des circuits génétiques comportant plusieurs couches ou la nécessité d'imposer un seuil pour considérer la réponse comme positive ou négative, alors que les concentrations des espèces biochimiques sont par essence des signaux analogiques continus, incite à proposer de nouvelles approches. Nous souhaitons donc dans cette thèse utiliser le métabolisme pour effectuer les calculs biologiques. Cette thèse propose des outils de conception et d'analyse pour développer des circuits métaboliques qui offrent une solution originale aux limites des circuits classiques de biologie de synthèse.

Tout d'abord, il est nécessaire de définir les défis de conception de ces circuits. Il faut dans un premier temps détecter les composés, grâce à des biosenseurs qui ne sont pas l'objet principal de la thèse mais qui sont évoqués en détail au chapitre 6. Une base de données de composés détectables par ce type de biosenseurs transcriptionnels est aussi disponible en chapitre 4. L'autre problème à résoudre est l'étape métabolique, qui transforme un composé non détectable en un composé détectable grâce à une enzyme, partie essentielle de nos futurs circuits métaboliques. Ce problème est conceptuellement similaire à celui de la rétrosynthèse, qui cherche à produire un composé cible à partir de composés chimiques disponibles à l'achat. La rétrosynthèse peut être formulée comme un problème d'exploration d'espace combinatoire, où les algorithmes employant la forte brute ne sont pas les plus adaptés. De nouveaux algorithmes d'apprentissage renforcé comme la recherche arborescente de Monte-Carlo (MCTS pour Monte-Carlo Tree Search par la suite) ont permis des succès majeurs dans des domaines comme les échecs ou le Go, où la force brute échouait. Le MCTS sera donc employé en chapitre 3 pour s'attaquer au problème de la biorétrosynthèse, en utilisant une heuristique de recherche adaptée au problème. La spécificité majeure de la biorétrosynthèse (comparée à la rétrosynthèse chimique traditionnelle) est la nécessité de modéliser la promiscuité enzymatique (la capacité de l'enzyme à catalyser la même réaction chimique sur des substrats différents), et sera présentée aux chapitres 1 et 2. En effet, la promiscuité enzymatique est un phénomène biologique majeur, et la considérer permet de résoudre le problème de bases de données actuellement incomplètes ainsi que de fournir des enzymes pouvant servir de point de départ pour de l'évolution dirigée pour catalyser de nouvelles réactions, qui est un sujet majeur pour l'industrie de l'ingénierie métabolique. Une autre approche d'apprentissage par renforcement, l'apprentissage actif, permet aussi d'évoluer dans des espaces combinatoires complexes, et sera utilisé au chapitre 5 pour optimiser le système de production dans lequel sont exprimés nos circuits métaboliques. En effet, ceux-ci seront exprimés dans des systèmes acellulaires: ce sont des lysats cellulaires supplémentés par des composants actifs permettant de remplacer certaines fonctions perdues lors de la lyse de la cellule (mix énergétique). Or, ce genre de système est extrêmement sensible à la concentration des composants chimiques du mix énergétique qui doit donc être optimisé.
Des outils d'analyse de résultats sont aussi nécessaires pour vérifier que le comportement du circuit conçu est bien le comportement attendu. Différents types de modèles en biologie existent, à différents niveaux de détail et de complexité. Je me suis cependant attachée à décrire en priorité les paramètres pouvant changer dans le contexte expérimental (la quantité d'ADN, de substrat pour la réaction enzymatique par exemple) et minimiser le nombre de paramètres qui ne sont pas variés dans la configuration expéri-

mentale afin de limiter les problèmes d'identifiabilité et de paramétrage du modèle. Des effets biologiques particulièrement intéressants dans le cadre de cette thèse étaient les effets de compétition de ressources, détaillés en chapitre 9, quand plusieurs gènes sont en compétition pour les ressources finies du systèmes (ribosomes et polymérases) pour exprimer la protéine qu'ils encodent. Dans ce contexte, pour développer des outils de conception et d'analyse de circuits métaboliques, le plan de la thèse est le suivant:

- Chapitre 1: Présentation de l'algorithme de rétrosynthèse et des règles de réaction telles que développées avant mon arrivée dans l'équipe
- Chapitre 2: Utilisation d'outils de rétrosynthèse pour explorer l'espace biochimique
- Chapitre 3: Présentation d'un nouvel algorithme de rétrosynthèse basé sur le MCTS
- Chapitre 4: Base de données de métabolites permettant de détecter le résultat de circuits métaboliques
- Chapitre 5: Méthode d'apprentissage actif pour l'optimisation d'un système acellulaire d'expression de gènes
- Chapitre 6: Construction des biosenseurs et utilité de la modélisation
- Chapitre 7: Développement et modélisation d'un biosenseur pour les flavonoides
- Chapitre 8: Revue de littérature sur les modèles adaptés aux systèmes acellulaires
- Chapitre 9: Implémentation et modélisation de circuits métaboliques simples en systèmes acellulaires
- Chapitre 10: Implémentation et modélisation de circuits métaboliques complexes en systèmes acellulaires

0.1 Outils de conception assistée par ordinateur

La première partie a consisté à développer des outils d'apprentissage actif et par renforcement pour améliorer la conception de circuits métaboliques et optimiser la biodétection et la bioproduction.

0.1.1 Chapitre 1: Sélection d'enzyme et découverte de chemins métaboliques

Les méthodes présentées dans ce chapitre ont d'abord été décrites dans l'équipe avant mon arrivée, avec le développement du logiciel RetroPath2.0,

encodé dans la plateforme KNIME. Les réactions biochimiques catalysées par des enzymes sont décrites en utilisant des règles de réaction, apprises depuis des bases de données en prenant en compte un contexte chimique variable (plus ou moins étendu) autour du centre de réaction. Cette prise en compte variable permet de modéliser la promiscuité, ou tout du moins de faire des hypothèses controllées de promiscuité lors de la biorétrosynthèse. Une des limites majeures de l'outil présenté dans ce chapitre est l'algorithme de force brute sous-jacent, utilisé pour effectuer la recherche de chemins métaboliques. Les limites des méthodes présentées permettent donc de justifier le développement de l'algorithme de RetroPath3.0, qui utilise cependant le même formalisme en chemo-informatique pour la modélisation des réactions enzymatiques et de la promiscuité.

0.1.2 Chapitre 2: Enumération de structures chimiques et cribable virtuel avec RetroPath2.0

L'objectif de ce chapitre est de présenter d'autres applications d'outils de rétrosynthèse, et particulièrement du formalisme de modélisation des réactions chimiques que nous utilisons dans l'équipe. Mon travail a consisté à prouver formellement que de nouvelles règles chimiques développées spécialement permettent d'énumérer l'ensemble des isomères d'une molécule. Ces règles d'énumération ont ensuite été utilisées pour optimiser la température de transition de polymères, prouvant que notre méthodologie est valide sur un cas simple. Ensuite, ces règles ont été comparées à des règles biochimiques (présentées au chapitre précédent) pour améliorer l'activité anti-bactérienne d'aminoglycosides, dans un processus de criblage virtuel. Elles fonctionnent mieux que les règles biochimiques car restent dans le domaine de validité de la QSAR utilisée pour la prédiction. Enfin, les règles biochimiques ont été utilisées pour compléter l'annotation du métabolisme d'Escherichia Coli, en s'intéressant au problème des métabolites non identifiés en spectrométrie de masse et en suggérant une structure et une voie de synthèse pour ces pics non identifiés. Ce problème est important en métabolomique et cet outil permet de suggérer des prédictions testables pour des biochimistes intéressés par ce type de problème. Ainsi, ce chapitre permet de démontrer la flexibilité et l'utilité de notre formalisme pour une variété de problèmes biologiques.

0.1.3 Chapitre 3: Recherche arborescente de Monte-Carlo guidée par similarité chimique pour l'ingénierie métabolique

Dans ce chapitre, je présente l'implémentation d'un nouvel outil de biorétrosynthèse, qui permet de résoudre certaines des limitations présentées dans les chapitres 1 et 2. Le formalisme utilisé pour décrire les règles de réaction, les apprendre à partir de bases de données et les appliquer à des composés chimiques sont les mêmes que dans ces deux chapitres. Ma contribution principale a été d'adapter l'algorithme de MCTS à notre problème, en le guidant avec un score permettant de prendre en compte à la fois la plausibilité chimique de la réaction que nous utilisons ainsi que la probabilité que nous ayons une enzyme à disposition qui catalyse réellement cette réaction. Le score chimique en particulier prend en compte à la fois le substrat et les produits attendus de la réaction, qui permet de trier les réactions par plausibilité chimique. J'ai développé le logiciel RetroPath3.0 autour de cet algorithme, et l'ai validé sur 2 jeux de données différents. D'abord, sur un premier jeu de données, petit et curé manuellement, la capacité de l'algorithme à retrouver les chemins métaboliques exacts décrits dans la littérature a été vérifiée. Ceci nous permet de nous assurer de la plausibilité biologique de nos prédictions. Ici, RetroPath3.0 obtient de meilleurs scores que la version précédente. Ensuite, sa capacité à trouver des chemins métaboliques pour des composés ayant été produits par ingénierie métabolique, mais sans connaissance du chemin exact, a aussi été vérifiée (sur un jeu de données plus conséquent de 152 composés). Les nombreux paramètres de recherche de RetroPath3.0 ont été testés et expliqués en détail dans une annexe pour faciliter le travail de développeurs souhaitant utiliser et améliorer l'outil. De plus, d'autres options ont été développées, pour accélérer (en terme de temps de calcul) la recherche de chemins, rendant l'outil plus performant pour des utilisateurs fréquents. J'ai ensuite démontré sa modularité en incluant des prédictions de toxicité dans l'algorithme, afin d'éviter des intermédiaires toxiques. De plus, une fonctionnalité a été ajoutée qui n'existe pas dans d'autres outils de biorétrosynthèse, et qui consiste à suggérer de potentiels suppléments à mettre dans le milieu de culture pour obtenir des chemins viables. Cette pratique est courante en ingénierie métabolique mais aucun outil ne permettait de réaliser ce genre de simulations auparavant. Ainsi, ce logiciel, qui résout certains des problèmes mentionnés dans les chapitres précédents, est un outil de biorétrosynthèse intéressant pour la communauté, modulaire et en accès libre. C'est un outil particulièrement utile dans le cadre de cette thèse, qui cherche à utiliser des chemins métaboliques pour effectuer des

calculs en biologie de synthèse, car il permet de concevoir les chemins en question.

0.1.4 Chapitre 4: Jeu de données de métabolites déclenchant une réponse cellulaire transcriptionnelle ou traductionnelle

Le but de ce chapitre concernant le cadre général de cette thèse (le développement de circuits métaboliques) est le suivant: en combinaison avec les algorithmes de biorétrosynthèse présentés aux chapitres précédents, un jeu de données de composés détectables peut être utilisé pour trouver un chemin enzymatique permettant de transformer un composé indétectable en composé détectable. Cette stratégie a été utilisée à de nombreuses reprises au cours de cette thèse, notamment aux chapitres 9 et 10. Ce chapitre présente donc la curation manuelle d'un nombre conséquent d'articles, pour permettre aux biologistes de synthèse de savoir quels composés ont déjà été détectés, avec quelle méthode et dans quel organisme.

0.1.5 Chapitre 5: Exploration à grande échelle guidée par apprentissage actif pour maximiser la production en système acellulaire

Dans ce chapitre, j'aborde un autre problème de la conception de circuits métaboliques et de l'exploration d'espaces combinatoires. La problématique était la suivante: étant donnée une protéine d'intérêt (qui pourrait aussi être un métabolite, tant qu'il est détectable à haut débit), comment choisit-on les composants chimiques du mix énergétique en système acellulaire pour maximiser la production de la protéine en question? Alors que l'espace combinatoire est bien trop grand pour être exploré de façon exhaustive, les algorithmes d'apprentissage actif, qui suggèrent une nouvelle série d'expériences pour optimiser une métrique d'intérêt, sont parfaitement adaptés à ce problème. J'ai donc développé une méthode d'apprentissage actif, couplée à des robots de distribution de liquides, pour optimiser la production de protéines en lysat cellulaire. La question conceptuelle est similaire à celle de la rétrosynthèse, car les deux problèmes sont des problèmes d'exploration d'espace combinatoire, comme présenté en introduction. De plus, d'un point de vue pratique, améliorer la qualité du lysat permet d'obtenir de meilleurs résultats expérimentaux en testant nos circuits métaboliques en systèmes acellulaires. Dans un premier temps, nous avons appliqué cette

méthode au lysat d'un expérimentateur expert, multipliant la production par 34 et obtenant un modèle prédictif en 10 itérations successives du cycle d'apprentissage actif. En effet, ce cycle consiste à demander à un ensemble de modèles de machine learning de prédire de quels points ils ont besoin pour optimiser leur capacités de prédiction et la production du système. Ainsi, plus les nombre d'itérations augmente, plus les modèles et la productivité s'améliorent. Dans un deuxième temps, après avoir identifié les points les plus informatifs de notre jeu de données (en cherchant à prédire l'ensemble de nos données à partir d'un sous-ensemble de taille fortement réduite), nous avons réussi à optimiser d'autres lysats, venant d'autres expérimentateurs ou supplémentés avec des antibiotiques. Cette méthode permet donc rapidement d'optimiser des lysats cellulaires pour des biologistes souhaitant utiliser ce type de systèmes pour exprimer une protéine d'intérêt, en utilisant des algorithmes d'apprentissage actif disponibles en libre accès sur GitHub.

0.2 Analyse et modélisation de circuits métaboliques: des données à la connaissance

La deuxième partie a consisté à développer des méthodes d'analyse, pour générer des connaissances à partir de données biologiques, et modéliser les réponses de biocapteurs. Grâce à ces méthodes d'analyse, nous avons aussi développé des circuits métaboliques de plus en plus complexes.

0.2.1 Chapitre 6: Biosenseurs transcriptionnels sur-mesure pour l'ingénierie métabolique

Ce chapitre décrit les avancées récentes en conception de biosenseurs transcriptionnels pour l'ingénierie métabolique. Ces biocapteurs sont une partie indispensable de n'importe quel circuit en biologie de synthèse, car ils permettent la détection de signaux de sortie du circuit. C'est pour cette raison que j'ai étudié ce type d'objet biologique durant ma thèse. Etant donné le nombre important de revues déjà parues sur ce sujet durant les dernières années, ma première contribution à cet article a été de définir un plan permettant d'identifier les défis et opportunités dans ce domaine dans les années à venir, moins souvent présentées dans les autres revues, à savoir développer des modèles mathématiques pour affiner les propriétés des biosenseurs et l'importance de développer des biosenseurs avec et pour les systèmes acellulaires. J'ai ensuite effectuée la revue de littérature présentée ici, permettant au lecteur d'avoir un nombre important d'exemples réussis pour trouver la

solution à son problème de détection, ainsi que les défis à venir pour le domaine.

0.2.2 Chapitre 7: Développement d'un modèle minimal et généralisable de biosenseurs transcriptionnels: l'exemple des flavonoides

Comme mentionné précédemment dans cette thèse, les biosenseurs sont une partie essentielle d'un circuit en biologie de synthèse, car ils permettent la détection de signal. C'est la raison pour laquelle développer, modéliser et analyser des biosenseurs est la première étape dans le développement d'outils d'analyse pour des circuits plus complexes, car la couche de détection de signal détermine tout ce qui peut être compris en amont de la détection. Ma contribution principale fut l'analyse mathématique et la modélisation de nos biosenseurs. Plus précisément, j'ai observé en analysant les données que changer le nombre de plasmides de nos biosenseurs changeait non seulement l'amplitude de la réponse, comme attendu, mais aussi sa sensibilité. J'ai donc choisi une stratégie de modélisation adaptée, basée sur une équation de Hill étendue, qui prend en compte le nombre de copies du plasmide à la fois pour la quantité de facteurs de transcription présents et pour le nombre de sites d'accrochage, qui est le degré de liberté de notre système. Après avoir ajusté le modèle aux données et vérifié la consistance des estimations de paramètres en utilisant des simulations aléatoires, le modèle a été utilisé pour suggérer des modifications à faire pour changer le comportement du biosenseur pour atteindre des objectifs donnés, comme une sensibilité plus ou moins importante, en modifiant le nombre de copies du plasmide, les forces d'attachement du facteur de transcription à l'ADN ou à son inducteur.

0.2.3 Chapitre 8: Les modèles pour les systèmes acellulaires en biologie de synthèse: plus facile, mieux et plus rapide

Les systèmes acellulaires offrent de nombreux avantages comparés à l'expression *in vivo*, en particulier pour le développement de circuits métaboliques synthétiques. Tout d'abord, ils permettent un prototypage bien plus rapide. Ensuite, dans l'objectif de fabriquer des circuits complexes, ils permettent un contrôle fin de la concentration d'ADN, qui est un élément essentiel des circuits présentés par la suite. Avant de développer un modèle, il faut donc connaitre l'état de l'art dans le domaine, présenté dans cette revue.

0.2.4 Chapitre 9: Des transducers métaboliques "plug and play" étendent l'espace chimique de détection de biosenseurs acellulaires

Dans ce chapitre, une implémentation de transducers métaboliques acellulaires, conçus en utilisant les outils présentés en partie 1, est présentée. Tout d'abord, un biosenseur de l'acide benzoique est développé et optimisé, modifiant à la fois la quantité de reporter et de facteur de transcription. Ma première contribution pour ce travail a donc été de modéliser ce test en utilisant des équations de Hill. Ensuite, des transducers (convertissant du signal à l'aide d'enzymes) ont été implémentés dans ces systèmes acellulaires, pour l'acide hippurique et la cocaïne. Un élément très intéressant dans ces deux systèmes est que le signal atteint un pic avant de redescendre à des concentrations intermédiaires d'enzymes, montrant que la compétition pour les ressources du système a un rôle majeur dans les effets observés. J'ai donc procédé en deux étapes, en modélisant d'abord le transducer de l'acide hippurique en incluant la compétition de ressources, puis en appliquant la même modélisation à la cocaïne, en prenant en compte les différences en terme de force des promoteurs de nos deux constructions. Cette modélisation, expliquant les données expérimentales, fut ensuite validée sur de nouvelles expériences en analysant un décalage du pic du signal du transducer de l'acide hippurique en variant la concentration de l'ADN codant pour le facteur de transcription, et augmentant ainsi la compétition pour les ressources. Ainsi, ce chapitre permet de démontrer l'importance de la prise en compte de la compétition pour les ressources en biologie de synthèse, à la fois pour la conception et l'analyse de circuits même relativement simples.

0.2.5 Chapitre 10: Perceptrons métaboliques pour du calcul neuronal dans des systèmes biologiques

Dans cet article, des circuits métaboliques complexes en biologie de synthèse furent développés en utilisant les outils de conception présentés en partie 1, et des outils d'analyse permettant d'apprendre d'expériences précédentes et d'améliorer les circuits. Plus précisément, dans un premier temps, les biosenseurs et les transducers métaboliques furent modélisés en utilisant des fonctions de Hill sur-mesure et modulaires. Ensuite, dans un deuxième temps, le comportement obtenu en combinant ces circuits a été prédit *in silico* et testé à la fois *in vivo* et dans des systèmes acellulaires. Notre but était d'abord de construire des additionneurs pondérés, qui peuvent

être modifiés pour fabriquer des perceptrons, le plus simple des algorithmes d'intelligence artificielle, qui sont essentiellement des sommes pondérées digitalisées. Le contrôle requis sur la quantité d'ADN des différents éléments (en particulier des enzymes) était uniquement possible en système acellulaire. Mon modèle fut utilisé pour prédire la quantité d'enzyme nécessaire à la réalisation des portes logiques que nous souhaitions implémenter dans nos perceptrons acellulaires. Ainsi, dans ce projet, ma contribution a consisté à analyser, modéliser et prédire les futures séries d'expériences. Ceci a permis le développement de circuits métaboliques complexes, qui sont donc une preuve de concept de cette nouvelle approche pour effectuer des calculs en biologie de synthèse.

0.2.6 Conclusion et perspectives

De nombreux défis restent à résoudre après ce travail de thèse. Dans les outils de conception développés, une meilleure prise en compte de la promiscuité permettrait des avances substantielles pour le domaine de l'ingénierie métabolique. De plus, la comparaison des méthodes existantes sur les mêmes jeux de données permettrait enfin à la communauté de comparer réellement les mérites des différents algorithmes proposés par les développeurs et serait une avancée notable. Par ailleurs, les outils de modélisation et d'analyse présentés en deuxième partie peuvent toujours être améliorés pour une meilleure prise en compte des réalités biologiques sous-jacentes, en analysant en détail les raisons des échecs de modélisation. De plus, les circuits complexes présentés en chapitre 10 devraient être testés avec d'autres briques élémentaires pour prouver la généralisation possible du concept, montré à l'heure actuelle sur un nombre réduit de cas. Cependant, malgré ses limites, la modélisation est une aventure essentielle en science car elle permet la formalisation de connaissance et l'identification du chemin restant à parcourir pour vraiment comprendre les phénomènes étudies.

Remerciements

Tout d'abord, je souhaite remercier le jury d'avoir accepté d'évaluer mon travail, et pris le temps de lire mon manuscrit.

Jean-Loup, merci de m'avoir fait découvrir le monde de la recherche, et d'avoir accepté d'encadrer mon travail de thèse. C'est grâce à mon intégration dans ton équipe que j'ai pu goûter aux joies de la recherche et voir ce que peut offrir une carrière scientifique.

Je souhaite remercier tous les membres de l'équipe BioRetroSynth et de l'INRA que j'ai pu côtoyer au cours de mes années de thèse, ainsi que les personnes avec qui j'ai pû collaborer à Montpellier et les membres de DocJ. C'est grâce à vous que ma thèse aura été aussi enrichissante, grâce à votre enthousiasme, nos discussions et votre soutien.

Thomas, cette thèse n'aurait pas été la même sans toi. Merci pour ces échanges scientifiques ou non pendant trois ans. C'est grâce à toi si j'ai pu apprendre et progresser autant en bio-informatique, et ta patience et tes conseils ont été précieux.

Olivier, ce fût un beau partenariat scientifique, intéressant, amusant et enrichissant. C'était un plaisir d'apprendre à piloter des robots avec toi, merci d'avoir été là pour la fin de ma thèse!

Manish, Olivier, Pierre, Heykel, Paul, Melchior, Ioana, Alexandra, Amir, Angelo, j'étais ravie de vous côtoyer au cours de ces années, de discuter de science ou autre lors de nos déjeuners animés et des fameuses pauses café.

Claire, ça fait bien longtemps qu'on se soutient, depuis ces premières années à Ginette et DPM. Des années qu'on se lance dans le même genre de projets, et qu'on peut compter l'une sur l'autre, pour parler science, thèse, projets

de vie, chocolat, que ce soit à Singapour ou en Californie. Du courage pour la fin de la tienne, et merci pour ce beau partage d'expérience.

Flore et Thibault, c'est grâce à vous qu'il y aura une crèche à ma soutenance. Merci pour toutes ces soirées chez vous, et d'être une oreille compatissante lorsque je raconte mes nombreuses et longues histoires.

Arthur, Clément, Victor, vos prénoms sont par ordre alphabétique pour ne pas faire de jaloux! Merci pour votre amitié toutes ces années, sans vous la vie serait moins belle.

Laure, Noémie, Axelle, Pierre, Thomas, Alexis, et tous les autres, merci pour votre amitié, votre soutien, et tous ces moments qui ont fait de ces années de thèse une belle période à vos côtés.

Baudoin, merci pour le soutien moral sans faille que tu m'as apporté pendant plusieurs années, pour toutes nos discussions qui ont fait avancer mon projet de thèse et mes projets de vie. Je suis heureuse de pouvoir partager autant avec toi, et notre rencontre est sans nul doute un point positif de toute cette aventure.

Papa, Maman, merci pour votre soutien depuis si longtemps, de toujours croire en moi et de me pousser à donner le meilleur et à oser me lancer dans de nouvelles aventures. C'est un peu grâce à vous si je suis là où je suis aujourd'hui. Merci à mes deux petites soeurs Louise et Noémie pour leurs questions qui m'ont forcée à vulgariser sur des sujets importants comme "mais en fait c'est quoi ton métier?" ou "à quoi ça sert de modifier des bactéries? D'ailleurs c'est quoi une bactérie?". Merci à ma grand-mère, mon frère et mon beau-père pour leur soutien.

Je souhaite enfin terminer en remerciant la DGA et l'Ecole Polytechnique qui ont financé mon projet de thèse.

Contents

Abstract **i**

Résumé **iii**

Résumé français détaillé **v**

0.1 Outils de conception assistée par ordinateur vii

0.2 Analyse et modélisation de circuits métaboliques: des données à la connaissance xi

Remerciements **xv**

Table of Contents **xix**

Introduction **1**

Synthetic biology's aims and advances 1

Design tools for metabolic circuits 7

Analysis and modeling tools for metabolic circuits 15

Thesis structure and contributions 19

I Computational design tools **23**

1 Enzyme Selection and Pathway Design **25**

1.1 Abstract 26

1.2 Introduction 26

1.3 Enzyme selection 27

1.4 Pathway Design 39

1.5 Summary and conclusion 46

2 Molecular structure enumeration **49**

2.1 Introduction 50

2.2 Results and Discussion 53

2.3 Conclusions 69

2.4 Methods 70
2.5 Supplementary Data 78

3 RetroPath3.0: Similarity-guided Monte Carlo Tree Search for metabolic engineering 81
3.1 Abstract 82
3.2 Introduction 82
3.3 Theoretical background 83
3.4 Results and Discussion 87
3.5 Conclusion 93
3.6 Materials and methods 95
3.7 Supplementary Tables 104
3.8 Supplementary Note 1: Parameters' Role and Effects 108

4 Detectable Compounds Dataset 117
4.1 Abstract 118
4.2 Data 120
4.3 Experimental design, materials and methods 120

5 Active learning for cell-free optimization 125
5.1 Abstract 126
5.2 Results and Discussion 126
5.3 Methods 132
5.4 Supplementary Data 140

II Analyzing and modeling metabolic circuits 149

6 Transcriptional Biosensors for Metabolic Engineering 151
6.1 Abstract 152
6.2 Introduction 152
6.3 Designing a transcriptional biosensor to detect a compound of interest 153
6.4 Computer-assisted fine-tuning of biosensor properties 157
6.5 Custom-made biosensors' new application domain: cell-free metabolic engineering 158
6.6 Conclusion 160

7 Building a minimal and generalisable model of transcription factor-based biosensors: Showcasing flavonoids 161
7.1 Abstract 162
7.2 Introduction 162
7.3 Materials and methods 164
7.4 Results 168

7.5 Discussion . . . 180
7.6 Supplementary Data . . . 183

8 Models for Cell-free Synthetic Biology 191
8.1 Abstract . . . 192
8.2 Introduction . . . 192
8.3 Translation and transcription processes in cell-free . . . 194
8.4 Resource competition in cell-free . . . 195
8.5 Metabolism in cell-free . . . 197
8.6 Conclusion . . . 198

9 Plug-and-Play Metabolic Transducers 199
9.1 Abstract . . . 200
9.2 Introduction . . . 200
9.3 Results . . . 201
9.4 Discussion . . . 210
9.5 Methods . . . 210
9.6 Mathematical Modeling of Cell-Free Biosensors . . . 215
9.7 Mathematical model derivation . . . 220

10 Metabolic Perceptrons for Neural Computing in Biological Systems 243
10.1 Abstract . . . 244
10.2 Introduction . . . 244
10.3 Results . . . 246
10.4 Discussion . . . 259
10.5 Methods . . . 260
10.6 Supplementary data . . . 269

Conclusion & perspectives 287
Design tools for metabolic circuits . . . 287
Analysis and modeling tools for metabolic circuits . . . 289

List of Symbols 294

List of Tables 295

List of Figures 297

Bibliography 301

Introduction

Synthetic biology's aims and advances

Introduction to synthetic biology

Ever since the dawn of time, mankind has wanted to understand and master the world. The first major revolution in human history is the start of agriculture, where our ancestors, by tending to, selecting, studying and eventually domesticating plants and animals started leading a sedentary life. This first revolution led to the emergence of larger groups of persons, that could be supported by farming instead of hunting and gathering. Cities and civilizations emerged and collapsed for centuries, until the world as we know it – connected, globalized and fueled by scientific discoveries – came to exist.
That thirst for knowledge led mankind to try and understand the greatest mystery of all: what life is and how it came to be, here on Earth. Understanding the basic mechanisms of life, from first principles to evolved mammals such as ourselves, has been a scientific endeavor for centuries, from Da Vinci's illegal dissections to Mendel's peas. What gave rise to modern biology is the fundamental discovery by Watson and Crick, in collaboration with Wilkins and Franklin, of the structure of deoxyribonucleic acid (DNA) [1] – the code for life. The fact that the genetic code was carried by a molecule, in which a single atom change could cause a phenotypic mutation, was hinted by the famous essay "What is life" by Schrödinger [2], but knowing the structure of the actual molecule allowed for unprecedented characterization of living systems. Biology evolved from an observational science to an experimental one, where scientists could tamper with and modify genes and DNA to understand their effect on phenotypes, allowing for unprecedented control of biological experiments.
Projects that were initially considered science fiction by contemporaries, such as the Human Genome Project [3], are now completed. While scientists hoped decoding our DNA would be enough to understand our biology,

phenomena such as epigenetics or environmental factors have been revealed to be major players and hamper our understanding of life. However, the declining costs of sequencing and synthesizing DNA (shown in Figure 0.1 [4]) gave rise to a new interdisciplinary field of scientific research: synthetic biology.

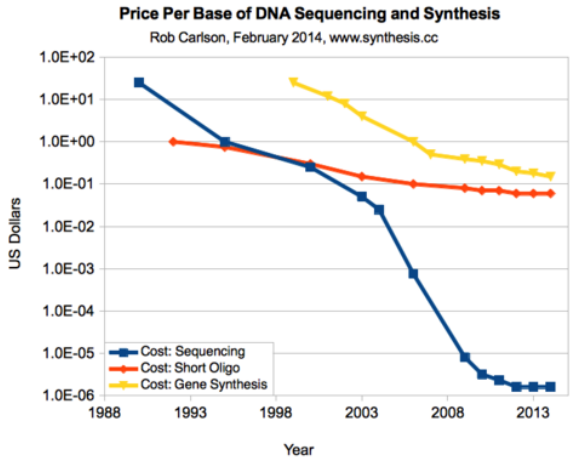


Figure 0.1 **Price per base of DNA Sequencing and synthesis.** Extract from [4]

As Richard Feynman (supposedly) famously wrote on his blackboard before his death, 'What I Cannot Create, I Do Not Understand'. Synthetic biology aims to create and understand novel biological objects, using standard parts that can inter-operate. The community wishes to turn biology into an engineering discipline, by designing and constructing new biological entities such as enzymes, genetic circuits, and cells that can be modeled, understood, and tuned to meet specific performance criteria. On the long way towards predictive engineering of biology, the scientific community can test its knowledge of underlying phenomena. For example, by simplifying the complex G protein-coupled receptor (GPCR) signal transduction system in yeast and reducing it to core, modular components, Ellis's team was able to understand and model much more deeply the functioning GPCR signaling than when studying the natural system [5].

While a more detailed history of the field from Monod [6] to 2014 is reviewed elsewhere [7], only the birth of synthetic biology as a discipline will be presented in this thesis. Synthetic biology as a field of research is thought to be born in 2000, with three papers that mark a date in the domain: the repressilator, the toggle switch and the negative auto-regulator. The repressilator is a network of three transcription factors repressing each other, thereby creating oscillations [8]. Such a network could also converge to a stable steady state, and the authors used mathematical modeling to identify important parameters such as protein half-time that influence the oscillations, and act upon those [8]. Gardner *et al.*, on the other hand, used two

transcriptional repressors repressing each other, which lead to stable states where one is dominant and the other repressed [9]. Small inducers allow from transitioning from one state to the other, implementing what is called hysteresis in mechanics. Lastly, Becskei *et al.* characterized the increased stability of negative auto-regulation versus absence of feedback in a simple loop, using both mathematical modeling and experimental verification [11]. The three systems have been extensively modeled, studied and reproduced in the past years and mark a milestone in synthetic biology. For example, Lugagne *et al.* used dynamic feedback and periodic forcing to maintain the oscillator in its unstable position [10].

While the rest of this introduction will focus on biological circuits and the current tools and strategies to design them, other notable advances in synthetic biology are worth mentioning to complete the picture of the field and the tremendous impact it could have on our understanding of the world. Briefly, examples include whole genome synthesis (*Mycoplasma genitalium* [12], *Escherichia coli* (*E. coli*) in 2019 [13], using 61 codons instead of 64, or *Saccharomyces cerevisiae* [14]), incorporation of unnatural amino acids [15] or inactivation of retroviruses in pig to potentially allow xenotransplantation [16]. All those emerging technologies are poised to change our world once they mature. An interesting perspective from field experts can be found in Church *et al.*, where the authors discuss possible technological advances, current hurdles, and links with society [17]. I will now focus on an important endeavor of synthetic biology: circuit design.

Current strategies for circuit design and their limits

Historically, for its short period of existence, synthetic biology has focused on building and testing circuits that have predictable behavior for a desired function, as is commonly done with logic gates in electronics. Those circuits generally share two important properties: they are digital and genetic. Genetic, because they preferentially use transcriptional regulation, through multiple genetic layers, to perform computation [18, 19]. Genetic digital circuits are extremely useful and have been used in a variety of contexts, including biosensors for detection of pollutants [20, 21] and medically-relevant biomarkers [22, 23], smart therapeutics [24, 25], and dynamic regulation and screening in metabolic engineering [26, 27, 28]. A number of design methods and tools are available for those that intend to design genetic circuits.

Some of those software tools are particularly noteworthy of interest. Cello [29] is essentially a programming language to design computational circuits in living cells, with the user specifying the desired function, sensors, actuators and elements of interest such as organism and validity operating

conditions. The software then generates DNA sequence encoding the desired circuit. The ribosome binding site (RBS) Calculator [30] allows the user to design libraries for RBS that span order of magnitudes for his gene of interest. Prediction of RBS strength is of particular interest in synthetic biology has it allows tuning the expression of the gene of interest. A range of tools exist to design DNA parts, perform codon optimization (selecting the best synonymous codons for a given amino acid in the organism the protein will be expressed in), visualize plasmids or predict clustered regularly interspaced short palindromic repeats (CRISPR) binding sites or otherwise facilitate experimental design.

Multiple examples and opinions for the future of genetic circuits have been published [31, 32, 33, 34, 35]. Using their software Cello, Voigt's team managed to automatically design 60 circuits, of which 45 functioned as planned, while they also built a 3-gate circuit consisting of four layers with CRISPR/Cas [29]. Despite its elegance and a number of successes, using genetic circuits has some downfalls, especially unexpected causes for failures due to either compositional context (surrounding DNA parts), host context (using the circuit in another organism) or environmental context (circuit behaves differently, for example according to media composition) [36, 37]. A few element of particular interest will be mentioned here: load ([38, 39], leading to the creation of a load driver by [40]), small number of inputs, time synchronization between branches of the circuit [41, 42], burden (with the recent availability of burden controllers [43, 44]). Ideas and concepts from control theory to address those issues are becoming more and more routine for synthetic biologists [45].

As an alternative to genetic circuits, metabolism can be used to perform computation, as asked in an opinion paper [46], and performed by Courbet *et al.* [47] or by living organisms [48]. DNA computation [49], while impressive for solving complex computational problems [50, 51], only efficiently functions *in vitro*. Digital logic has tremendously changed the world of electronics, by allowing faster computation, digital storage and enabling multiple applications that make the world as we know it today. However, using digital logic in biology is sometimes problematic, as defining threshold between 0s and 1s for different inputs and moreover across layers can become tricky. On the other hand, using analog computation is much closer to how organisms naturally process information [52] and has been implemented using genetic layers [53, 54].

Recent enabling technologies

In the past decades, numerous enabling technologies have driven the explosion of the field of synthetic biology, tackling problems such as part sourcing or speed of assembly (see Figure 0.2). One of the most obvious ones is the falling price of DNA sequencing and synthesis, as shown in Figure 0.1. Decreasing costs mean that a biologist, instead of painstakingly obtaining one DNA sequence from an organism of interest that can be hard to cultivate, and spending weeks in the process, can now order codon-optimized sequences from a provider and obtain them in a couple of weeks. In 2008, this was expressed as a hope for the synthetic biology [55] ("One of the hopes of the synthetic biology community is to shift from reliance on laborious classical genetic engineering techniques to DNA synthesis") and this is now a reality. Such a decrease in synthesis cost would not have had such an impact if it had not been accompanied by new assembly techniques that allow for faster gene assembly, such as Gibson [56] or golden gate assembly [57, 58].

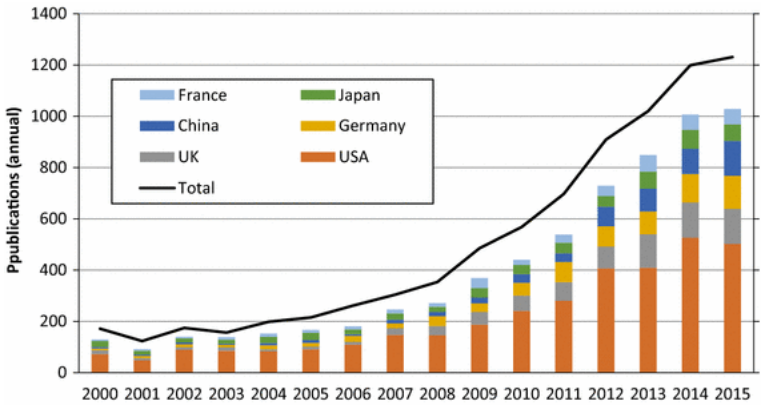


Figure 0.2 **Evolution of the number of publications in synthetic biology**. This Figure is obtained from [59]. No changes were made to the Figure, which was published under a Creative Commons Attribution 4.0 International License.

The advent of liquid handling robots could also be an enormous step forward for the field, by speeding up both the assembly and testing stages of the engineering process. While those are harder to master than previously expected by the community, they are at the heart of some interesting projects published in the past years. Indeed, the number of experiments that can be performed with a robot and high throughput testing is incommensurate with the number of experiments performed on a day to day basis by a biologist. While utilizing both robots and high throughput measurements is not adaptable to any project or situation, generating such data with the will, manpower and skills to analyze it has the potential to revolutionize synthetic biology and metabolic engineering (a field whose objective is to optimize bioproduction of chemicals). They provide possibility of large scale experi-

ments in settings where we know we do not have a sufficient understanding of the phenomenon to efficiently model it and data really does provide an advantage: testing genes for metabolic engineering projects, buffer compositions, enzyme rational modifications, screening of genes from uncultured organisms... Biofoundries seek to provide integrated informatics and experimental tools to facilitate experimental design and construct testing, so as to free synthetic biologists from repetitive tasks and give them means to test ideas in a higher-throughput fashion [60]. Just as standards for data exchange in Synthetic Biology (Synthetic Biology Open Language (SBOL), [61]) and systems biology (Systems Biology Markup Language (SBML), [62]) and associated software compatible with those standards allowed for greater reproducibility and comparisons, widely adopted standards and software to allow easy and reproducible programming of liquid handling robots will hopefully mark a milestone in the field. Such a standard could be Roboliq [63], or BioBlocks [28]. Even without such advanced data and program sharing practices, liquid handling robots coupled with active learning already contribute to synthetic biology [64] or chemistry [65].

Another technology that is in its early stages but could become a game-changer is cell-free systems. The idea of cell-free systems is not new: break a cell wall so that the cell is not alive anymore, and use the metabolic, transcription and translation machinery to produce biological components of interest to the experimentalist instead of fueling cell growth. This technology has been used for production of proteins and biological research since the fifties, but has received renewed interest from the synthetic biology community recently [66]. Cell-free transcription-translation (TX-TL) was really born in 2003 [67] and a detailed video protocol was later published [68]. It has since been deeply investigated and improved, and numerous strategies for regenerating energy have been tested and implemented to better the energy mix provided to the extract [69]. While numerous explanations exist to explain why cell-free systems exhaust over time, the reason is likely a combination of all of the following: DNA and RNA degradation, metabolite exhaustion, toxic products accumulation, ribosome degradation... Despite the current limits on our understanding of those systems, they can and have been used for extensive prototyping for various projects: biosensor development [70], prototyping of enzymes for metabolic engineering projects [71, 72] or part characterization [73]. Moreover, circuits developed for cell-free systems can be freeze-dried, stored at room temperature and reused months later, making for potential great diagnostic tools outside the laboratory [74].

Aim of this thesis in the context of synthetic biology

In the larger context of synthetic biology, this thesis has one major aim: propose a new way to perform computation, using metabolic analog circuits instead of digital genetic ones, which would overcome some of the limitations presented above. Moreover, this new way to do computation in synthetic biology is developed mostly in cell-free systems which allow for faster prototyping as presented previously. Furthermore, due to the originality of the proposed approach, new tools and methodological developments were required. Therefore, this thesis is centered around building design and analysis tools for metabolic circuits. I will present the state of the art in those two domains and what this thesis brings to those two topics.

Design tools for metabolic circuits

How to design a metabolic circuit

We define synthetic metabolic circuits as small networks of metabolites and reactions that perform an arbitrary operation, whose result can be analyzed. The problem of designing metabolic circuits can be separated in two different sub-problems, which require vastly different skills and tools to solve. The first one is the problem of output detection: how do we detect the output from a metabolic circuit, i.e. a metabolite. Indeed, any attempt at designing complex circuit becomes futile if the output of the circuit cannot be reliably detected. While biosensor design was not specifically tackled in this thesis, huge improvements in this domain have been made in the past years, and are presented in Chapter 6 [75]. For this thesis, I collected various compounds that are deemed detectable, either using publicly available databases or literature curation. This work is presented in Chapter 4 [76], is publicly available on GitHub and can be used as a starting point for a computer and human readable representation of detectable outputs of metabolic circuits, or inputs to digital ones.
The other problem to solve is much more complex: how to select an enzyme, or multiple enzymes, that can chemically transform a given compound into a detectable one? This problem is conceptually similar to the problem tackled by retrosynthesis [77]: how to produce a given compound with a set of allowed starting chemicals in a chemical synthesis setting? More precisely, the concept of retrosynthesis was first proposed by Corey in 1969 [78], which led him to earn the Nobel Prize in chemistry in 1990 [79]. In his Nobel lecture [80], Corey defines retrosynthesis analysis as: "a problem-solving technique

for transforming the structure of a synthetic target molecule to a sequence of progressively simpler structures along a pathway which ultimately leads to simple or commercially available starting materials for chemical synthesis." While I will detail the specificities of retrosynthesis and bio-retrosynthesis (cheminformatics representation notably) in section "The specifities of retrosynthesis" below, I will first present its striking similarities with other informatics problems from which we can learn.

Algorithms for navigating complex combinatorial spaces

Retrosynthesis can be formulated as a combinatorial search problem, where we aim to find a path of chemical reactions leading to our desired product from various initial compounds. It is a combinatorial search because multiple chemical reactions could be used at each step, but not all of them lead to useful products, which will cause a combinatorial explosion when using brute force algorithms, becoming computationally intractable. Other problems of this class include finding the best buffer composition for highest protein expression (presented in Chapter 5, and submitted for publication), the best amino acid sequence for yellow fluorescence [81], the best sequence of moves to win a game of chess or Go [82, 83], or even the best staff schedule for organizing shifts in a hospital under constraints [84] and playing video games [85]. All those situations present common features: huge combinatorial spaces (modifying k amino acids of a sequence of interest leads to 20^k combinations, without including potential unnatural amino acids), huge branching factor (i.e.: what one can do next from a given position). For example, in the case of buffer optimization, when testing 4 different concentrations for 11 components, the design space is composed of 4,194,304 different combinations. In chess or in Go, the branching factor is the number of moves allowed in a given position: it is on average 35 for chess and 250 for Go [86]. In retrosynthesis, this would be the number of chemical transformations that apply to a compound. For example, in work by Segler and co-authors, the branching factor is 46,175 when considering all chemical transformations that can be applied to a substrate, which the authors reduce to 50 by selecting the most promising ones [87]. As seen from the previous examples, strategies from various branches of algorithmics can be used to tackle these apparently different but computationally similar problems.
A major class of algorithms to perform such searches are reinforcement learning algorithms, defined as "an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward" [88]. A history of the field until 1996 can be found in Kaelbling *et al.* [89]. In plainer terms, reinforcement learning

algorithms are self-learning algorithms, that are composed of parts exploring the combinatorial space (allowed moves in a game, chemical reactions in retrosynthesis), and a part giving a reward to the algorithm if it is closer to the desired objective ('better' chance to win in the game, more realistic image, less traffic jams and other specific metrics, as these algorithms can be applied to a wide range of problems). Definitions of the moves and the rewards are application-specific. Major leaps have been taken since this review [89], notably because of the power of deep learning and the computational hardware and software advances that make it possible. Not only did reinforcement learning combined with deep learning change Go or chess computer programs, but it also could have tremendous impact in fields as diverse as traffic control [90], image recognition and generation [91], targeted advertising [92] or news recommendation [93] (examples are taken from [94]). Two sub-classes of reinforcement leaning algorithms have been used in this thesis: Monte-Carlo Tree Search (MCTS) in Chapter 3 and active learning in Chapter 5.
Active learning is conceptually simple and presented in Figure 0.3: given a set of starting points, a machine-learning method is trained on those points, and suggests new experiments, based on a feature to optimize which can be either the value of interest (fluorescence, activity), the uncertainty of the model (where the model knows the quality of its prediction is low) or a combination of both [95]. The point of the method is that the algorithm itself decides which experimental points to test next. This is often used in settings where new information (given by the oracle) is costly in time or resources: which document to translate by a professional translator to feed into a language translation algorithm [96], which amino acid sequence to test to improve this enzyme activity [97] and so on. However, despite their simplicity, those algorithms require *a minima* a way to evaluate a new data point, and often the uncertainty associated to this new data point, meaning they are not suited to any problem.

A MCTS is another reinforcement learning algorithm. The term Monte-Carlo Tree Search was coined by Coulom in 2007 [98], by applying Monte Carlo methods (i.e.: solving deterministic but intractable problems using random sampling) to tree searches for decision making in Go. The general idea is to build a tree to explore the search space, while balancing exploitation of promising branches and exploration of unknown directions. It proceeds in iterations, where the steps of one iteration are presented in Figure 0.4. The search starts at the Tree root, and children are selected using a Tree policy based on the score of the node and the number of times it has been visited. A leaf child is expanded by adding children to it, according to chosen heuristics. The original part of the algorithm, using Monte Carlo

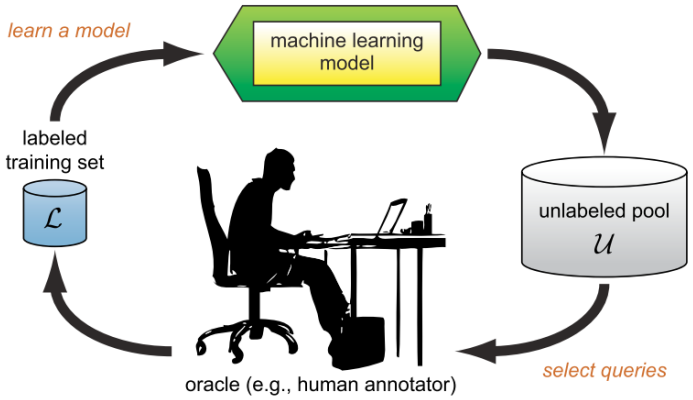


Figure 0.3 **Principles of active learning**. This Figure is obtained from [95]. In synthetic biology, the oracle is the result of an experiment performed by a human or a robot.

to evaluate the node, comes at the simulation step: random exploration is performed from this node. In chess or Go, that would usually correspond to playing a number of random moves until the game is won or lost. The last step, termed back-propagation, consists of updating the Tree with the results of the simulation. The tree itself decides what to explore to achieve the reward it ought to maximize: winning in chess or Go [82, 83], scoring higher at various video games [85] or optimizing a staff schedule according to staff constraints [84].

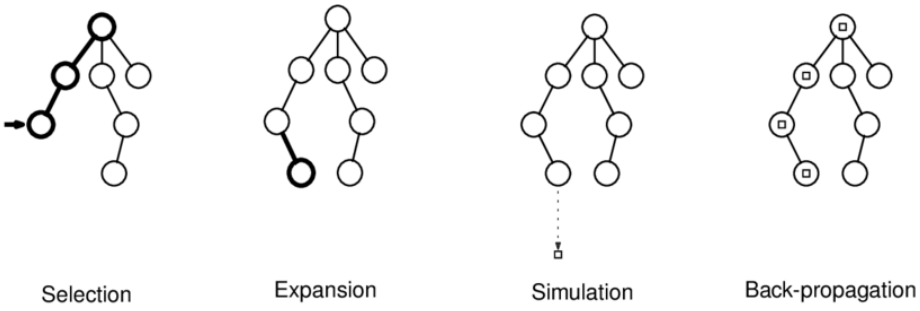


Figure 0.4 **Principles of Monte Carlo Tree Search**. This Figure is obtained from [95]. **Selection**: Tree is traversed with the *tree policy*, which is the way to select the points within the tree, balancing exploration and exploitation of known nodes. **Expansion**: New node added to the tree, selected using the *tree policy*. **Simulation**: rollouts are played from new node using the *default policy*, which is used to explore unknown space by random evaluation (Monte Carlo). **Back-propagation**: Final state value is back-propagated to parent nodes.

One advantage of MCTS (and a notable difference with active learning) is that it provides a systematic way to evaluate points even if this evaluation is not straightforward. For example, in a game of chess, a state of the chess board cannot be truly evaluated unless the game is finished and one of the players has won. While one could argue that who has the advantage could be evaluated by the number of pieces each player possesses or predefined

set-ups that are supposed to give a strategic advantage, the real evaluation of who has won is only at the end of the game. Evaluating the board based on heuristics can provide hints but no definitive answer. This is also true in retrosynthesis: one can only evaluate a synthesis plan for a given chemical once all the starting chemicals are found, despite a synthesis plan seeming promising. In order to overcome this problem, the advantage of MCTS is to perform random simulations, which can be theoretically be performed until the end of the game. Those algorithms are incredibly powerful, as exemplified by the history of the win of programs against human masters in chess and Go (detailed below). Therefore, in contrast to active learning where the point of interest needs to be evaluated by an oracle, in MCTS the evaluation comes from random simulation of the possible outcomes from the point and the search does not necessitate oracle intervention.

In 1997, a historical milestone was reached in computer science when Deep-Blue won its chess games against Gary Kasparov, especially in the eyes of the general public [99]. Yet, the algorithm proceeded by brute force: it evaluated all combinations, there was no artificial intelligence involved, just incredible calculation power, which is why the achievement was later downplayed [100]. Such an approach is not possible in Go, as there are many more combinations of moves ("In total, the number of different possible arrangements of stones stretches beyond 10^{100}" [101]), that cannot all be evaluated in any meaningful way by a brute force algorithm given humanity's current and foreseeable computational calculation power. It was then seen as the next frontier in computational science applied to games [102]. In 2016, experts were still saying that computer programs did not stand a chance against human players [103]. Yet, AlphaGo defeated Lee Sedol 4 times out of 5 with a program trained on thousands of expert games and some self play [104]. This was seen as an incredible feat (acknowledged in Science's magazine's breakthrough of the year [105]), but AlphaGo's latest version, that only learned through self play with reinforcement learning using a MCTS algorithm guided by neural networks, without ever seeing a human game, won against its previous version 100-0 [82], showcasing the incredible power of MCTS guided by deep reinforcement learning. Those programs changed professional chess, as more and more scandals of players cheating with smartphones are reported [106], and also gave rise to a new discipline: centaur chess, where players and machines are teamed together against other centaurs. What is interesting is that centaurs beat both humans and machines individually, showing associating human creativity and machine calculating power makes the best of both worlds [107].

The specificities of retrosynthesis

Retrosynthesis, and especially bio-retrosynthesis, does not only necessitate an efficient combinatorial search algorithm. Indeed, while moves in chess or Go are straightforward to implement computationally, that is not the case with enzymatically catalyzed chemical transformations, which are the basic moves to be carried out in retrosynthesis. I will first explain how chemical reactions can be encoded and then the specificity of using enzymes, which require a way to encode promiscuity (i.e.: capacity to catalyze multiple chemical reactions, on different substrates).

Chemical reaction encoding

While a chemical reaction can be conceptually simple to represent with the names of the compounds (i.e.: 6 carbon dioxide + 6 water = glucose + 6 oxygen), the question becomes more complex once one intends to account for the actual chemistry of the reaction, and not reason simply by the names of the molecules. Molecules have long been represented by structural formula (2D graphs) or 3D representations that were hard (or computationally inefficient) to manipulate, but new standards emerged and are now widely adopted, such as the IUPAC International Chemical Identifier (InChI) or Simplified Molecular Input Line Entry System (SMILES) (that uses methods from graph theory to depict molecules in a simple ASCII string [108]) for molecules [109] and the SMILES arbitrary target specification (SMARTS) language for reactions. While the InChI has the immense advantage of providing a unique depiction for every chemical, both machine and human-readable (for experts), it is not efficient for cheminformatics manipulation. On the other hand, SMILES are not unique despite efforts towards canonization by the community, but allow to work with SMARTS, which render possible easy matching of substructures in a molecule, in a format that is once again both human and machine-readable. This depiction of substructures in SMARTS allow for representation of a chemical reaction as a rule depicting a graph transformation between two sub-graphs encoding the changes occurring at the reaction center [110]: one sub-graph of the substructure common to all known substrates, the other to all known products. An important point to note with the SMARTS representation is that an atom–atom mapping (AAM) between the reactants and products is usually required for computing this representation of the chemistry of the reaction. Moreover, both when calculating the AAM and applying the rule under a SMARTS format to a compound, finding a Maximum Common Substructure (MCS) is necessary, which is a NP-hard problem and therefore computation-

ally intensive, as well as highly dependent on the quality of the input data (for example, chemical reactions should be balanced, i.e.: have the same number of atoms of each type on each side).
Other approaches that do not necessitate AAM or graph handling were therefore also developed by the cheminformatics community. Such an approach was pioneered by Faulon [111] and further developed by Carbonell and Faulon on "reaction signatures" [112, 113], that encode compounds in vectors where bits correspond to the occurrence of a given fragment in the molecule. This conceptually resembles the Extended Connectivity FingerPrint (ECFP), that are standard for calculating compound similarities [114]. Using these representations of compounds, a reaction can be computed as a difference between products and substrates signatures, where bits now encode the net changes in fragments between substrates and products. Such representations allows for much faster reaction computation, but necessitate computationally intensive reconstruction of the molecule from a bag of fragments once the reaction has been applied. Moreover, fragment-based representations are often used for machine-learning applications in cheminformatics as they allow an easy and chemically relevant representation of data, as a bit representing whether a fragment is present or not in a molecule has chemical significance [115, 87].
In this thesis, I mostly used SMARTS representation for reaction rules (taken from the RetroRules database [116]), and ECFP for compound similarity calculations. Other formats for representing compounds and reactions include Bond Electron Matrix [117], Reaction-center Difference Match patterns [118, 119] or MACCS keys [120]. However, once cheminformatics representations are chosen for computing the reactions rules, questions remain on how to treat the problem of enzyme promiscuity.

Enzyme promiscuity

First, enzyme promiscuity needs to be defined. "Substrate promiscuity occurs when enzymes carry out their typical catalytic functions using non-canonical substrates. By contrast, 'catalytic promiscuity' occurs when the catalytic abilities of the active site are used to catalyze a distinctly different type of reaction." [121]. The term moonlighting can also be found for proteins that perform multiple biological functions, and catalytic promiscuity for enzymes would fall in that domain [122]. The alternative substrates concerned by an enzyme's substrate promiscuity usually share some chemistry, such as bonds, groups or overall aspect [123]. Enzyme promiscuity is a key characteristic of enzymes, as it is thought to be a major driver of evolution or adaptability of organisms [124, 125, 126, 127], as well as key for

metabolic engineering as promising starting enzymes for protein engineering [128] or for screening against new substrates [129]. It is now thought to be common [130, 123] and have influence on metabolism in what is termed "underground metabolism". It could explain failure modes of current gene essentiality modeling [131] and some approaches propose including it for extending or gap-filling of metabolic models [132]. It is interesting to note that while enzymes can reach high catalytic efficiency on those alternative substrates, most enzyme databases (with the notable exception of Braunschweig Enzyme Database (BRENDA) [133]) record only the main reaction, that has the highest catalytic rate. It is therefore necessary to model enzyme promiscuity to make the most of current knowledge on the biochemistry of enzymes and of the data currently available in databases.
Various attempts have been made at either predicting [126] or scoring enzyme promiscuity [134]. However, none of these approaches really solve the problem of encoding promiscuity for retrosynthesis, where the degree of promiscuity is paramount. Different strategies have been adopted meanwhile. The BNICE framework uses reaction rules based on the EC number classification [135, 117, 136, 137], that sorts enzymatic reactions according to the chemistry involved and are usually extremely generalist (i.e.: using a broad description that can apply to a number of molecules). On the other hand, the approach taken in this thesis is to adopt a more data-driven method, by automatically generating rules from databases of interest [116, 138]. The user can tune the amount of promiscuity allowed, using a feature called the diameter that will be described in more details in Chapters 1, 2 and 3 and has been described in the literature [112, 116, 138]. Tuning the amount of promiscuity, used in conjunction with an efficient search algorithm, is key to control the combinatorial space explosion as more generalist rules apply more broadly on substrates of interest and generate more products that themselves have to be included in the retrosynthetic search.

The first part of this thesis will investigate the problem of using efficient algorithms for combinatorial search for metabolic circuit design. More precisely, both the issues of pathway design using retrosynthesis and buffer optimization using active learning will be covered. The second part of this thesis will concern itself with analyzing the results from those metabolic circuits, while current analysis and modeling tools are presented in the next part of this introduction.

Analysis and modeling tools for metabolic circuits

As mentioned at the beginning of this introduction, synthetic biology was defined from the start as an engineering discipline, stressing its use of mathematical modeling for circuit design and understanding. However, as was sensed from the beginning of the discipline, care must be taken due to our limited knowledge of biological effects and the degree of details required in a model. For example, as early as 2007, emphasis was put on the importance of modeling, to reduce the number of experiments needed to understand the behavior of a biological system [139]. However, Kaznessis argues that there should be first a reflection on the type of model according to the situation: from an engineering point of view, the model needs to be 'at the level of design degrees of freedom'. The author then pointed out two such degrees of freedom in synthetic biology: either at the network level by changing the topology of the circuit, or at the bio-molecular level when one wishes to play on promoter sequences or chemical species quantities. The synthetic biologist therefore has to consider not only his knowledge of the biology underlying his system, but also the tools he can leverage to study it. Only parameters that can be experimentally controlled, tuned or modified need to be modeled as their impact on the system's behavior can be studied. There is no advantage from an engineering standpoint in modeling effects that cannot be controlled or modified in a given experimental set-up (ribosome diffusion in the cell for example). This is an approach fit for synthetic biology seen as an engineering discipline. Obviously, modeling can also be seen as an interesting endeavor in itself, and as a way to formalize knowledge and identify gaps that we need to fill in our knowledge of a system [140]. Modeling can be a way to ensure our knowledge of a system is sufficient for understanding its behavior, or even be a driver of new biological discovery. I will first make a broad presentation of different modeling strategies, mainly for synthetic biology and metabolic engineering, before stressing the current challenges faced specifically when developing metabolic circuits.

Different modeling strategies

Kinetic models

Kinetic modeling has a long history in biology, including for example the famous works of Michaelis and Menten on enzyme biochemistry [141] or Monod for bacteria growth rates [6]. Kinetic models can have varying de-

grees of detail, from a couple of biochemical species, to dozens or to hundreds of them, each level of detail with its own type of challenges. The difficulty of small-scale models will be to capture the subtleties in the data, while the difficulties in larger scale models will be the parameterization and the efficient computational simulation of such systems. Another key issue for large scale models is to be identifiable: i.e., that parameters can indeed be obtained from experimental data, which is obviously a growing issue as the network and number of parameters grow in size [142]. One interesting point to keep in mind is that "as measuring techniques improve, models will necessarily need to become more complex to match the experimental results" [55]. The authors stress again the fact that models should be simple as long as they accurately represent data, and augment in complexity along with experimental techniques. There is no necessity in modeling with extensive detail and unknown parameters a system that can be readily explained with a simpler model using for example the equilibrium assumption about transcription factor binding. Interesting examples include the modeling used in the original papers of the repressilator [8] or the toggle switch [9] for small scale; on a larger yet manageable scale, a detailed mechanistic modeling of an arsenic biosensor including different promoters, feedback regulation and architectures has been performed [143]. The most comprehensive mathematical model for a whole cell so far is the genome-scale model of *Mycoplasma genitalium* which represents all the formalized knowledge we have about this organism [144]. A lot of effort was put in this work to efficiently simulate different cellular processes using the best tool available for each, and manage the different time-scales they occur at.

Stoichiometric models

Using a different approach, constraint based modeling aims at considering large scale systems (often genome-scale models) and representing biological knowledge in the form of constraints on reactions or fluxes, and usually solving this system for a given objective function. The most famous example of constraints based modeling is undoubtedly Flux Balance Analysis (FBA), where the assumption of steady-state allows for fast solving using linear programming, and the assumption of optimality for the cell allows finding fluxes and growth rates among the authorized space of flux given by the constraints. While FBA has had numerous applications since its inception in biotechnology or to identify essential genes [145], numerous methods exist to develop more complex and biologically accurate stoichiometric models [146]. For example, various frameworks extend FBA to include limited cellular resources (resource balance analysis) [147, 148], thermodynamic constraints

[149], promiscuity of enzymatic reactions [132] or enzyme structure activity for assessing impact of temperature on the organism [150]. Some modelers argue for the importance of hybrid models that integrate constraints-based modeling with kinetics to allow dynamics simulation (dynamic FBA) [151]. While this is a burgeoning and fascinating field of research, this type of modeling does not typically apply to questions in synthetic biology but rather in metabolic engineering as they answer questions from a much more systemic point of view than the behavior of the circuit of interest.

Current challenges in synthetic biology modeling

A number of reviews and opinions exist that present current challenges for synthetic biology. Karamasioti and co-authors notably insist on the importance of accounting for context dependency and robustness of circuit design to unknown or varying parameters [36]. Those problems, among others, will be briefly presented below. Then I will present where this thesis fits in the context of modeling in synthetic biology applied to metabolic circuits.

Resource competition and load

While it has been known for a long time in synthetic biology that excessive demands on cellular resources can change circuit behavior, only recently can burden be actually measured experimentally [43, 44]. While numerous strategies now exist to model resource competition [152], Del Vecchio's team developed easy to use, well-thought and understandable tools now widely used for modeling those effects in synthetic biology [153, 154]. Those models modify the Hill equation [155] traditionally used for modeling promoter activation in synthetic biology circuits to account for those limited resources. Other approaches consider fractions of ribosomes dedicated by cells to different partitions of the proteome: circuit proteins, ribosomes, metabolic enzymes, and housekeeping proteins. This allows to analyze the interactions of gene circuits with the host cells (including growth rate) and not only the effect of increasing circuit size like previous model [156].

Balancing mechanistic and empirical modeling

As has been hypothesized a number of times in this introduction, adapted models for synthetic biology have the correct amount of mechanistic versus empirical modeling so as to explain degrees of freedom for the bio-engineer while remaining tractable. Yordanov *et al.* link steady-state dose response

shifts of biological systems to the underlying biochemical parameters [157]. Other papers look at the importance of modeling DNA copy number [158, 159] to understand effects beside simple increase in mRNA production.

Efficient parameterization of models

A major question for synthetic biology at the moment, notably for large scale kinetic modeling, is the correct parameterization of models with biologically relevant parameters. One way to approach the problem is to make consistent parameters from available databases, knowing their limits (such as experimental conditions for measures, *in vitro* measurements and not *in vivo*, reliability of measurements ...) and build consistent parameter sets from those databases [160]. Others include efficiently characterizing parameter uncertainty in the model [161], including by identifying circuit topologies whose behavior is robust to variation in parameters [162].

Single-cell modeling

With the expanding experimental high-throughput toolkit available to the experimentalist, it is now possible to gain insight into single-cell circuit behavior. This allows fascinating discoveries into whether differences in behavior are due to stochastic effects of the process in itself (such as transcriptional bursting [163]), or if the cells have slightly different rates for those processes, which would also explain the population distribution of circuit response [164]. Calibrating models for single cells [165] and generalizing such measures to ensure true understanding of a circuit behavior are on their way but the synthetic biology community is not there yet. Real time control of a single cell proves this is indeed possible given the correct experimental set-up and computational tools [10].

Bacterial community modeling

One of the next frontiers in synthetic biology is community distributed pathways or circuits [166]. Despite advances in modeling such behavior, for example quorum sensing [167], there is still much to do, notably to make the most of newest parallel computing technologies with tools tailored for biology to represent individual trajectories of cells in a population [168]. Those models can also be interfaced with other modeling approaches presented before, to model both individual cells and community approaches with sufficient level of detail to capture interesting collective behaviors.
While numerous challenges exist in mathematical modeling for the field of

synthetic biology, the main aim of this thesis regarding metabolic circuits was not theoretical but practical: it was to have models that could reproduce the experimental data at hand, to ensure our knowledge was sufficient to explain the observations, and give insights to make the circuits function, rather than to develop new modeling strategies. However, given the importance of resource competition in our circuits, this was included in Chapters 9 and 10 with two different strategies best adapted to the situations at hand.

Thesis structure and contributions

Due to the extensive collaboration this thesis involved, my contributions, mentioned in the present introduction, will also be detailed at the beginning of each chapter. The conclusion will also analyze my contributions and their limits.

Part 1: Computational design tools: efficient navigation through complex combinatorial spaces for retrosynthesis and experiment design

The first part of this thesis will be focused on utilizing efficient algorithms for navigating complex combinatorial spaces to design metabolic circuits.

- With Chapter 1 [169], we will see in details the retrosynthesis algorithm that was developed prior to my arrival in the team (RetroPath 2.0). Within this Chapter, I wrote the pathway design part.

- With Chapter 2 [170], we will see how such tools can be used for navigating the chemical space for purposes other than retrosynthesis. For this Chapter, my main contribution was the mathematical proof of the use of such algorithms for isomer enumeration.

- With Chapter 3, we will see how a new similarity-guided search algorithm using Monte Carlo Tree Search was implemented for more efficient retrosynthesis. I conceived, developed and tested the software, as well as wrote the article.

- With Chapter 4 [76], I will present a dataset that was published to facilitate use of retrosynthesis for biosensor design, necessary to detect metabolic circuits output. For this Chapter, I did literature curation

and data formatting. I also regularly updated the dataset with literature curation.

- With Chapter 5, we will see the use of active learning algorithms for improved protein production in cell-free systems. For this Chapter, I developed the active learning methods used in the submitted article presented as Chapter 5. This can be used to optimize the cell-free composition for improving metabolic circuits' expression.

Part 2: Analyzing and modeling metabolic circuits: from data to knowledge

The second part of this thesis will be focused on analysis and modeling tools for metabolic circuits.

- With Chapter 6 [75], I will review how to build biosensors and the next steps in the field of transcriptional biosensor design and tuning. Biosensors are the first building block for metabolic circuits, as they allow signal detection, and I wrote the review presented in this Chapter.

- With Chapter 7 [171], we will see the development and modeling of an *in vivo* biosensor for pinocembrin and naringenin. My work in this article involved modeling our biosensor, and notably accounting for the effect of plasmid DNA copy number on biosensor sensitivity.

- After this *in vivo* work, my colleagues and I decided to use cell-free systems for the rest of our metabolic circuit developments, for the various advantages they present. I will first review the state of the art of modeling cell-free systems in a review presented as Chapter 8 [172].

- We will then see with Chapter 9 [173] a cell-free implementation of simple metabolic circuits designed with the tools presented in Chapter 1. My contribution to this article consisted of modeling our results for understanding whether our current knowledge on resource competition could explain the results obtained by the other authors.

- Finally, I will present in Chapter 10 [174] an implementation of metabolic circuits. Those were also designed using tools presented in Chapter 1. My contribution was the design and empirical modeling of the individual parts of the metabolic circuits, and prediction of circuit behavior

for successive rounds of experiments, in collaboration with the other authors of the article.

Part I

Computational design tools: efficient navigation through complex combinatorial spaces for retrosynthesis and experiment design

1 Enzyme Discovery: Enzyme Selection and Pathway Design

This work was published in Methods in Enzymology by Pablo Carbonell, Mathilde Koch, Thomas Duigou and Jean-Loup Faulon.
Only minor modifications to the published paper have been introduced in the Chapter below.

Detailed contribution to this thesis

The methods discussed in this publication were first described in [138] and [175]. P.C. wrote the Enzyme Selection section, M.K. and T.D. wrote the Pathway Design section and J.-L.F. supervised the project.
The aim of this Chapter is to present methods used in the team before my arrival to perform retrosynthesis and my contribution presented in the RetroPath3.0 Chapter. The biochemical reactions catalyzed by enzymes are described using reaction rules, learned from data taking into account more or less of the chemical context around the reaction center, allowing us to encode enzymatic promiscuity. One of the main limits of the tools presented in this Chapter is the brute force algorithm that is used to perform the search, as such a huge combinatorial space necessitates better search algorithms. The limits of the methods presented in this Chapter therefore justify the novel developments presented in the RetroPath3.0 Chapter. However, the chemical reaction rules extraction and usage is conceptually identical.

Full reference

Carbonell P., Koch M., Duigou T., Faulon J.-L. (2018) Enzyme Discovery: Enzyme Selection and Pathway Design. *Methods in enzymology* 10.1016/bs.mie.2018.04.005.

Contributions as stated in the article

Not available.

1.1 Abstract

In this protocol, we describe *in silico* design methods that can assist in the engineering of production pathways that are based on enzymatic transformations. The described protocols are the basis for automated processes to be integrated into an iterative Design–Build–Test–Learn (DBTL) cycle in synthetic biology for chemical production. Selecting the right enzyme sequence for a desired biocatalytic activity from the extensive catalogue of sequences available in databases is challenging and can dramatically influence the success of bioproducing chemical compounds. A method for enzyme selection is presented that helps identifying candidate enzyme sequences through a scoring approach that considers not only sequence homology but also reaction similarity. Selecting a viable biochemical pathway for compound production requires screening large sets of reactions in a process involving combinatorial complexity. A method for pathway design using retrosynthesis is presented. The protocol allows the discovery of alternative chemical pathways leading to the final product by using reaction rules of selectable degree of specificity. The protocols can be reversed through clustering discovery and product identification processes. The integration of these protocols into a general pipeline provides a toolbox for enhanced automated synthetic biology design and metabolic engineering.

1.2 Introduction

Industrial biotechnology is facing two main challenges to develop more sustainable alternatives to petroleum-based chemistry: (1) its high R&D costs and (2) the limited range of compounds currently available for bioproduction. Computational strategies could help in addressing both these issues by integrating the increasing number of available tools into a unified pipeline to develop engineered organisms for the production of high-value compounds [176]. Enzymes are the essential building blocks enabling the chemical biotransformations leading to formation of the desired compounds. Several tools exist to select candidate sequences for the enzymes at each step of a given pathway including antiSMASH for biosynthetic gene clusters [177], as

well as tools based on reaction homologies like EC-Blast [178] and E-zyme [179] or based on machine learning [180, 181].
Alternative metabolic pathways are first *in silico* designed and assessed before being built and tested [182, 183, 184, 185]. While some computationally driven strategies make combinations of known metabolic reactions albeit not necessarily in the same chassis [186], others allow to design pathways that use novel reactions not stored in metabolic databases through the use of promiscuity hypotheses [187, 188, 112, 138, 189, 137, 190, 191].
In this protocol, we describe *in silico* design methods associated with the engineering of pathways for chemical production through enzymatic transformations with the goal of automating processes that can be integrated into iterative DBTL cycles like those present in synthetic biology projects. As depicted in Figure 1.1, we consider several scenarios where computational methods can provide solutions to enzyme-related design problems often found in the context of bioproduction of chemicals: enzyme selection, cluster discovery, and pathway design.

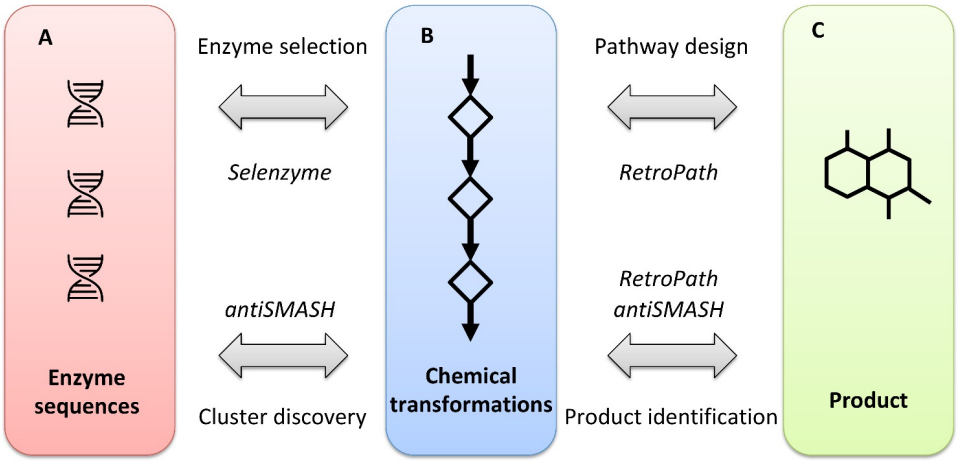


Figure 1.1 **Enzyme selection, cluster discovery, and pathway design.** (A) Enzyme selection identifies candidate enzyme sequences. (B) Pathway design allows discovering alternative chemical transformations leading to the product. (C) The discovery process is reversed when the final product is unknown, cluster discovery allows annotating chemical transformation for a given gene cluster, and product identification predicts chemical products of the pathway.

1.3 Enzyme selection

1.3.1 Introduction to Enzyme Selection

This protocol considers the scenario where a production pathway and its associated product are known. In that case, we use computational methods to replace individual enzymes by more efficient and/or specific alter-

natives. Choosing appropriate enzyme sequences to efficiently catalyze the set of reactions that are contained in production pathways is a relevant design problem whose solutions will depend on the specifications, for instance whether the system will work on cell or cell-free environments or the process conditions. Selecting the right starting enzyme sequences for each case can dramatically influence the success of the overall project. Low-performance enzymes can create bottlenecks in the pathways and lead to poor titers, yields, and productivity. When bottlenecks are present, enzyme engineering provides strategies like directed evolution for adaptation of the enzyme sequence toward the desired activity. However, applying directed evolution might not be always feasible and, even when it is the case, choosing the right initial parent enzyme is an important requirement to increase success rates. Here, we describe a method for automated sequence selection with the ability to mine reactions that are expressed in generalized rules. Our approach provides a systematic way for addressing either natural or non-natural chemical diversity of biosynthetic pathways by limiting ourselves not only to natural biochemical reactions but also to *de novo* engineered pathways. The method can therefore assist in the selection of heterologous candidate enzymes to express for both natural and non-natural target reactions.

1.3.2 Protocol Description (Selenzyme)

Enzyme sequence selection is often performed based on sequence homology, i.e., by mining protein databases like UniProt through Basic Local Alignment Search Tool (BLAST) searches. This approach is useful in order to identify families and classes of enzymes that potentially share similar functions. Multiple sequence alignments of homologous can help us to identify alternative parent enzymes by looking at conserved regions and hot spots for directed evolution through [192]. However, this approach has still some room for improvement, as described in this protocol by looking closer at the chemical transformations that are achieved by these homologous sequences. Notably, looking for enzymes with similar annotated chemical reactions helps us to identify alternative enzymes not found by performing a search that is based solely on sequence alignments.

Preparation Steps

Pathway Representation and Use of Generalized Transformations The goal of this method is to have a proper methodology that can be applied to systematically search for enzyme sequences and for any given pathway such as the ones identified by protocols described in this chapter. Most importantly,

the approach should be able to work with reactions for which no known sequences are available in enzyme databases. The method should be able to provide a list of suggested enzyme sequences as parent templates for enzyme discovery of sought activities.
The first step is to encode each of the reactions in the pathway in a meaningful way that can be processed to compare chemical similarity.

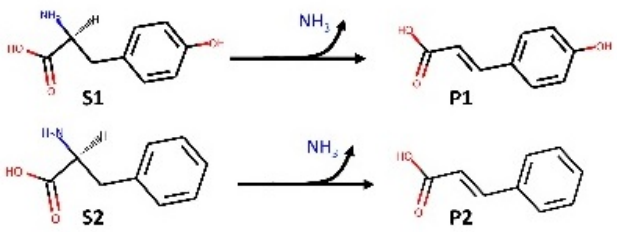
1. Reactions can conveniently be represented using the SMILES/SMARTS notation [108], as it provides an intuitive notation and can be either sketched by hand or through a chemical editor.
2. A reaction SMILES is formed by combining together the SMILES symbols for reactants, agents (optional) such as catalysts, solvents, etc., and products (a detailed example is shown in Figure 1.4). Whereas SMILES representation for chemicals is commonly found in online reaction databases like KEGG, MetaCyc, etc., reactions represented in SMILES format are found less frequently. For instance, the reaction database Rhea allows downloading any reaction in Molecular Design Limited (MDL) RXN format rather than SMILES.
3. Molecular format converters like OpenBabel [193] or molecular toolkits like RDKit [194] can read and interconvert between reaction formats.
4. Moreover, a useful feature of the SMARTS representation is that it allows for generalized transformations, generally called reaction rules. Reaction rules are a powerful tool in this context because they allow us to define a class of reactions and therefore to consider alternative enzyme candidates. In this chapter, the power of using reaction rules for retrosynthesis is shown, and Figure 1.4 provides examples of SMARTS encoding for reactions.

Computing Reaction Similarity

Several approaches are possible in order to define a metrics that measures similarity between reactions. Notably, Reaction Decoder Tool (RDT) [195] (available at https://github.com/asad/ReactionDecoder) provides one type of reaction comparison based on bond changes, reaction centers, or substructures. This tool requires AAM in order to be able to identify reaction centers and bond changes (see section about atom–atom mapping in this chapter). Here, we take a different approach to describe a simplified algorithm that does not require AAM and is based on computing similarities between reactants.

Description of the general workflow In order to compute similarity between two chemical species, we employ a fingerprint approach as described in the following workflow and summarized in Figure 1.2.

A.



B.

S1=Tyrosine	11111111111111111111011111111111111111111111111111111111111111
P1=Coumarate	11111111111111111111111111111110111111011111111111111111111110111
S2=Phenylalanine	11111111111111111111011111111111111111111110111111111111111111111
P2=Cinnamate	11111011111011111101111111101001111110111111111111111111111110110
S=S1 AND S2	11111111111111111111111111111111111111111110111111111111111111111
P=P1 AND P2	11111011111011111111111111110101111111111111111111111111111111110
L1=S1 AND NOT P1	00000000000000000000000000000100000010000000000000000000000001000
L2=S2 AND NOT P2	00000100000100000010000000001011000000100000000000000000000001001
L=L1 AND L2	00000000000000000000000000001000000010000000000000000000000000000
R1=P1 AND NOT S1	00000000000000000000100000000000000000000000000000000000000000000
R2=P2 AND NOT S2	00000000000000000000100000000000000000000001000000000000000000000
R=P1 AND P2	00000000000000000000100000000000000000000000000000000000000000000

Figure 1.2 **Computing reaction similarity. A** Similarity between tyrosine ammonia lyase EC 4.3.1.23 (top) and phenylalanine ammonia lyase EC 4.3.1.24 (bottom) is compared by using fingerprints. **B** For clarity, short fingerprints of 64 bits were computed using RDKit limited for the two substrates (tyrosine and phenylalanine) and the two products (coumarate and cinnamate). A first approach consists on computing similarity between left (S) and right (P) reactants by counting the number of common bits in the fingerprint. A second approach consists on computing left (L1, L2) and right (R1, R2) fingerprints for the reaction centers (bits not in common between left and right reactants) and then to count number of common bits in the left (L) and right (R) reaction centers.

1. Fingerprints are binary vectors where each chemical feature is represented by one or more components of the vector as bits [196]. The features spanned by the vector are generally related to topological elements of the chemical structure like fragments, pharmacophores, etc. Since the number of components that exist in the chemical space can become very large, fingerprint vectors are usually limited in size through a compression technique known as "hashing". Hashed fingerprints can be computed by several cheminformatics toolboxes like RDKit for Python or through workflow systems like KNIME [197].

2. To compute similarity between chemicals, we compare their fingerprints. For that purpose, there are several metrics available [196], the Tanimoto similarity being one of the most widely used. For binary fingerprints, Tanimoto similarity can be computed as the ratio of common bits divided by the total number of bits in the molecules, making this calculation very efficient.
3. In order to compute similarity between reactions, we can use two types of strategies based on reactant or reactions fingerprints.

   Reactant fingerprints

   - For each left reactant, we compute pairwise similarities to each right reactant and keep the closest one.
   - For each right reactant, we compute pairwise similarities to each left reactant and keep the closest one.

   Reaction fingerprint

   - Define an overall reaction fingerprint as the difference between the sum of fingerprints of the products and the sum of reactants.
   - Alternatively, it can be computed by using OR and XOR logical operations instead of computing the sum and difference, respectively. However, this last approach has the drawback that it does not make a difference between chemical features that are created or transformed through the chemical reaction.
4. Finally, reaction similarity is computed by comparing the resulting fingerprints
   - For reactant fingerprints, average both left and right closest similarities in order to obtain an overall similarity between the two reactions.
   - For reaction fingerprints, compute similarity between the overall fingerprints.
   - An alternative solution for reaction fingerprints is to keep a table with two fingerprint vectors, one consisting on the substrate features that are transformed and another one for the product features and measuring similarity in a similar approach as for reactant fingerprints (see Figure 1.2).

Reaction Directionality In the previous description, we assumed a preferred direction for the reactions. When this is not the case, we may compute similarity in both directions in order to keep the one that is the closest one. In principle, databases like MetaCyc, KEGG, or BRENDA [198, 133] contain information about reaction′s preferred direction, reversibility and, increasingly, data about free Gibbs energy. When this information is not available,

several strategies are possible in order to assign a preferred direction to the reaction.

1. Estimation of reaction thermodynamics by using group contribution methods [199]. Each chemical group in the reaction contributes to the free Gibbs according to some tabulated experimental data. The balance between left and right groups provides roughly an estimate of free energy and eventually allows determining the reaction direction. This is the strategy used by some databases like MetaCyc [200] and in some organism models [201].
2. Knowledge-based reaction direction inference. If a set of reactions with a high level of confidence knowledge of their direction are available, we can use such information in order to infer the most plausible direction for another given reaction. The algorithm is as follows:
    a) Start with initial set of reactions with curated direction information.
    b) Collect information about reactants and cofactors for each reaction in the reference set.
    c) Extract a list of currency metabolites (common cofactors) from a reference database.
    d) FOR each non-annotated reaction:
        i. Identify left and right cofactors and main reactants based on information from a b and c.
        ii. Search for top similar annotated reactions
        iii. Extract frequency of left/right cofactor pairs.
        iv. Set direction with highest frequency in top similar reactions based on left/right cofactor pairs

Screening, ranking and selection

Database screening In order to identify candidate enzyme sequences for a given chemical transformation, screen the target reaction against the reactions in an annotated metabolic database like Metacyc, MetaNetx [202], or Biochem4j [203].

1. The calculation should be performed in a computationally efficient way. A similarity algorithm like the one described in the previous section based on fingerprints is therefore required.
2. Reactions in the database are classified in descending order of similarity according to the fingerprint algorithm. For each reaction, add annotated sequences to the candidate list until either the maximum

number of allowed sequences is reached or some similarity minimum threshold has been exceeded.

Properties′s calculation The list of candidate sequences obtained in the previous step might contain redundancies as well as over-representation for some reaction classes. Therefore, the next step is to analyze the set in order to characterize each protein according to key properties and to ultimately provide some ranking leading to candidate selection.

1. Based on the information contained in the database, reactions can be classified according to their reaction Enzyme Commission (EC) class. This information can help in order to perform a quick initial preselection.
2. Redundancy in sequences can be easily identified through CD-HIT [204], which provides a fast way of clustering large sets of sequences.
3. A multiple sequence alignment of the sequences is then applied to get a more detailed comparative analysis of the sequences with the advantage that it can help to identify conserved regions in the protein, using T-Coffee [205] or ClustalW.
4. Additional position-based and global properties can be computed using packages like EMBOSS [206], amino acid indexes [207], or online tools like PredictProtein.
5. Other properties: position-specific scoring matrices (PSSM) using PSI-BLAST, functional domains using pfam collections and HMMER.

Ranking of Sequence Candidates Once the list of sequence candidates with calculated properties has been generated, the list needs to be prioritized. The criteria for ranking might vary depending on the application. For enzyme sequences that are going to be expressed in heterologous host, a ranking can be formed based on the following criteria:

1. Protein properties: percentage of secondary structure (helices, sheets, turns), molecular weight, isoelectric point, percentage of polar amino acids, etc.
2. Functional properties: target reaction similarity, UniProt protein evidence, sequence conservation in the alignment, etc.
3. Host-specific properties: sequence taxonomic distance to host organism, solubility or toxicity toxicity.

By combining together these properties through weighted sums, we can obtain a score for the sequence. Weights in the score can be heuristically fitted

by downloading the proteome sequences of the host organism and calculating typical values obtained for each of these parameters. A more complex scoring function can be also developed by employing machine learning using host-specific values for computed protein properties as a training set.

Selenzyme: Online Enzyme Selection Tool

Selenzyme is a free online tool for enzyme selection that integrates the aforementioned features to efficiently search for enzyme sequences starting from some target reaction [175].

1. User′s queries consist of a target reaction expressed in SMILES or RXN format or cross-referenced to an external database identifier or EC classification.
2. The software outputs a table of candidate sequences that can be ranked based on different criteria. In addition, the user can manually add or remove additional sequences to the table. A multiple sequence alignment of the output table can be visualized through MSAViewer [208].

An example of output table from Selenzyme is shown in Table 1.1. The input query was the reaction SMARTS shown in Figure 1.4. Several properties were computed for the sequences that are annotated for the closest reactions to the target. A score is calculated for each sequence based on decreasing order by reaction similarity, taxonomic distance, sequence conservation, and protein evidence.

Score	Seq. ID	Organism source	Tax. Distance	Protein Evidence	Consv. Score	% Helices	% Sheets	% Turns	% Coils	Mol. Weight	Isoelec. Point	Polar (%)
127.7	P61891	*E. coli* O157:H7	3	3	58	39.5	33.8	7.4	24.7	32,337.3	5.4603	39.423
98.7	P61895	*Raoultella terrigena*	6	3	59	35.9	38.3	8.6	24.9	23,055.38	5.043	40.889
97.7	Q83Q04	*Shigella flexneri*	6	3	58	39.2	33.8	7.8	24.7	32,349.36	6.5453	39.423
77.7	Q59838	*Salmonella muenchen*	8	3	58	41.2	31.8	8.6	24.3	29,502.96	5.7944	40.636
69.7	P61897	*Pectobacterium carotovorum* subsp. *carotovorum*	9	3	60	28.3	40.6	13.8	29	16,148.7	9.297	42.208
65.7	C5BF98	*Edwardsiella ictaluri* (strain 93-146)	9	3	56	38.2	35.1	6.8	25.3	32,348.45	4.9809	36.859
57.7	A1JIV0	*Yersinia enterocolitica* serotype O:8/biotype 1B (strain NCTC 13174/8081)	10	3	58	35.6	34.6	9.5	25.8	32,581.62	5.815	40.193
56.7	P37226	*Photobacterium profundum*	10	3	57	41.6	34.5	7.1	22.3	32,293.2	4.7141	37.5

46.7	B5FGF5	*Vibrio fischeri* (strain MJ11)	11	3	57	43.1	29.2	8.8	24.4	32,252.09	4.5494	37.942
46.7	A8FRU0	*Shewanella sediminis* (strain HAW-EB3)	11	3	57	47.1	31.2	4.4	22.7	32,010.79	4.5659	37.299
43.7	Q5WU94	*Legionella pneumophila* (strain Lens)	11	3	54	36.6	30.6	14.3	23.6	36,063.95	5.7759	45.758
39.7	A4G5Z9	*Herminiimonas arsenicoxydans*	12	3	60	47.6	27.5	9.9	20.1	35,364.65	7.0784	41.641
38.7	Q5H496	*Xanthomonas oryzae* pv. *oryzae* (strain KACC10331/ KXO85)	12	3	59	42.3	29.5	9.6	23.7	34,925.04	5.4464	39.939
34.7	B0BQN0	*Actinobacillus pleuropneumoniae* serotype 3 (strain JL03)	12	3	55	51.5	29.2	6.6	17.9	33,404.75	5.4643	38.17

31.7	Q14IT0	*Francisella tularensis* subsp. *tularensis* (strain FSC 198)	12	3	52	45.9	33.7	11.6	14.2	34,090.77	7.3844	40.752
29.7	Q8XXW5	*Ralstonia solanacearum* (strain GMI1000)	13	3	60	47.3	30.4	6.7	20.8	35,422.8	6.6924	39.818
29.7	Q393V1	*Burkholderia lata* (strain ATCC 17760/ LMG 22485/ NCIMB 9086/ R18194/ 383)	13	3	60	54.5	26.9	9	14.7	35,131.36	5.7875	40.549
28.7	A1VRQ1	*Polaromonas naphthalenivorans* (strain CJ2)	13	3	59	51	22.1	8.3	23.7	34,845.99	5.4269	38.415
24.7	Q2GCH6	*Neorickettsia sennetsu* (strain Miyayama)	13	3	55	51.2	30.4	5.7	18.1	33,851.77	8.3718	39.683

Table 1.1 **Output Table From Selenzyme Based on Reaction SMARTS for EC 1.1.1.37**

Integration With RetroPath2.0 and Other Workflows Selenzyme has been integrated into the SYNBIOCHEM-automated DBTL pipeline for fine chemical production [209]. The output of the pathway discovery tool RetroPath2.0, described in this chapter, can point to Selenzyme queries allowing enzyme sequence discovery for each step in the pathway.
Selenzyme provides a RESTful service allowing query submission through web-based applications. A KNIME node [197] making use of the RESTful service is available at http://www.myexperiment.org/packs/734. The reaction query can be generated using the cheminformatics workflows in KNIME and the resulting tables containing sequences can be easily processed downstream.

Other Applications of the Protocol

1. The described protocol has been focused on enzymes in production pathways. However, the application of the protocol can be extended to other applications such as in the development of enzyme-mediated biosensors [77], in the design of transporters, etc.
2. The protocol described here provides a first step when selecting for some target reaction. The resulting list of candidate enzymes as well as the resulting multiple sequence alignment (Figure 1.3) can be used as a starting point in order to carry out structure-based enzyme analysis of design.

Improvements of the Protocol

Several improvements are possible on the described protocol.

1. Estimating enzyme efficiency based on kinetic parameters when they are available. Database like BRENDA [133] provides kinetic values obtained by enzymatic assays, which can provide a first estimate of enzyme efficiency.
2. Using machine-learning techniques to predict enzyme efficiency using kinetic parameters in databases as training set [181].
3. Balancing multi-enzyme pathways: enzyme selection for multi-step pathways should consider overall pathway performance, i.e., searching some compatibility in the reactions (same source, kinetics, etc.). This approach can also be used to guide pathway tuning through transcriptional and translational parts either to refactor a natural gene cluster or to balance a *de novo* pathway from multiple heterologous sources.

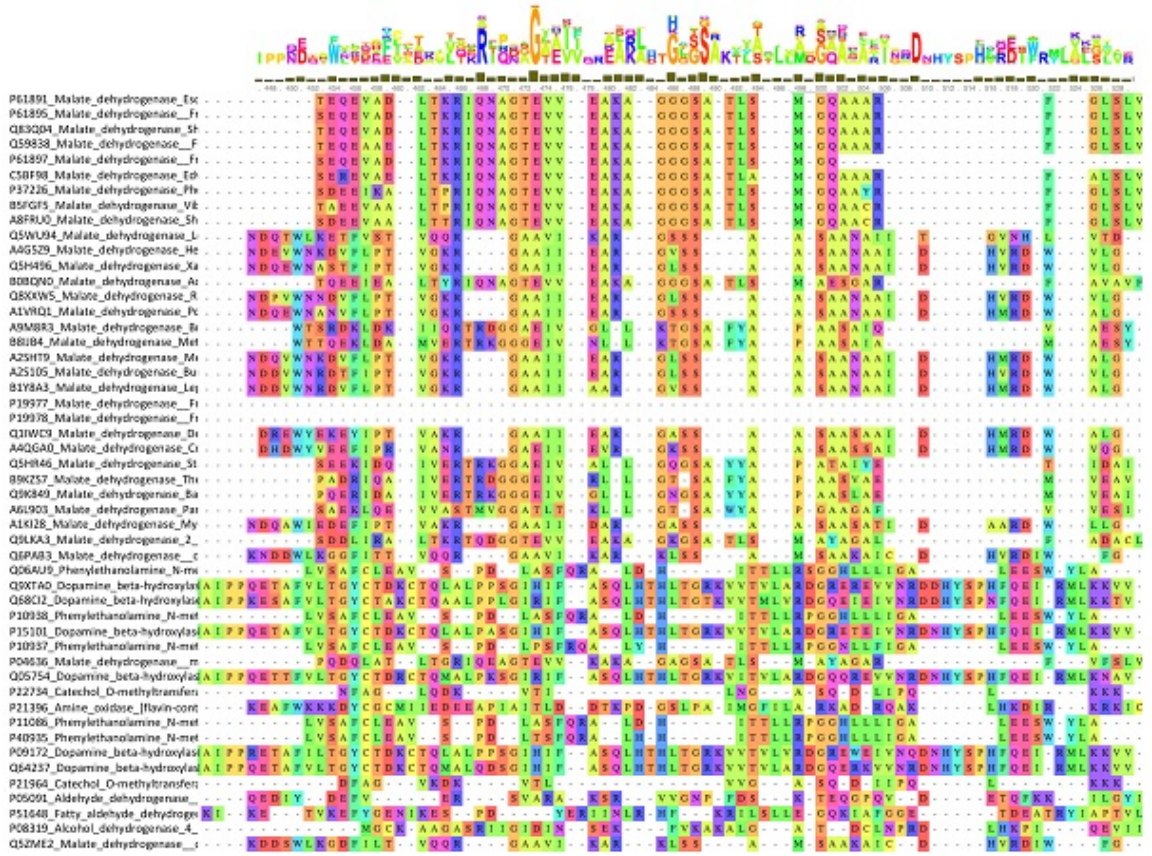


Figure 1.3 **Selenzyme result for EC 1.1.1.37** Detail of a multiple sequence alignment displayed by Selenzyme for the example given in Table 1 for EC 1.1.1.37

1.4 Pathway Design

1.4.1 Introduction to Retrosynthesis

This protocol considers the scenario where a desirable product is known, but no natural pathway leading to it is known. In such a case, we use retrosynthesis to discover possible pathways. Such algorithm generates networks linking the target compounds that one desires to bioproduce (the source) and the metabolites of the chassis strain (the sink) by applying chemical rules. These networks are then processed to extract biologically relevant information. For example, pathways can be enumerated [210] and ranked based on several criteria including enzyme availability and performance, product and intermediate compound toxicities [211], or the theoretical yield of the desired compound [187, 188, 212, 190, 213]. Users of retrosynthesis-based solutions often face a challenge common to most of these tools: the algorithms and the underlying data are often not fully documented and released. In most cases, authors provide fine-tuned webservers [187, 113, 188, 189, 214]

filled with pregenerated data that focuses on some exemplar cases. On the contrary, [138] provide the open-source workflow RetroPath2.0 based on the KNIME analytics platform [197] that uses community nodes and is therefore fully modular and can easily be tuned to a user's need. We describe first a protocol based on the RetroPath2.0 workflow and the necessary steps to generate and use retrosynthesis for bioproduction. The tool is available at myExperiment.org along with a set of reaction rules and some classic metabolic engineering examples to test RetroPath2.0 features.

1.4.2 Protocol Description (RetroPath2.0)

Preparation steps

Sink and Source Definition The first step to generate a retrosynthesis map is to encode all compounds of interest in a format that will allow processing by the retrosynthesis algorithm. Source compounds are the compounds the workflow starts iterating upon (compounds one desires to produce) and the sink compounds are the compounds where the algorithm stops (either the metabolome of the chassis organism or compounds easily supplemented in the media).

1. Gather compounds from a whole-cell metabolic model of the chassis organism of interest. It must include structural data for compounds.
2. Filter out compounds with incomplete structure. They often stand to define a class of compounds and cannot be processed further.
3. Select the sink compounds: either the whole set of compounds from the chassis organism or a subset selected based on expert knowledge. One can, for example, remove compounds belonging to blocked pathways by performing a flux-balance analysis. Sink compounds can also include molecules easily supplemented in the media.
4. Choose source compounds that one desires to bioproduce and collect associated structural information in order for the algorithm to process them.

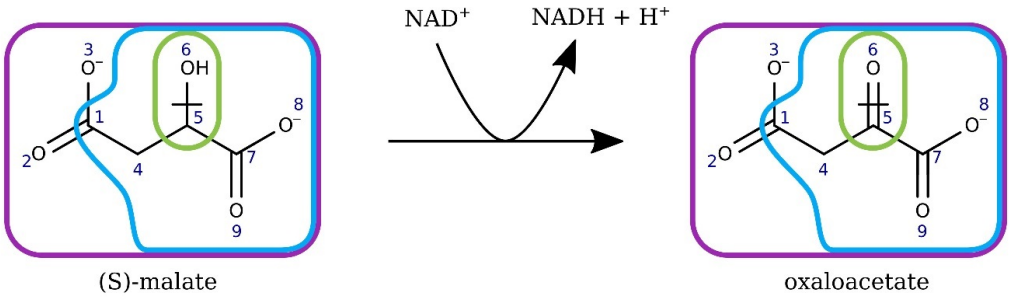
Reaction Rules The second step is to encode (bio)chemical reactions that will be used to perform the retrosynthesis. RetroPath2.0 uses reaction SMARTS to encode reactions. It is a SMIRK-like reaction rule [108] format defined by RDKit [194].

1. Select a database containing reaction information. Such database should at least provide the structure of each compound involved in a reaction.
2. Remove all reactions that do not modify the structure of a compound (transport reactions for example) and reactions involving incomplete structures (class of compounds, R-groups, etc.). Remove stereochemistry.
3. Perform an AAM to identify the reaction center (i.e., the part of the molecule that changes during the reaction). This AAM is also necessary later on to compute the reaction SMARTS.
4. Before building the monocomponent rules, one should consider the reactions in the reverse direction, i.e., consider the natural products as substrates and natural substrates as products. This is only needed for a retrosynthetic usage. Decompose multiple substrate reactions into components. There are as many components as there are substrates and each component gives the transformation between one substrate and the products. Each product must contain at least one atom from the substrate according to the AAM. This strategy enforces that only one substrate can differ at a time from the substrates of the reference reaction when applying the rule. Cosubstrates and coproducts that are currency cofactors (such as water, CO2, adenosine triphosphate (ATP), Nicotinamide Adenine Dinucleotide Phosphate (NADP), etc.) can be ignored from the rules under the assumptions that they are available in the cell and that there is no gain for retrosynthesis analysis in modeling promiscuity on these compounds.
5. Compute reaction rules as reaction SMARTS for each component. Do it for varying diameters around the reaction center (2–16 in RetroPath2.0) by removing from the reaction components all atoms that were not in the spheres around the reaction center atoms (Figure 1.4).

Building a Retrosynthesis Network

Once the user has performed the preparation steps, the rules, sink, and source are provided as inputs to RetroPath2.0 that builds the retrosynthesis network. The following section describes the steps followed by the algorithm. It will help highlighting key tunable parameters for the advanced user: scoring of the enzymes, number of compounds kept for the next iteration, maximum number of steps, as well as the role of the reaction diameter.

A



B

D = ∞ `[#8:6]-[#6:5](-[#6:4]-[#6:1](-[#8-:3])=[O:2])-[#6:7](-[#8-:8])=[O:9] >>[#8-:3]-[#6:1](=[O:2])-[#6:4]-[#6:5](=[O:6])-[#6:7](-[#8-:8])=[O:9]`

D = 4 `[#6:1]-[#6:4]-[#6:5](-[#8:6])-[#6:7](-[#8-:8])=[O:9] >>[#6:1]-[#6:4]-[#6:5](=[O:6])-[#6:7](-[#8-:8])=[O:9]`

D = 0 `[#6:5]-[#8:6]>>[#6:5]=[O:6]`

Figure 1.4 **Rules and SMARTS for reaction 1.1.1.37** Retropath 2.0 rules and corresponding SMARTS for reaction 1.1.1.37 at various diameters. **A** Reaction 1.1.1.37 with atom mapping. The level of promiscuity is modeled by the diameter (∞ purple, 4 blue, or 0 green), i.e., the number of bonds to consider around the reaction center (atoms 5 and 6), defining the atoms kept in the rules. **B** The SMARTS corresponding to the reaction rules at the various diameters. For readability, cofactors are not considered.

General Workflow Presentation The RetroPath2.0 workflow essentially follows the algorithm described below, proceeding in an iterative manner. For the sake of clarity, we present here the algorithm for one initial source compound, but it can process multiple compounds in parallel.

1. Verify the source compound is not already present in the sink.
2. Start iteration by applying the rules to each compound of the source set. The workflow starts by applying rules at the biggest diameter allowed by the user, typically d = 16. If no product is formed, the diameter is decreased to the available diameter value just below, allowing matching for shorter fragments and therefore more promiscuity. This step is repeated until some products are formed or until all possible diameters have been exhausted. For each compound, the products are computed using the RDKit KNIME nodes [194]. Products are standardized and duplicates are merged. All substrate-product pairs are added to the growing network along with the reaction rules linking them.
3. (Optional) Products can be scored according to a rule prioritization scheme. The one used in RetroPath2.0 will be briefly described below. Products are ranked according to the score of the rule used to produce them at the current iteration.

4. In the next iteration, the set of products becomes the new source set. However, before iterating, the workflow removes from the new source set all compounds that belong to the sink (as these are already solutions and there is no need to iterate) and the workflow adds the product set to the sink in order to avoid applying reactions on the same products during subsequent iterations.
5. Apply steps 2–4 either a predefined number of iterations or until the source set is empty.
6. The final produced graph is composed of the list of links between substrates and products annotated with their corresponding reaction rule. Products belonging to the initial sink set are annotated as such.

Remark: Rule Scoring by Enzyme Sequence Consistency As mentioned in the optional step 3 of the algorithm, products can be scored according to a rule prioritization scheme. In RetroPath2.0, this is done by scoring the confidence one can have that a rule is indeed linked to a trustworthy enzymatic sequence. More specialized scoring strategies are possible depending on the selected enzyme sequence. A description of the Selenzyme protocol can be found in this chapter.

Pathway Enumeration Between Two Pools of Compounds

Computing the Scope The aim is to compute the metabolic scope connecting the source compounds to the sink compounds, i.e., the set of compounds and reactions that are involved in at least one pathway.

1. Forward search: starting from source compounds, find all reachable compounds. A compound is considered as reachable whenever it is in the initial sink or if it is producible using a fireable reaction.
2. Backward search: starting from the sink compounds, add to the scope all reactions that can be fired, as long as the reaction substrates are all reachable.
3. (Optional) Simplify the scope by only keeping the shortest paths from the compounds of interest to the sink.
4. Visualize and explore the scope thanks to ScopeViewer, a web application provided in [138].

Enumerating Pathways For each source compound, proceed to the following steps to find pathways producing this source compound.

1. Build the stoichiometric matrix. The stoichiometric matrix describes the directed subnetwork involving the set of compounds and reactions identified at the scope step, starting from the source compound of interest.
2. Enumerate elementary flux modes. An elementary mode corresponds to a minimal unique set of reactions that (i) verified the stoichiometric constraints of the network and (ii) is able to carry nonzero fluxes at the system′s steady state [215]. Only enumerated flux modes linking source compounds to the sink compound are kept in order to form the final list of pathways.
3. Enumerate the pathways from the elementary flux modes. We provide at https://github.com/brsynth/rp2paths a separate utility program "RP2paths" allowing one to enumerate pathways from the results generated by RetroPath2.0.

1.4.3 Use case: 1,4-Butanediol Pathways Prediction Using RetroPath2.0

1,4-Butanediol is an important commodity chemical used as a starting point for the synthesis of other chemicals and polymers such as the polybutylene terephthalate, a unique engineering plastic. While most of the production of 1,4-butanediol is performed by chemical synthesis and is still making use of petroleum-based feedstock, a bioprocess alternative has been first reported in [191].
Here we showcase the usefulness of RetroPath2.0 in order to predict pathways enabling the bioproduction of 1,4-butanediol in *E. coli*.

Materials and Methods

To build the retrosynthetic graph and enumerate the pathways, we use the procedure below:

- 1,4-Butanediol is considered as the source compound.
- Sink compounds were extracted from the iJO1366 *E. coli* whole-cell model [216] and MetaNetX cross-references [202].
- Reaction information is collected from the MetaNetX database and reaction rules are generated following the aforementioned guidelines. The generated set contains from 6900 to 19,000 unique rules, depending on the diameter considered.

- RetroPath2.0 is applied with a maximum of four retrosynthesis iterations, keeping a maximum of 100 compounds for the next iteration, and reaction diameters ranging from 12 to 8.
- Pathways are enumerated thanks to the RP2paths software.

Using RetroPath2.0 we successfully retrieved the bioproduction pathway reported by [191], as well as five alternatives enabling the production of 1,4-Butanediol diacrylate (BDA) (depicted in Figure 1.5). Four pathways, including the one reported by [191], propose a common strategy where the coenzyme A to be attached to 4-hydroxybutyrate is supplied by different CoA-related chemicals (namely CoA, acetyl-CoA, succinyl-CoA, and butanoyl-CoA).

1.4.4 Other applications of the Protocol

The choice of rule sets, sink, and source depends on the application.

1. For instance, if one wishes to find all producing pathways for a given compound, the source will be the target, the sink the metabolites of the chassis strain, and the rules the reversed form of all known metabolic reactions, as was presented here.
2. To degrade a given xenobiotic, the rule set can be the same metabolic reactions in the forward direction, the sink will be the metabolites of the chassis strain, and the source is the compound to degrade.
3. Another choice of settings allows for another application of interest: choosing a set of known detectable compounds as sink, a set of target compounds one wishes to detect as source, and the set of forward rules, one can design sensing-enabling pathways [77].

The versatility of this tool allows us to use it for various applications, as showcased in Chapter 2 [170] .

1. Using a set of rules for isomer generation or fragment exchange, instead of enzymatic rules, we can generate all structural isomers of a given compound or virtually screen the chemical space around a set of molecules of interest.
2. Moreover, the way we encode our rules using diameter allows us to tune the promiscuity we allow in our compounds′ generation. Therefore, this workflow can be used for metabolome completion; with novel molecules generated using promiscuous enzymatic reaction rules.

1.5 Summary and conclusion

We have covered in this chapter some possible scenarios involving enzyme discovery and pathway design for synthetic biology and metabolic engineering applications. For a given biochemical transformation, we can select enzymes sequences through the Selenzyme [175] protocol. Starting from a desired target compound, we can identify production pathways through the RetroPath2.0 [138] protocol. As shown in Figure 1.1, such discovery process can be in some cases reversed, for instance when an interesting pathway is known, but not its product. A tool such as antiSMASH [177] allows for the annotation of the transformation steps involved in the pathway starting from the gene cluster, as well as allowing integration of predicted biosynthesis pathways for secondary metabolites with genome-scale models of metabolism. Once a putative biochemical pathway has been inferred from a gene cluster, antiSMASH and RetroPath provide means to predict and enumerate its main products and side products of enzyme promiscuity. The tools and protocols described in this chapter and the resulting integrated pipeline offer a rich synthetic biology toolbox for enzyme discovery.

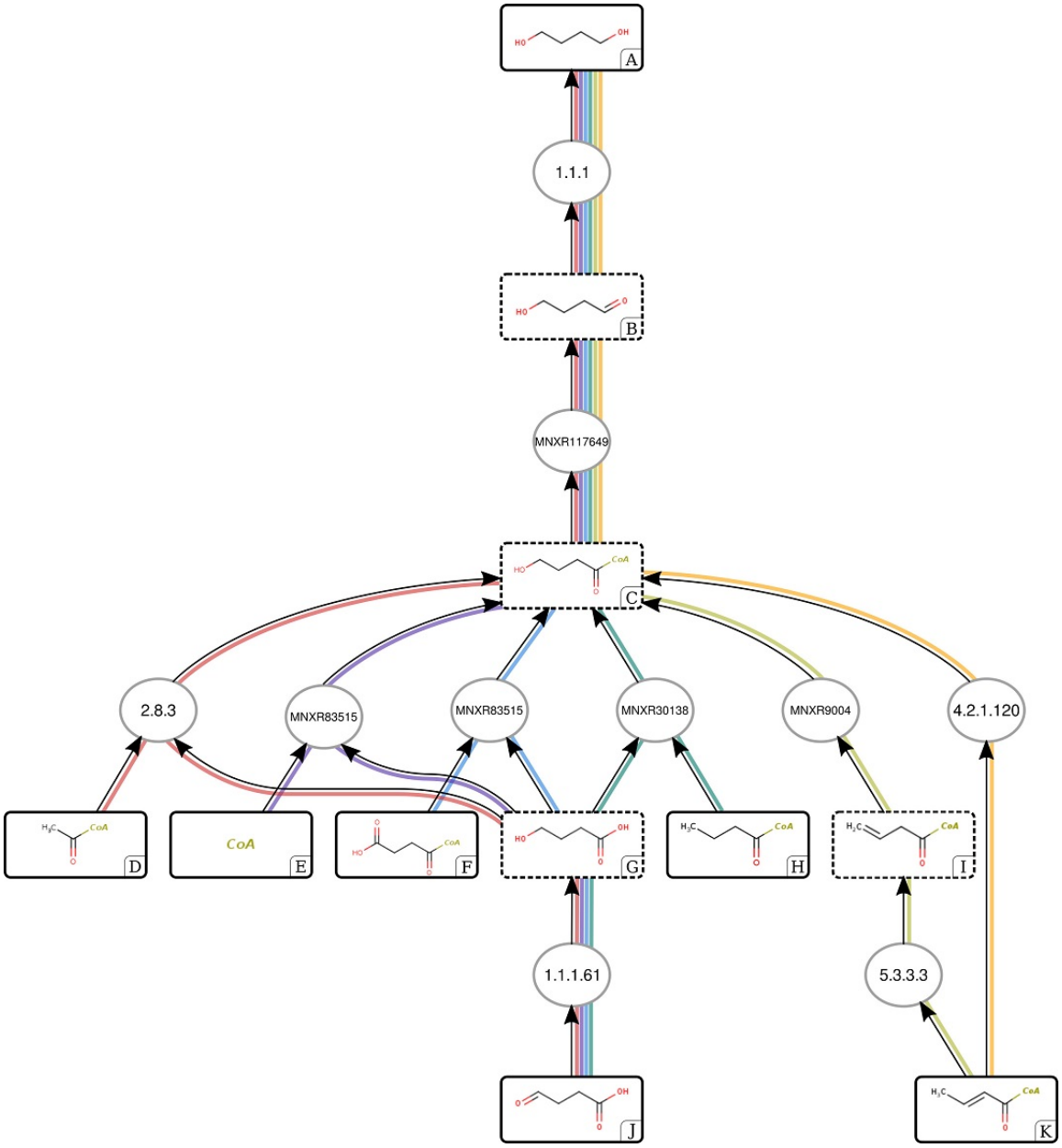


Figure 1.5 **Enumerated pathways for 1,4-Butanediol production**. Each pathway is depicted by a distinct color. Pathway referenced in Yim *et al.* (2011) is in red (J-G-D-C-B-A). Compounds are represented by their structures, and reactions by their EC numbers when known (else by the MetaNetX reaction ID). 1,4-Butanediol and sink compounds are surrounded by a solid line, intermediates by a dashed line. A, 1,4-butanediol; B, 4-hydroxybutanal; C,4-hydroxybutyryl-CoA; D, acetyl-CoA; E, CoA; F, succinyl-CoA; G, 4-hydroxybutyrate; H, butanoyl-CoA; I, vinylacetyl-CoA; J, succinate semialdehyde; K, crotonoyl-CoA. Cofactors have been removed for clarity.

2

Molecular structures enumeration and virtual screening in the chemical space with RetroPath2.0

This work was published in Journal of Cheminformatics by Mathilde Koch, Thomas Duigou, Pablo Carbonell and Jean-Loup Faulon.
Only minor modifications to the published article have been introduced in the Chapter below.

Detailed contribution to this thesis

The aim of this Chapter is to present other applications of retrosynthesis tools, and especially of the chemical rule encoding used in the team. While my work involved mainly the mathematical proof behind the isomer enumeration algorithm presented in this Chapter, presenting various uses of such algorithms and reaction representation supports our use of the same reaction formalism in the next Chapter.

Full reference

Koch M., Duigou T., Carbonell P., Faulon J-L. (2017) Molecular structures enumeration and virtual screening in the chemical space with RetroPath2.0 *Journal of Cheminformatics*, 10.1186/s13321-017-0252-9.

Contributions as stated in the article

The work was directed by J.-L.F. who sketched the proof of the claim, designed all workflows, and generated the results presented in the isomer enumeration and metabolome completion and metabolomics sections. M.K.

wrote the proofs in the isomer enumeration section, T.D. wrote all workflows and produced the results in the section search for molecules maximizing biological activities. P.C. used the workflows to generate the results in the section virtual screening in the chemical space. All authors contributed to the manuscript write-up.

Abstract

Background: Network generation tools coupled with chemical reaction rules have been mainly developed for synthesis planning and more recently for metabolic engineering. Using the same core algorithm, these tools apply a set of rules to a source set of compounds, stopping when a sink set of compounds has been produced. When using the appropriate sink, source and rules, this core algorithm can be used for a variety of applications beyond those it has been developed for.
Results: Here, we showcase the use of the open source workflow RetroPath2.0. First, we mathematically prove that we can generate all structural isomers of a molecule using a reduced set of reaction rules. We then use this enumeration strategy to screen the chemical space around a set of monomers and predict their glass transition temperatures, as well as around aminoglycosides to search structures maximizing antibacterial activity. We also perform a screening around aminoglycosides with enzymatic reaction rules to ensure biosynthetic accessibility. We finally use our workflow on an *E. coli* model to complete *E. coli* metabolome, with novel molecules generated using promiscuous enzymatic reaction rules. These novel molecules are searched on the MS spectra of an *E. coli* cell lysate interfacing our workflow with OpenMS through the KNIME Analytics Platform.
Conclusion: We provide an easy to use and modify, modular, and open-source workflow. We demonstrate its versatility through a variety of use cases including molecular structure enumeration, virtual screening in the chemical space, and metabolome completion. Because it is open source and freely available on MyExperiment.org, workflow community contributions should likely expand further the features of the tool, even beyond the use cases presented in the paper.

2.1 Introduction

The number of known chemical reactions is huge, at the time this manuscript was written there were $\simeq$ 84 million single- and multi-step reactions in the

Chemical Abstract Service (CAS) database [217]. Yet, many reactions in CAS are redundant because the same reactions are applied to different reactants. Identifying identical reactions can be performed by computing reaction rules. Reaction rules represent reactions at the reaction center only. In other words, a reaction rule comprises only the substructures of the reactants and the products for which the atoms are either directly involved in bond rearrangements or are deemed to be essential for the reactivity of the reaction center. While a set of reaction rules is of course not available for all known chemical reactions, rules have been compiled for focused applications, such as retrosynthesis planning [218, 219], the discovery of novel chemical entities in medicinal chemistry [220], xenobiotic (including drug) degradation [185], metabolomics [214], and metabolic engineering [188, 136, 191, 190]. Depending on the application, the number of rules varies from less than one hundred to few thousands, but in all cases the number of known reactions per application far exceeds the number of rules (there are for instance more than 14,000 reactions in metabolic databases such as MetaNetX [202]). There are several ways of coding reaction rules (for instance, BE-matrices [221] and fingerprints [112]) but most of the time the rules can be represented by reaction SMARTS [108], as it is done in the current paper.
The purpose of reactions rules is to generate reaction networks. The rules can be used in a forward manner to find for instance the metabolic degradation products of a drug, or in a reverse manner to find the reactions producing a desired product from a set of available reactants. In this later usage one produces retrosynthesis reaction networks. Several tools have been developed in the past to generate (retrosynthesis) reaction networks and reviews are available for synthesis planning [218, 219], and for metabolic engineering [182]. Disregarding if the rules are applied in a forward or reverse manner, network generation tools are making use of the same core algorithm. Starting from a source set of compounds the core algorithm applies the rules in an iterative fashion either a predefined number of times or until a sink set of compounds has been produced. At each iteration, the algorithm fires the rules on the source set producing new molecular structures and determines the new source set of molecules the rules will be fired upon at the next iteration. That set must comprise molecules that have not been processed before. Further details on the core algorithm and the differences between the various implementations are provided in [222] and [138]. In the current paper we make use of an open source workflow (RetroPath2.0 [138]), which follows the above core algorithm. This workflow is not based on original codes but instead was constructed entirely by assembling KNIME nodes [223] developed by the cheminformatics community (primarily RDKit nodes [194]). RetroPath2.0 is the first open source release of a retrosynthesis reaction network generation, and its usage in the current paper beyond network

generation demonstrates its versatility.

As already mentioned, reaction network generation tools coupled with reactions rules have been developed and used primarily for synthesis planning and metabolic engineering, but can they be used to enumerate molecules (isomers for instance) and more generally to search chemical structures in the chemical space?

In principle yes if one can devise reaction rules enabling the production of any molecule in the chemical space. Such a set of rules necessarily exists for all known molecules (such as those in the CAS database) since they have been produced through either natural or synthetic chemical reactions. In practice and as already stated, reaction rules so far developed are application limited. Yet, within their respective application fields, specific rules have been used to discover novel molecules and reaction pathways. Taking experimentally validated examples, the rules associated with the ligand-based de novo design software Design of Genuine Structures (DOGS) (inSili.com LLC) [220] have enabled the production of new chemical entities inhibitors of DAPK3 (death-associated protein kinase 3) [224], metabolic rules for promiscuous enzymes have allowed the discovery of novel metabolites in *E. coli* [181] and have also been used to engineer metabolic pathways producing 1,4-butanediol [191] and flavonoids [225].

Going beyond application limited reaction rules, the main contribution of the present paper is to propose a set of transformation rules that enables the generation of any isomer of any given molecule of the chemical space. Precisely, we prove the claim that any isomer of any given molecule of N atoms, can be reached applying at most $O(N^2)$ rules.

As illustrations, our transformation rules are used to screen the chemical space for structures that are similar to a given set of well-known monomers and to search aminoglycosides structures maximizing antibacterial activities. The compounds produced by our rules are not necessarily chemically accessible, since our transformation rules are not constructed based on chemical synthesis schema. To probe the (bio)synthetic accessibility of our solutions, we also perform search in the (bio)chemical space using enzymatic reaction rules. The enzymatic rules are also used to propose novel molecules completing *E. coli* metabolic network and for which masses are found in cell lysate mass spectra. All results presented in this paper have been produced making use of the open source workflow RetroPath2.0. RetroPath2.0 and the associated data are provided as Supplementary and can be downloaded at MyExperiment.org . The only differences between the various usages we have made of the RetroPath2.0 are within 1) the set of reaction rules and 2) the way molecules are selected at each iteration during the network generation process.

2.2 Results and Discussion

The purpose of this section is to showcase the versatility of RetroPath2.0 by taking use cases of interest to the community. We first propose reaction rules to enumerate isomers (section isomer enumeration), we then use the rules to screen in the chemical space structures that are similar to some known monomers (section virtual screening) and compute property distribution (Glass transition temperature) in both the Chemical Space and PubChem, we next use a Quantitative Structure Activity Relationship (QSAR) to search aminoglycosides types molecules for which antibacterial activity is maximized using both isomer transformation rules and enzymatic rules (section search for molecules maximizing biological activities), and we finally use enzymatic rules to find novel metabolites in *E. coli* and annotate the Mass Spectrometry (MS) spectra of an *E. coli* cell lysate interfacing RetroPath2.0 with OpenMS [226] (section metabolome completion and metabolomics).

2.2.1 Isomer enumeration

Isomer enumeration is a long-standing problem that is still under scrutiny [227, 228]. Our intent here is not to provide the fastest enumeration algorithm but to demonstrate how RetroPath2.0 can perform that job once appropriate reaction rules are provided. However, we provide in Figure 2.1 a comparison of RetroPath2.0's execution time with the OMG and PMG software tools [228, 229] specifically dedicated to isomers enumeration. RetroPath2.0 is found faster than OMG but slower than PMG. Thereafter, we outline two approaches making use of RetroPath2.0. The first is based on the classical canonical augmentation algorithm [230] and the second consists of iteratively transforming a given molecule such that all its isomers are produced. We name this latter approach isomer transformation. In both cases we limit ourselves to structural (constitutional) isomers, as there already exist workflows to enumerate stereoisomers [231].

Canonical augmentation

The principle of canonical augmentation, which is an orderly enumeration algorithm, is to grow a molecular graph by adding one atom at a time and retaining only canonical graphs for the next iteration [230]. The algorithm first proposed by Brendan McKay has been used to generate the GDB-17 database of small molecules [232]. The original algorithm has also been mod-

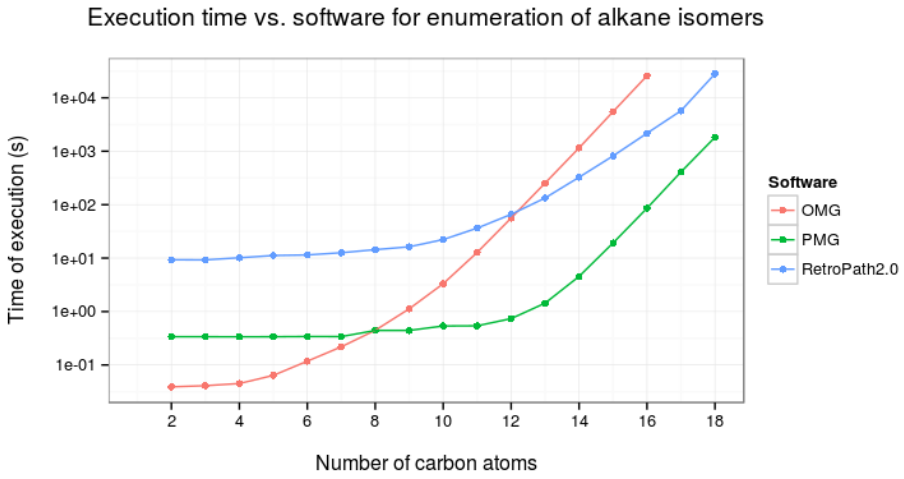


Figure 2.1 **Execution time for each tested software on the enumeration of alkane isomers.**

ified such that at each step a bond (not an atom) is added to the growing molecules [228]. In the present implementation we use the original McKay algorithm [230], consequently, the number of iterations is the number of atoms one wishes the molecule to have. The algorithm can easily be implemented into RetroPath2.0 by choosing as a source set a single unbonded atom, and a rule set depicting all possible ways an atom can be added to a molecular graph (see method section for more information). Considering that an atom can be added to a growing molecule through one, two, or more bonds (depending on its valence), the set of reaction rules is straightforward however cumbersome if one starts to consider all possible atoms types. For this reason we limit ourselves to carbon skeleton as it is usually done in the first step of isomer enumeration algorithm. Figure 2.2 below depicts the set of rules that generate all triangle free carbon skeletons.

We note that rules R_2 to R_4 will generate cycles since the added atom is attached to the growing molecule by 2 to 4 four bonds, thus only rule R_1 is necessary to grow acyclic molecules (alkanes for instance). Table 2.1 provides the numbers of structural isomers of alkanes found up to 18 carbon atoms running RetroPath2.0 with rule number 1 in Figure 2.2.

Isomer transformation

The isomer canonical augmentation algorithm becomes more complex when one starts to consider different atom and bond types. To overcome these difficulties the idea of the transformation enumeration approach is to start with

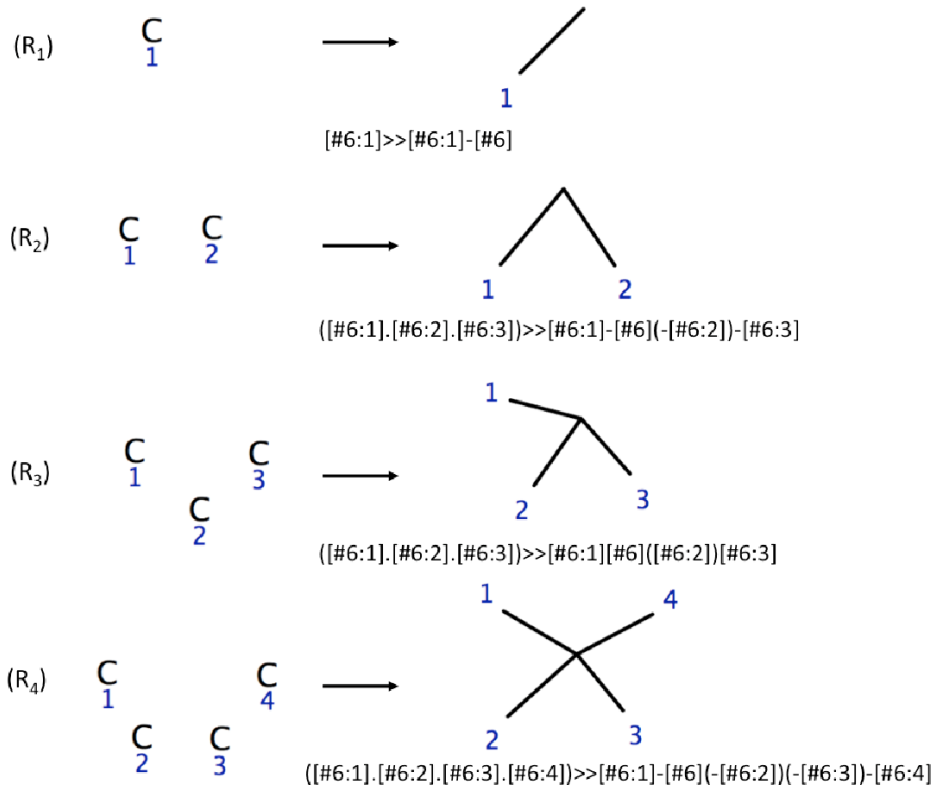


Figure 2.2 **Reaction rules for canonical augmentation of carbon skeletons**. The corresponding reaction SMARTS string is provided for each rule.

one fully-grown molecule to which one applies all possible transformations such that all the structural isomers of the initial molecule are generated. This approach can be implemented in RetroPath2.0 using a hydrogen saturated molecule as a source and a reaction rule set enabling to transform the molecule while keeping the correct valence for each atom. Because atom valences are maintained the total number of bonds must remain the same after the transformations have taken place. In order to maintain the number of bonds constant, for any reaction rule the number of bonds created must equal the number of bonds deleted.

RetroPath2.0 applies a reaction rule to a given molecule by first searching all occurrences in the molecule of the sub-graph representing the reactant (left side of the rule). To this end the labels on the sub-graph are removed. Then for each occurrence of the unlabeled sub-graph in the molecule, the labels are restored and the bonding patterns on the molecule are changed accordingly. The process is illustrated in the Figure 2.3 where it can be seen that rules R_a and R_b are identical (i.e. they produce the same solutions). In general, two rules $R_a = (L_a, A)$ and $R_b = (L_b, B)$ will produce the same solutions if a one-to-one mapping π can be found between the labels L_a and L_b of the rules such that the set of edges (A) in R_a is transformed by π into the edges (B) of R_b, i.e. $\pi(A) = B$.

Nbr of carbon atoms	Nbr of structures output by canonical augmentation algorithm	Nbr of structures output by isomer transformation algorithm	Nbr of iterations for isomer transformation algorithm
1	1	1	1
2	2	1	1
3	3	1	1
4	5	2	2
5	8	3	3
6	13	5	3
7	22	9	4
8	40	18	5
9	75	35	5
10	150	75	6
11	309	159	7
12	664	355	7
13	1466	802	8
14	3324	1858	9
15	7671	4347	9
16	18030	10359	10
17	42924	24894	10
18	103447	60523	10

Table 2.1 **Number of generated alkane isomers by canonical augmentation algorithm and isomer transformation algorithm.** The numbers agree with earlier calculations [233]. For a given number of carbon atoms (N), the canonical augmentation generates all alkanes from 1 to N carbon atoms, while the isomer transformation enumeration generates alkanes having only N carbon atoms, one can thus verify that at any given number of carbon atoms N, the numbers of structures generated by the canonical augmentation algorithm equals the sum of numbers of isomers generated by the transformation algorithm up to N.

Claim

The 19 rules described in Figure 2.4 allow us to generate all isomers of a given molecule at most $3/4 * (N^{2\smile}N)$ iterations, where N is the number of atoms, respecting the following constraints: the maximal valence is 4 and there cannot be two double bonds on the same atom in a 3 or 4 membered ring.

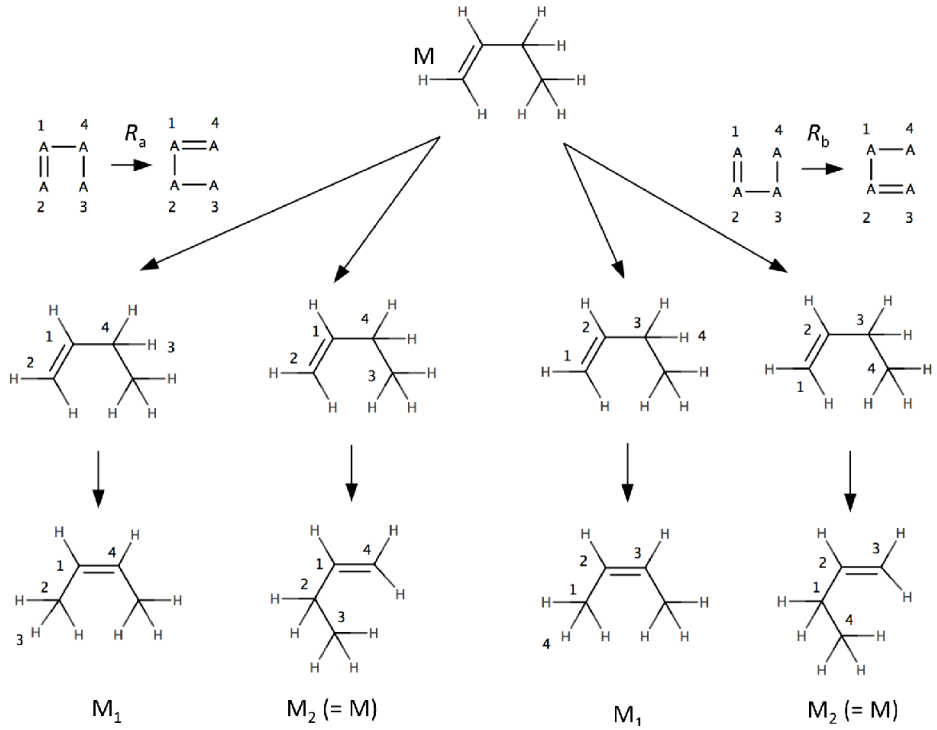


Figure 2.3 **Identical rules**. There are two different ways (two different possible matchings for the reactants of the rules) of applying rules R_a and R_b, each rule produces molecules M_1 and M_2. The molecules produced by R_a are identical to those produced by R_b because the rules are identical. R_a is identical to R_b because when applying the one-to-one label mapping $\pi(1,2,3,4) = 2,1,4,3$ on the edges of the R_a one obtains the edges of R_b.

Lemma 1

The minimal number of bonds one can change is 4 and the 19 rules described in Figure 2.4 generate all minimal transformations respecting the following constraints: the maximal valence is 4 and there cannot be two double bonds on the same atom in a 3 or 4 membered ring.

Proof: The minimal transformation one can perform consists of deleting one bond and creating another one. Since the bond created must be different from the one deleted at least three atoms (A_1, A_2, A_3) must be involved. Let a_{12}, a_{13}, and a_{23} be the bond orders between the three atoms and let b_{12}, b_{13}, and b_{23} the bond orders after the reaction has taken place. Because the atom valence is maintained the following system of equations holds:

$$\begin{aligned} (L1) a_{12} + a_{13} &= b_{12} + b_{13} \\ (L2) a_{12} + a_{23} &= b_{12} + b_{23} \\ (L3) a_{13} + a_{23} &= b_{13} + b_{23} \end{aligned} \tag{2.1}$$

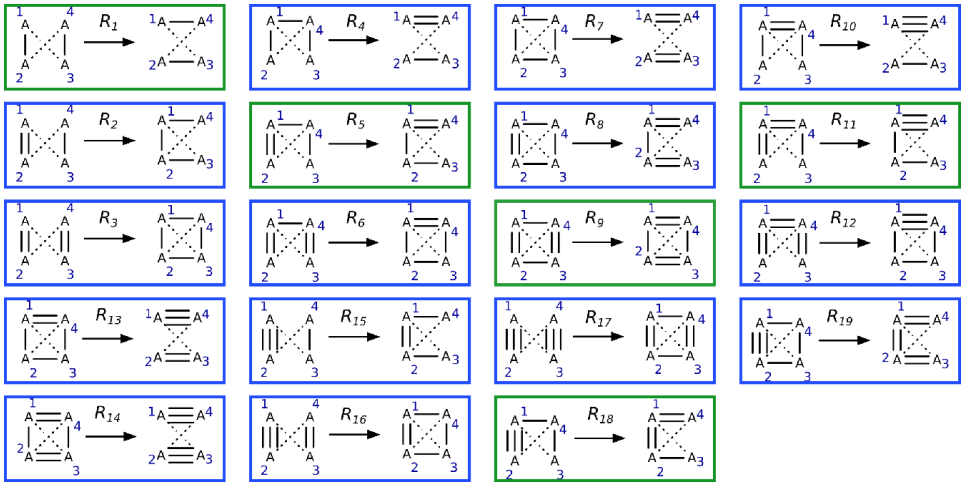


Figure 2.4 **Isomer transformation rule set**. All reactions rules are solutions of system of equations (2) and are not identical (see text and Figure 2.3 for definition of identical rules). Reactions in green move bonds around without creating or deleting cycles. Reactions in blue change bond order by creating or deleting at least one cycle. To each reaction corresponds a reverse reaction. The reverse reaction of R_1 is R_1, for R_2 it is R_4, for R_3 : R_7, for R_5: R_5, for R_6 : R_8 and the reverse reaction of R_9 is R_9. The reverse reaction for R_{10} is R_{15}, for R_{11}, R_{18}, for R_{12}, R_{19}, for R_{13}, R_{16} and for R_{14}, R_{17}. The bond order a_{13} and a_{24} can take any value from 0 to 3. The full list of rules excluding triple bonds can be found in Figure 2.10.

$(L1) + (L2) - (L3) \implies a_{12} = b_{12}$, which implies $a_{23} = b_{23}$ and $a_{13} = b_{13}$. It is therefore impossible to proceed to a minimal transformation with only 3 bonds involved. Let us consider 4 atoms. There are 6 possible bonds between those atoms. Let us consider that we are changing 4 bonds, since we aim to find minimal transformations. Let us call a_{13} and a_{24} the two fixed bonds, without loss of generality. Valence conservation (with $b_{13} = a_{13}$ and $b_{24} = a_{24}$) gives us the following system:

$$\begin{aligned}
(L1) a_{12} + a_{14} &= b_{12} + b_{14} \\
(L2) a_{12} + a_{23} &= b_{12} + b_{23} \\
(L3) a_{23} + a_{34} &= b_{23} + b_{34} \\
(L4) a_{14} + a_{34} &= b_{14} + b_{34}
\end{aligned} \tag{2.2}$$

We can notice that $(L1) + (L3) = (L2) + (L4)$: we therefore have a system of 3 equations with 4 unknowns, so we can set an unknown and calculate the other solutions.

As we are looking for minimal transformations, we can assume that we are changing a bond order by 1 on this unknown that we can set. Since valence is conserved, if a bond order is increased, then a bond order from the same atom has to be decreased. As the problem is perfectly symmetrical in all variables at this point, we can thus assume without loss of generality (at least

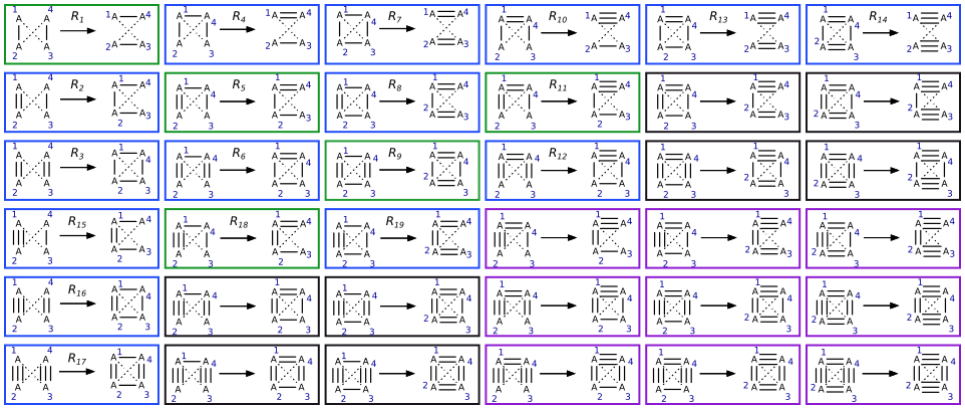


Figure 2.5 **Rules before solution space reduction due to valence and structure considerations**. Reactions in green move bonds around without creating or deleting cycles. Reactions in blue change bond order by creating or deleting at least one cycle. Reactions in purple were deleted because valence is limited to 4, and reactions in black were deleted because there cannot be two double bonds on the same atom in a 3 or 4 membered ring.

one bond has to be deleted) that $b_{12} = a_{12} - 1$. Then solving the system immediately gives us $b_{14} = a_{14} + 1$, $b_{23} = a_{23} + 1$ and $b_{34} = a_{34} - 1$. This system can only be solved in our case (positive bond orders, no quadruple bonds) if a_{14} and a_{23} are either 0, 1 or 2 and a_{12} and a_{34} are either 1, 2 or 3. This means that we have at most 81 (3^4) cases for initial bond orders where our isomer problem has a solution. However, this solution space can be further reduced by problem symmetry arguments. We can see that the roles of a_{12} and a_{34} are symmetrical, as well as the roles of a_{14} and a_{23}.
Let us call A_1 the atom with the highest considered sum of bound orders (neglecting the fixed orders a_{13} and a_{24}.). Therefore, it is such that

$a_{12} + a_{14} \geq a_{12} + a_{23}$ (higher considered sum of bound orders than A_2), or $a_{14} \geq a_{23}$ (Condition 1) $a_{12} + a_{14} \geq a_{14} + a_{34}$ (higher considered sum of bound orders than A_4), or $a_{12} \geq a_{34}$ (Condition 2) $a_{12} + a_{14} \geq a_{23} + a_{34}$ (higher considered sum of bound orders than A_3), is automatically verified when the other two are verified.
Condition 1 is not respected when $a_{23} = 2$ and $a_{14} = 0 or 1$ or when $a_{23} = 1$ and $a_{14} = 0$, without constraints on a_{12} and a_{34}: (2+1) * 9 = 27 solutions. For the same reason, 27 solutions do not respect condition 2. The solutions that do not respect both condition 1 and condition 2 are (2+1) * (2+1) = 9. By symmetry arguments, we therefore reduced the solution space from 81 to 81 - 27 - 27 + 9 = 36. These 36 reaction rules are presented in Figure 2.5.
We can further reduce the solution space by considering that the maximum atom valence is 4. The solutions that do not respect this constraint are such that $a_{12} + a_{14} = 5$, so $a_{12} = 3$ and $a_{14} = 2$ (and this automatically verifies

conditions 1 and 2). Since there are no constraints on a_{23} and a_{34}, we have 9 such solutions: the solution space has been reduced to 36 - 9 = 27 reactions. One more constraint, imposed by 3D conformation of the molecule, is that there cannot be two double bonds on the same atom in a 3 or 4 membered ring. This must be true for our initial molecule as well as for the produced molecule. For the initial molecule (as can be seen in the 4 black rules under rule 13 in Figure 2.5, when $a_{12} = a_{14} = 2$, since $a_{34} > 1$ (bond whose order will be reduced), there is a cycle if $a_{23}! = 0$. There are therefore 4 solutions where the initial molecule is invalid: when $a_{12} = a_{14} = 2$, and a_{34} is 1 or 2 (smaller than a_{12}) and a_{23} is 1 or 2.
This must also be true for the produced molecule. Two double bonds will be produced around atom 1 with $a_{12} = 3$ and $a_{14} = 1$ (this can be seen in the 4 black rules under rule 18 in Figure 2.5). There will be a cycle if $a_{34}! = 0$. There are therefore 4 solutions where the produced molecule is invalid: when $a_{12} = 3$,$a_{14} = 1$, and a_{34} is 2 or 3 and a_{23} is 0 or 1 (smaller than a_{14}).
Since these solutions respect valence constraints and problem symmetry, they are not included in the previous solution space reductions and therefore the solution space is reduced to 27 - 8 = 19 solutions. A summary table of solution space reduction is given in Table 2.4. Since we have found 19 different working solutions for all the cases we have left, we have proved that the minimal number of bonds one can change is 4 and the 19 rules described in Figure 2.4 generate all these minimal transformations.

Lemma 2

Let us consider M_b an isomer of M_a. We can apply a rule from this set of 19 rules that will reduce the sum of absolute order differences between those two molecules by at least 2 and at most 4.

Proof: Let (a_{ij}) be the order of bonds in M_a, (b_{ij}), $j \in [2, N]$, $i \in [1, j-1]$, the order of bonds in M_b, where N is the number of atoms in M_a and M_b. Since M_b is different from M_a, we can find i, j such that $a_{ij} > b_{ij}$. By valence conservation in atom A_j, we can find k such that $a_{jk} < b_{jk}$, and by valence conservation of atom A_k, we can also find l such that $a_{kl} > b_{kl}$. Therefore, we are considering 4 atoms and 4 bonds between those atoms, with at least 3 of their orders changing by 1. According to Lemma 1 (2.2.1) the minimal number of bonds one can change is 4, so we will also have to change the bond order between A_i and A_l. We are therefore considering a minimal transformation, so we know thanks to Lemma 1 that we can apply a rule from our set of rules to generate that transformation. Let us call

Ma' the molecule produced that way, and a'_{ij} its bond orders. Let us now calculate the sum of orders of Ma'. Then, by applying the rule, we have $a'_{ij} = a_{ij} - 1$, and therefore $|b_{ij}{}^{\smile}a'_{ij}| = |b_{ij}{}^{\smile}a_{ij}| - 1$. For the same reason, $|b_{kl}{}^{\smile}a'_{kl}| = |b_{kl}{}^{\smile}a_{kl}| - 1$. Moreover, $a'_{jk} = a_{jk} + 1$, and since a_{jk} is smaller than b_{jk}, we also have $|b_{jk}{}^{\smile}a'_{jk}| = |b_{jk}{}^{\smile}a_{jk}| - 1$. The only bond we did not choose to change is a_{li}. The order $a_{li'}$ of the transformed bond is either closer to b_{li} than was a_{li}, then the difference of the sum of absolute order differences is reduced by 4, or is further from b_{li}, and this sum is reduced by 2. Therefore, if M_a and M_b are different, we can apply a rule from this set of rules that will decrease the sum of absolute order differences by at least 2 and at most 4.

Lemma 3

Considering M_a and M_b an isomer of M_a, the 19 rules described in Figure 2.4 allow us to transform M_a into M_b using at most $3/4 * (N^{2\smile}N)$ single transformations, where N is the number of atoms, respecting the following constraints: the maximal valence is 4 and there cannot be two double bonds on the same atom in a 3 or 4 membered ring.

Proof: Let us consider M_b an isomer of M_a. If the sum of absolute order differences is not null, then M_b is different from M_a and using Lemma 2 (2.2.1), we know we can apply a rule that will strictly decrease the sum of absolute order differences. This sum is obviously positive, is an integer, and is strictly decreasing each time we apply a transformation rule so it will converge to 0 in $S/2$ transformations at most, where S is the sum of absolute order differences between M_a and M_b. When this sum is null, all bond orders are the same, which means the molecules are the same. An upper estimation of the maximum bond order difference is obtained when M_a only has triple bonds, which all have to be deleted. In that case, the sum of absolute order differences is: $S = 3 * (N^2 - N)/2$, where N is the number of atoms and $(N^2 - N)/2$ the number of defined orders (since $a_{ij} = a_{ji}$). Therefore, since the sum decreases by at least 2, the maximum number of transformations we need to apply is $3 * (N^{2\smile}N)/4$.

Proof of the main claim: Given the workings of the algorithm (breadth-first, as explained in section 2.4.2), the number of iterations for generating all isomers is the number of iterations for generating the furthest one in term of bond order difference from our starting molecule. Therefore, applying Lemma 3 (2.2.1), we know the maximum number of iterations of the algorithm is $3 * (N^2 - N)/4$.

Notice that although the number of iterations of the algorithm scales $O(N^2)$, the number of transformation rules applied (i.e.: single reactions) is proportional to the number of isomers.

Corollary 1

The maximum number of iterations to generate all alkanes is $N-1$, where N is the number of carbon atoms (hydrogens are not considered here).

Proof: Adapting the demonstration of Lemma 3 (2.2.1), we have to consider the sum of absolute order differences of the farthest isomers that can be reached. Since alkanes are acyclic, the number of bonds is $N-1$ (proven by a simple recurrence, the new atom being joined at a single point to the chain since the molecule is acyclic). Therefore, considering all bonds are different in the new molecule, the sum of absolute order differences is at most $2(N-1)$. Therefore, the maximum number of iterations of the algorithm is $N-1$.
The isomer transformation algorithm was applied to generate all alkanes up to 18 carbon atoms using rule R_1 of Figure 2.4, since it is the only rule with only single bonds. Results are presented in Table 2.1, where it can be seen that Corollary 1 (2.2.1) is verified in practice.

2.2.2 Virtual screening in the chemical space

In this section we used RetroPath2.0 to search all molecules that are at predefined distances of a given set of molecules. Such queries are routinely carried out in large chemical databases for drug discovery purposes [234], but in the present case we search similar structures in the entire chemical space. To perform search in the chemical space, we used a source set composed of 158 well-known monomers having a molecular weight up to 200 Da. Our rule set included the transformations colored green in Figure 2.4 (i.e. transformation rules where double bonds are not transformed into cycles and conversely). For each monomer, RetroPath2.0 was iterated until no new isomers were generated. Each generated structures at a Tanimoto similarity greater than 0.5 from its corresponding monomer were retained (Tanimoto was computed using MACCS keys fingerprints [120]).
Next, we wanted to probe if the generated structures exhibited interesting properties as far as polymer properties are concerned. To that end we first developed a Quantitative Structure Property Relationship (QSPR) model taking properties from [235]. We focused on polymer glass transition temperature T_g data [236]. The QSPR model was based on a ran-

dom forest trained using RDKit fingerprints descriptors [194]. The obtained model had a leave-one-out cross-validation performance of $Q^2 = 0.75$. The model was then applied to predict the T_g for the set of enumerated isomers. Figure 2.7 compares the distribution of predicted T_g values for the enumerated isomers with those obtained from isomer structures available from PubChem. T_g values for enumerated isomers appeared evenly distributed around 301.86 ± 25.69 K compared with the isomers that were available in PubChem (331.66 ± 46.19 K). This shift in the T_g values could be explained by the difference in distribution that necessarily exists between the isomers that are present in PubChem and the total number of enumerated isomers. As we lower the Tanimoto threshold, some monomers might become underrepresented in terms of isomer availability in PubChem. Figure 2.6 shows the distributions of both sets of isomers in function of the threshold. The increased ability of selecting polymers with T_g above or below room temperature for the enumerated set compared with the PubChem isomers is a desirable feature, as this parameter will determine the mechanical properties of the polymer [237]. In that way, performing a virtual screening of the chemical space of isomers of the reference monomers opens the possibility to engineering applications with improved polymer design.

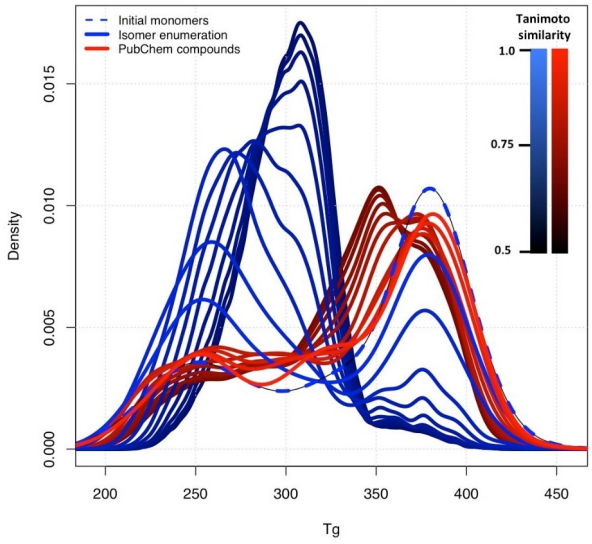


Figure 2.6 **Distributions of predicted T_g values for enumerated isomers and for isomers found in PubChem with varying Tanimoto threshold.**. Distribution of predicted polymer glass transition temperature T_g for enumerated isomers and for isomers found in PubChem of a reference set of 158 monomers with a Tanimoto similarity greater than a threshold varying between 0.5 and 1.

Moreover, we were interested in determining how many of the starting 158 monomers were accessible through biosynthesis. Namely, how many of the compounds can be synthesized by engineering a metabolic pathway in a chassis organism. This computation can be accomplished by RetroPath2.0

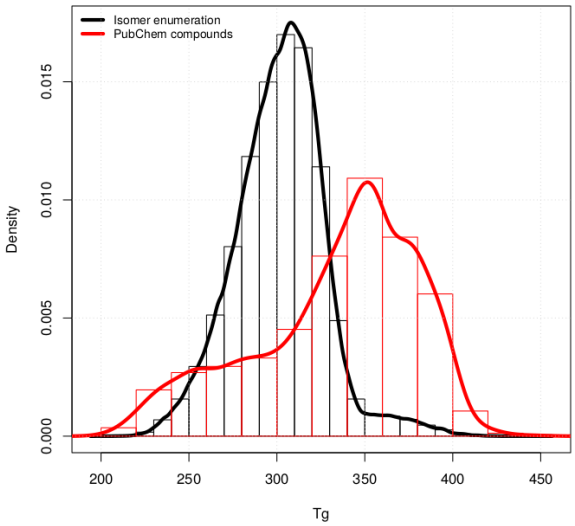


Figure 2.7 **Distributions of predicted T_g values for enumerated isomers and for isomers found in PubChem**. Distribution of predicted polymer glass transition temperature T_g for enumerated isomers and for isomers found in PubChem of a reference set of 158 monomers with a Tanimoto similarity greater than 0.5.

by defining all naturally produced chemicals as sinks in the workflow and using a collection of known enzymatic reaction rules in reversed mode. The process has been described in detailed elsewhere [138]. Through the application of the rules in a retrosynthetic fashion, it is possible to determine the routes that connect the target compounds to the natural precursors. Of the 158 available monomers, using the RetroPath2.0 workflow downloaded from MyExperiment.org [138], we were able to identify 17 compounds that can be naturally synthesized (Figure 2.8A). We provide in an archive containing the list of pathways for those 17 compounds.

The QSPR model for T_g was applied to the set of enumerated isomers. As shown in Figure 2.8B, the resulting set provided a good covering of the chemical space surrounding the starting monomer set. Moreover, a significant number of enumerated isomers show a high predicted T_g value, which may indicate a good candidate as a building block replacement for known monomers. Interestingly, those isomers that were close to biosynthetic accessible monomers (Tanimoto based on MACCS keys fingerprint > 0.8) have a distribution of predicted T_g values that significantly differ from the full set ($p - value < 1e - 12$ Welch t-test), with a mean $T_g = 352.1$K ($T_g = 301.9$ K in the full distribution). These close isomers to biosynthetically accessible monomers might be considered as good candidates for alternative biosynthesis since reaching them through biosynthesis may require only few modifications of the original catalytic route.

2.2.3 Search for molecules maximizing biological activities

In this section we are interested in searching chemical structures in the chemical space optimizing biological activities. This type of search can be solved using inverse QSAR procedures [238]. Inverse QSAR requires to first build a QSAR equation predicting activities from structure and then either (i) inverting the equation and enumerating structures matching a given activity [238] or (ii) searching in the chemical space structures similar to those used to build the QSAR equation [235] but having optimized activities. The second approach makes use of either deterministic methods such as lattice enumeration [239] or stochastic searches.

We propose here to use RetroPath2.0 to solve the inverse QSAR problem using a stochastic approach with isomer transformation rules and enzymatic rules for biosynthetic accessibility. To this end, we selected a dataset of 47 aminoglycosides structures for which antibacterial activities have been measured using a Minimum Inhibitory Concentration (MIC) assay [240]. The dataset is composed of natural aminoglycosides (gentamicin, tobramycin, neomycin, kanamycin A and B, paromomycin, ribostamycin and neamine) to which are added synthetic structures built on a neamine scaffold. This dataset has already been used to build a QSAR model based on Comparative MolecularField Analysis (CoMFA) analysis leading to a Q^2 of 0.6 for a Leave-One-Out (LOO) procedure [240]. We provide in Supplementary a QSAR workflow that makes use of RDKit fingerprints [194] and random forest as a learner leading to a higher Q^2 (0.7) for LOO. With that QSAR in hand we run RetroPath2.0 with a source set composed of the 47 aminoglycosides used in the training set, and two different reaction rules sets. The first set is extracted from the transforming enumeration rules depicted in Figure 2.4, the second set is composed of enzymatic reaction rules leading to neamine (an aminoglycoside) biosynthesis from glucose. Reaction rules for the second set were computed as explained in the method section resulting in 94 rules specific to the biosynthesis of aminoglycosides.

In both cases, reactions rules were fired on the initial source set composed of 47 structures. All rule products were ranked according to their predicted activities as calculated by the QSAR and were selected for the next iteration according to a tournament procedure described in the methods section which derives from [241]. The Figure 2.9 (A and B) below gives the top activity and the average population activity vs. iteration. The most active structures found by each rule set are also drawn in Figure 2.9 (C and D).

We observe in Figure 2.9 that the average curve in B is lower that the one in A. This is due to the fact that enzymatic rules generated a lot of compounds that are structurally far from aminoglycoside (i.e. H_2O, NH_4^+, $O_2 \ldots$). Moreover, the rules used for A, allow more transformation / modification, thus enabling to better explore the chemical space, and ultimately finding more active compounds. We note that the structure in Figure 2.9 (C) have a slightly better predicted activity (pACT = 9.015) than the initial compounds used in the training set, while the structure in Figure 2.9 (D) have the same predicted activity than gentamicin (pACT = 8.867).

2.2.4 Metabolome completion and metabolomics

In this last example we use enzymatic reaction rules in an attempt to complete the metabolome of species used in biotechnology. We are motivated here by current efforts invested to complete the knowledge on the metabolism of various organisms [214, 188, 182]. The benefits are numerous and include the identification of relevant biomarkers for many diseases; for personalized nutrition advice; and also for searching for relevant indicators and metabolites of plant and animal stress in agricultural practices and breeding programs. Additionally, knowing the metabolic space of microbes is an essential step for optimizing metabolic engineering and creating synthesis pathways for new compounds for industrial applications. Experimental evidences from metabolomics analyses are often informing us that with currently known metabolites one cannot cover the ranges of masses found in actual samples, and consequently there is a need of completing the metabolomes of interest. This need is clearly seen in the Human Metabolome Database (HMDB) where the number of reported masses has recently grown from 20,931 in 2013 [242] to 74,461 (at the time this manuscript was written), while annotated metabolites in metabolic databases are still in the range of 1847 (HumanCyc). Despite such a growth in databases, a significant amount of spectral peaks remains unassigned. This high fraction of unassigned peaks might be due to several factors including isotope, adduct formation, ion fragmentation, and multimers. Besides such sources of uncertainty in samples, many unassigned peaks should also be due to promiscuous activities of enzymes not yet characterized because of the lack of an appropriate description of the mechanisms of enzyme promiscuity.

To gain insights into those mechanisms enabling promiscuity, reaction rules have been shown to be appropriate [181] in particular the rules allowing to focus on the center of the reactions. To this end, several enzymatic reaction rules have been proposed such as those derived from bond-electron matrices [137], on the smallest molecular substructure changing during transfor-

mations [191], or on reaction rules that code for variable environments at reaction centers (see [188] and method section). That latter reaction rule system codes for changes in atom bonding environments where the reaction is taking place and the environment can range from including only the atoms participating to the reaction center to the entire set of atoms and molecules participating to the reaction. The advantage of that latter approach is that the size of the environment (named diameter) can be tuned to control the combinatorial explosion of possible products.
The degree of plasticity in metabolic networks that is uncovered by variable reaction center diameter is actually revealing an intrinsic feature of organisms linked to their adaptability, i.e. enzyme promiscuity. Promiscuity stands for the ability of enzymes to catalyze more than one reaction or to accept more than one substrate, a mechanism which can be traced to the evolutionary origins of enzymatic functions. Mimicking nature, such enzyme versatility can provide novel ways for biosynthesizing metabolite and even bioproducing non-natural molecule. To that end, the variable diameter method has shown itself to be especially well-suited for modeling the mechanisms of enzyme promiscuity as it has already enabled the experimentally validated discovery of a novel metabolite in *E. coli* and of the promiscuous enzymes producing it [181].
In this study, we make use of RetroPath2.0 to exemplify how variable reaction rule diameters can be used to complete the metabolome of *E. coli*. More precisely, we used as a source set all the metabolites present in *E. coli* iJO1366 model [216]. We first tested the two rule sets aforementioned, a set of about 100 reaction rules part of the BNICE framework [137] and a set of 50 reaction rules developed with the Sympheny software [191]. The reaction rules coded in the form of SMARTS string are provided in Supplementary at MyExperiment.org, along with the EC numbers corresponding to the rules. While the two rule sets were not developed to code only for *E. coli* reactions, for each EC number there is a corresponding enzyme annotated in*E. coli* so we kept all rules in the two sets. We then tested reaction rules with variable diameters using the procedure described in the method section to code for all *E. coli* metabolic reactions extracted from iJO1366 model. Rules were calculated for each reaction with diameters ranging from 2 to 16. Table 2.2 below provides the number of compound generated running RetroPath2.0 for one iteration on the metabolites of the iJO1366 model and the rules sets mentioned above (see Method section for additional details).

Table 2.2 shows that the number of compounds generated increases as the diameter decreases. This is consistent with the fact that shorter diameters will accept more substrates than higher ones and will thus produce more products. Although they were not constructed with diameters, the BNICE

Reaction rule set	**Nbr of compounds generated**	***E. coli* model coverage (1)**	**MS peaks coverage (2)**	**Median nbr of compounds per peak**	**Averaged nbr of compounds per peak**
E. coli Model [43]	751	100.0	12.3	1	1.5
Sympheny [9]	9448	48.2	40.4	3	6.3
BNICE [42]	8421	68.8	45.3	3	5.8
D16 (3)	1230	82.0	23.1	1	2.0
D10	2992	83.6	25.6	1	2.3
D8	5055	84.2	28.1	1	2.7
D6	11981	84.7	46.6	2	3.1
D4	37450	86.8	60.6	2	5.7
D2	162480	91.7	79.9	8	16.9

Table 2.2 **Metabolome completion.** Compounds generated by RetroPath2.0 using various reaction rules applied on *E. coli* iJO1366 model metabolites [216]. All numbers correspond to compounds having different InChIs at the connectivity level
(1) The *E. coli* model contains 751 compounds (with different connectivity InChIs). The column reports the % of these 751 compounds generated by the different rule sets. (2) The MS spectra were downloaded from Metabolight [243] and the OpenMS workflow described in the Method section retrieved a total of 800 distinct peaks. The column reports the % of peak assigned to at least one compound generated by the rule sets. (3) The number indicates the diameter

and Sympheny rule sets generally correspond to small environments comprising only few atoms and bonds around reaction centers, which explain why these two systems generate more products than high diameter rule sets. Nonetheless, even with high diameters, all variable diameter rule sets produce more molecules found in *E. coli* model than the BNICE and Sympheny rule sets. This might indicate that the variable rule sets correspond to a more accurate encoding of metabolic reactions than the other systems.
To further probe the coverage of the various rules sets listed in Table 2.2 we searched if the compounds produced could be found in MS spectra. To this end, we downloaded MS spectra from Metabolight [243] where masses have been measured on *E. coli* cell extracts. The spectra downloaded corresponded to a study aimed at probing the dynamics of isotopically labeled molecules (i.e. 13C labeled glucose) [244]. Since we are concerned here with wild type *E. coli* metabolome, we considered only the spectra where *E. coli* cells had not yet been exposed to labeled glucose (spectra acquired at time t=0). All compounds generated by our various rules sets were prepared to be read by OpenMS nodes [226] and a workflow was written with these nodes to annotate the MS spectra peaks (cf. Method section for details).
The results presented in Table 2.2 show that as the diameter decreases the

number of peak assignment increases, which is not surprising considering that the number of compounds generated increases as well. We observe that the Sympheny and BNICE rules sets give results similar to those obtained by the D6 rule set, albeit with a higher number of annotations per peak. In all cases the rule sets produced compounds not present in the *E. coli* model but with corresponding masses in the MS spectra. A supplementary Table available online gives a list of 40 such compounds having an identifier in MetaNetX [202] and produced by three identical reactions (i.e. reactions having the same substrates and products) generated using the Sympheny, BNICE and D6 rule sets. The compounds were produced by 53 reactions, some compounds being produced by more than one reaction. We note that the 40 compounds have been generated by rule sets for which at least one gene in *E. coli* has been annotated with the same corresponding EC number. The 40 compounds are thus potential new *E. coli* metabolites and their presence should be further verified using for instance MS/MS analysis.

2.3 Conclusions

In this paper we have presented a general method allowing one to explore the chemical space around a given molecule, or around a given set of molecules. The originality of the method is that the exploration is performed through chemical reactions rules. We have given a set of rules allowing us to generate any isomer of any given molecule of the chemical space. We also provide examples making use of reaction rules computed from enzymatic reactions. Using rules computed on known reactions has a definite advantage regarding the (bio)synthetic accessibility of the molecule produced, which is not necessarily the case for other techniques producing molecules de novo [235, 238, 241, 245, 246, 247, 248, 249].

Our method has been implemented into RetroPath2.0, a workflow running on the KNIME Analytics Platform [223]. RetroPath2.0 can easily be used with source molecules and reaction rules different than those presented in the paper. For instance the workflows provided in Supplementary at MyExperiment.org can be used with the reaction SMARTS rules and fragment libraries (as source compounds) of the DOGS software (inSili.com from [220]) developed for de novo drug design, other technique evolving molecules toward specific activities or properties [241, 248, 249] could also be implemented in RetroPath2.0 provided that one first codes reaction rules in SMARTS format.

Aside from searching molecules having interesting properties and activities RetroPath2.0 can also be used to complete metabolic maps by proposing

new metabolites biosynthesized through promiscuous enzymes, these new metabolites can in turn be used to annotate MS spectra and to that end we provide an interface with OpenMS [226]. Finally, RetroPath2.0 was originally developed to enumerate pathways producing a given target product from a source set of reactants. While we have benchmarked the workflow in the context of metabolic engineering, [138] it can also be used for synthesis planning as long as synthesis reaction rules are available.

2.4 Methods

2.4.1 Generating reaction rules

All our reaction rules are represented in the form of reaction SMARTS [108]. Reaction rules used for canonical augmentation are provided in Figure 2.2 and for isomer transformation in Figures 2.4, 2.5 and 2.10 and Table 2.3. Enzymatic reaction rules were computed taking enzymatic reactions from MetaNetX version 2.0 [202]. To compute rules, we first performed an AAM using the tool developed by [195] (Figure 2.11 A). Next, multiple substrates reactions were decomposed into components (panels C and D in Figure 2.11). There are as many components as there are substrates and each component gives the transformation between one substrate and the products. Each product must contain at least one atom from the substrate according to the AAM. This strategy enforces that only one substrate can differ at a time from the substrates of the reference reaction when applying the rule.

The following step consisted in computing reactions rules as reaction SMARTS for each component. We did it for diameters 2 to 16 around the reaction center (panels C and D in Figure 2.11) by removing from the reaction components all atoms that were not in the spheres around the reaction center atoms.
We extracted more than 24,000 reaction components from MetaNetX reactions, each one of those leading to a rule at each diameter (from 2 to 16).
We provide in Supplementary at MyExperiment.org a subset of 14,300 rules for *E. coli* metabolism. The rules were selected based on the MetaNetX binding to external databases and the iJO1366 whole-cell *E. coli* metabolic model [216]. We also provide enzymatic rules enabling the biosynthesis of aminoglycosides from Glucose. The reactions were extracted from the map00524 KEGG map [250], and rules were computed as above on reac-

tions for which a MetaNetX identifier could be retrieved. The resulting set comprised 94 rules calculated for each diameter ranging from 2 to 16.

2.4.2 RetroPath2.0 core algorithm

The RetroPath2.0 workflow essentially follows an algorithm proposed by some of us [222, 138] and its workflow implementation, which has already been described in details in [138], is summarized in Figure 2.12. We here focus on the different usages of RetroPath2.0 for the use cases provided in section 2.2.
In all cases the workflow performs the generation of structures in a breadth-first way by applying iteratively the same procedure. An iteration starts by applying reaction rules to each of the compounds of a source set. For each compound, the products are computed using the RDKit KNIME nodes one-component or two-component reactions [194]. Products are sanitized (removal of structures having incorrect valence), standardized and duplicates are merged. The set of products will become the new source set for the next iteration. The workflow iterates until a predefined number of iterations is reached or until the source set is empty.

In the case of isomer augmentation (workflow RetroPath2.0-Mods-isomer-augmentation, sections 2.2.1) the initial source set is composed of a single carbon atom and the rule used is R_1 in Figure 2.2, since it is the only rule that will produce acyclic molecules. The rule is fired on the source set, and the products become the new source set in the next iteration. The workflow is iterated a number of times equal to $N - 1$, where N is the number of atoms one wishes the final molecule to have.
In the case of isomer transformation (workflow RetroPath2.0-Mods-isomer-transformation, sections 2.2.1 and 2.2.2) the initial source set is composed of a molecule that is filled with the appropriate number of hydrogens using the RDKit KNIME node Add Hs. At each iteration rules are fired on the source set and the products obtained become the new source set for the next iteration. As an additional last step of each iteration, products that have already been processed in a previous iteration are filtered out before building the next source set. This necessitates maintaining a set (named sink) comprising all molecules so far generated. All products that have already been obtained are removed from the product set and the remaining molecules are (i) added to the sink set and (ii) used as the new source set for the next iteration. This avoids applying reactions on the same products during subsequent iterations. Disconnected structures are removed from the results by filtering out any product having several disconnected components (accord-

ing to the SMILES representation). When enumerating alkane, disconnected structures represents between 50 and 66% (depending of the alkane size) of the generated structures before filtering and merging duplicates. To generate the results of Table 2.1, since we are enumerating alkanes (no multiple bonds or cycles), the rule to be used is R_1 in Figure 2.4. To enumerate the isomers of the monomers in section 2.2.2, if we prohibit the transformation of multiple bonds into cycles and thus keep the number of single, double and triple bonds constant, the rules to be used are $R1$, $R5$ and $R9$ in Figure 2.4 (also found in Figure 2.10 since the monomers used do not contain triple bonds). Since this algorithm can become computationally intensive, we also provide an additional workflow (called RetroPath2.0-Mods-isomer-transformation-queue) to deal with memory management. This workflow illustrates how to introduce a First In First Out (FIFO) data structure for the source set (i.e. queue containing structures upon which rules will be fired) and use it for iteratively firing rules on small chunks of structures (e.g. chunk of 20 structures), new products obtained are then added to the source queue. Interestingly, the breadth-first approach for generating the structures can be replaced by a depth-first approach by replacing the queue (first in, first out structure) by a stack (last in, first out structure).
In the case of inverse-QSAR (workflow RetroPath2.0-Mods-iQSAR, section 2.2.3), the source set initially comprises the molecules used in the training set when building the QSAR. At each iteration, one or two molecules are chosen at random from the source set depending on the rule set that is being used (one molecule with enzymatic reaction rules, two molecules with isomer transformation rules). Rules are then fired on the selected molecules and an activity is predicted for each product using the QSAR equation. The source set is updated retaining molecules according to a selection tournament procedure borrowed from [241]. Briefly, the initial source set (i.e. the set of structures used at the start of the current iteration) is merged with the product set (i.e. the set of structures obtained after firing the rules). This merged set is then randomly split into 10 subsets and the 10 top best structures from each subset are retained according to their predicted activity. Finally, all the retained structures are pooled together to form the updated source set to be used at the next iteration. The workflow is iterated a (user) predefined number of times.
In the case of *E. coli* metabolic network completion (workflow RetroPath2.0-Mods-metabolomics, section metabolome completion and metabolomics), we provide three workflows. The first workflow is RetroPath2.0, which is fully described in [138] and is similar to the isomer transformation one. Here, RetroPath2.0 produces a list of molecules obtained using *E. coli* enzymatic reaction rules (see Generating reaction rules section). The second workflow takes as input the products generated by RetroPath2.0, computes the exact

mass for each product and prepare files to be read by OpenMS nodes for MS data peak assignment [226]. The last workflow is built with OpenMS nodes, it reads several MS data files in mzML format, two lists of adducts in positive and negative modes, and the files generated by the second workflow (containing RetroPath2.0 generated products with masses). The workflow searches for each compound the corresponding peak in the MS spectra. The workflow was parameterized for metabolomics analysis as described in OpenMS manual [251], the AccurateMassSearch node was set to negative ion mode as the experiment were carried out with an LTQ-Orbitrap instrument operating in negative FT mode (cf. protocols in [243]).
Further details on how to run all the above workflows are provided in the Supplementary at MyExperiment.org

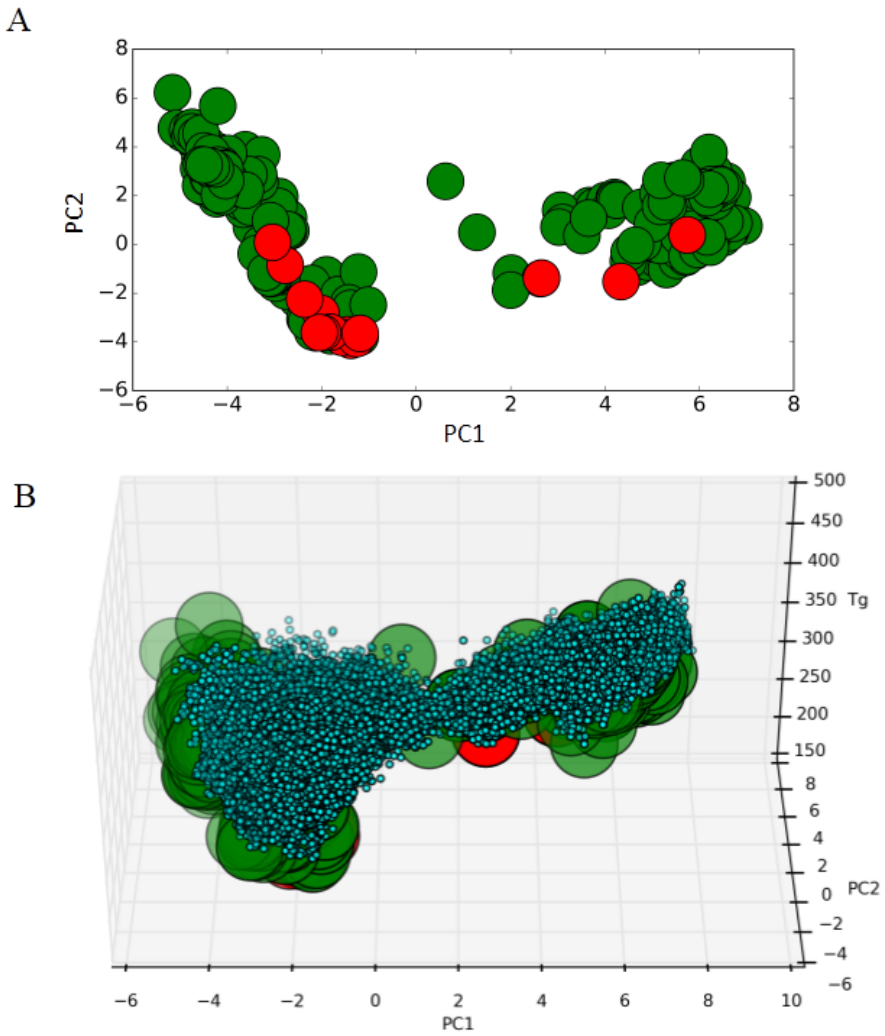


Figure 2.8 **Representation of monomers and isomers in the chemical space**. **A** Initial 158 monomers (green big circles) represented in the chemical space of chemical descriptors using the two main principal components computed from the MACCS fingerprints as axes. Monomers that can be produced through biosynthesis are represented as big circles in red. **B** Covering of the chemical space generated by the 574,186 isomers (blue) enumerated for the 158 monomers (green) with a Tanimoto similarity greater than 0.5 and associated predicted T_g property of the resulting polymer. Virtual monomers are depicted as small circles to facilitate visualization of their distribution around the starting monomers.

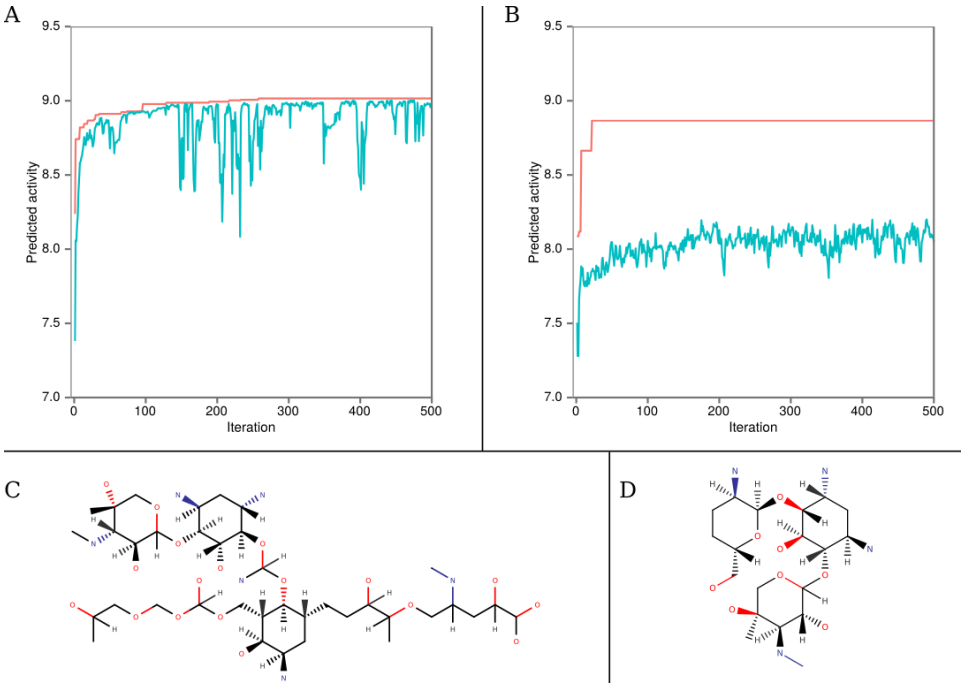


Figure 2.9 **Evolution of predicted activities**. **A** and **B** Evolution vs. iteration number of the best predicted activity (red) and average population predicted activity (blue) from among the newly generated structures using **A** transformation enumeration rules or **B** enzymatic rules. **C** and **D** Selected best structure generated after 500 iterations using either **C** transformation enumeration rules or **D** enzymatic rules.

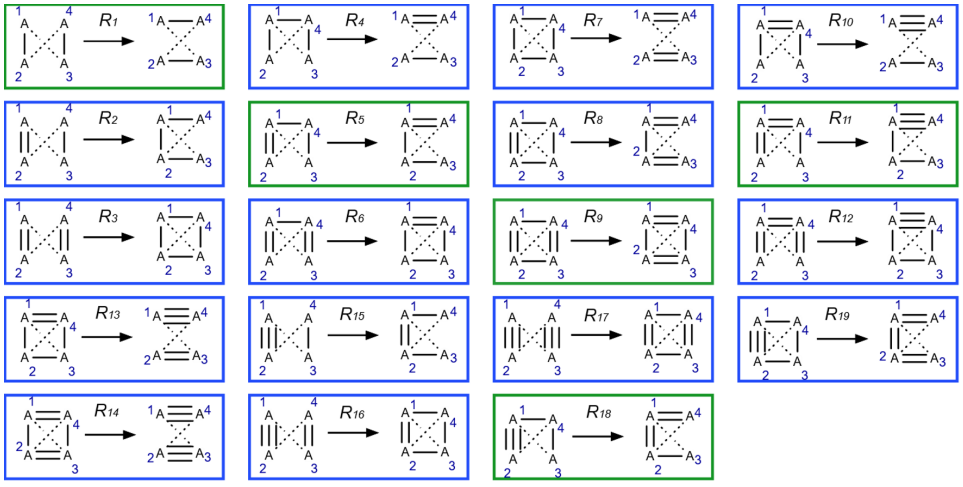


Figure 2.10 **Reduced isomer transformation rule set.**. Reduced set excluding triple bonds. Reactions in green move bonds around without creating or deleting cycles. Reactions in blue change bond order by creating or deleting at least one cycle. To each reaction corresponds a reverse reaction. The reverse reaction of R_1 is R_1, for R_2 it is R_4, for R_3 : R_7, for R_5: R_5, for R_6 : R_8 and the reverse reaction of R_9 is R_9. The bond order a_{13} and a_{24} can take any value from 0 to 3.

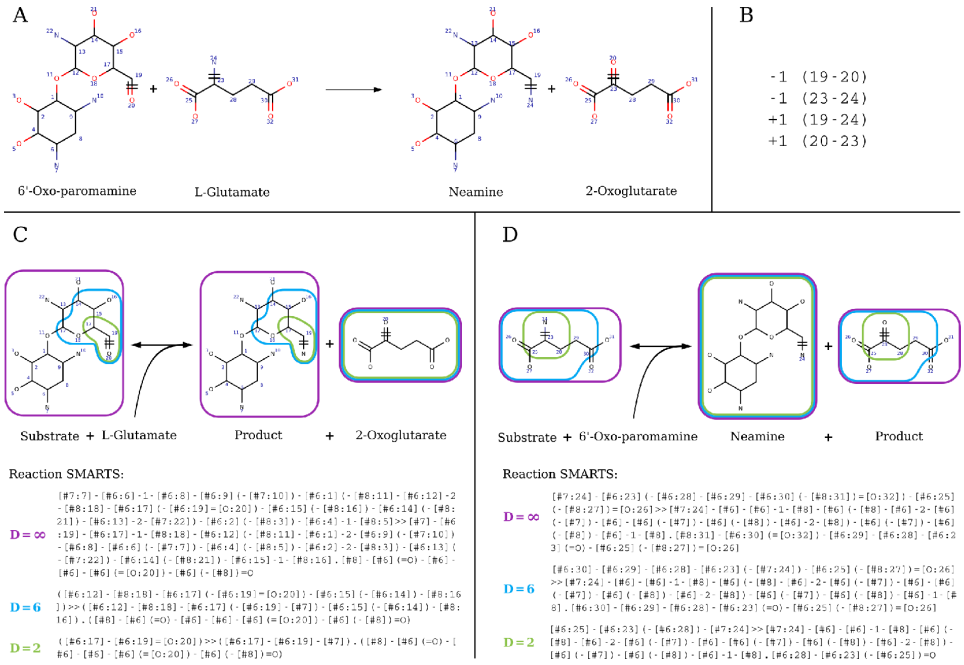


Figure 2.11 **RetroPath2.0 rules and corresponding SMARTS for reaction 2.6.1.93 at various diameters.** **A** Full reaction 2.6.1.93 with atom mapping. **B** The list of broken bonds (-1) and bonds formed (+1) is given by their atom numbers. **C** The corresponding SMARTS for the component modeling promiscuity on 6'-Oxo-paromamine: Substrate + L-Glutamate = Product + 2-Oxoglutarate. **D** The corresponding SMARTS for the component modeling promiscuity on L-Glutamate: Substrate + 6'-Oxo-paramamine = Neamine + Product. **C** and **D**. Rules are encoded as reaction SMARTS and characterized by their diameter (∞ purple, 6 blue or 2 green), that is the number of bonds around the reaction center (atoms 19, 20 and 23, 24) defining the atoms kept in the rule.

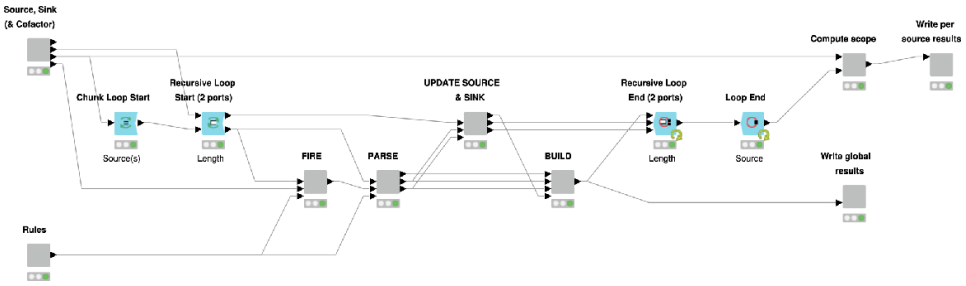


Figure 2.12 **RetroPath2.0 KNIME workflow. Inner view of the "Core" node where the computation takes place.** The "Source, Sink..." and "Rules" nodes parse the source, sink and rules input files provided by the user and standardize data so that it can be processed by downstream nodes. Definitions for source, sink, and rule sets are provided in the text. The outer loop ("Source" loop) iterates over each source compounds, while the inner loop ("Length" loop) allows to iterate the process up to a maximum number of steps predefined by the user. The nodes (i) "FIRE", (ii) "PARSE", (iii) "UPDATE SOURCE..." and (iv) "BUILD" are sequentially executed at each inner iteration. Respectively, they (i) apply all the rules on source compounds, (ii) parse and standardize new products, (iii) update the lists of source and sink compounds for the next iteration and (iv) merge results that will be written by the node "Write global results". Once the maximum number of steps is reached (or no new product is found), the "Compute scope" node identify the scope linking each source to the sink compounds, then these results are written by the node "Write per source results". Only the main nodes involved in the process are shown.

2.5 Supplementary Data

Rule ID	Rule
R_1	$([*:1]-[*:2].[*:3]-[*:4])>>([*:1]-[*:4].[*:2]-[*:3])$
R_2	$([*:1]=[*:2].[*:3]-[*:4])>>([*:3]-[*:2]-[*:1]-[*:4])$
R_3	$([*:1]=[*:2].[*:3]=[*:4])>>([*:1]-1-[*:2]-[*:3]-[*:4]-1)$
R_4	$([*:2]-[*:1]-[*:4]-[*:3])>>([*:2]-[*:3].[*:1]=[*:4])$
R_5	$([*:3]-[*:4]-[*:1]=[*:2])>>([*:3]-[*:2]-[*:1]=[*:4])$
R_6	$([*:2]=[*:1]-[*:4]=[*:3])>>([*:2]-1-[*:3]-[*:4]=[*:1]-1)$
R_7	$([*:1]-1-[*:2]-[*:3]-[*:4]-1)>>([*:2]=[*:3].[*:1]=[*:4])$
R_8	$([*:3]-1-[*:4]-[*:1]=[*:2]-1)>>([*:3]=[*:2]-[*:1]=[*:4])$
R_9	$([*:1]-1=[*:2]-[*:3]=[*:4]-1)>>([*:1]-1=[*:4]-[*:3]=[*:2]-1)$
R_{10}	$([*:2]-[*:1]=[*:4]-[*:3])>>([*:2]-[*:3].[*:1]\#[*:4])$
R_{11}	$([*:3]-[*:4]=[*:1]=[*:2])>>([*:3]-[*:2]-[*:1]\#[*:4])$
R_{12}	$([*:2]=[*:1]=[*:4]=[*:3])>>([*:2]-1-[*:3]-[*:4]\#[*:1]-1)$
R_{13}	$([*:2]-1-[*:3]-[*:4]=[*:1]-1)>>([*:2]=[*:3].[*:1]\#[*:4])$
R_{14}	$([*:1]-1=[*:4]-[*:3]=[*:2]-1)>>([*:2]\#[*:3].[*:1]\#[*:4])$
R_{15}	$([*:1]\#[*:2].[*:3]-[*:4])>>([*:3]-[*:2]=[*:1]-[*:4])$
R_{16}	$([*:1]\#[*:2].[*:3]=[*:4])>>([*:3]-1-[*:4]-[*:1]=[*:2]-1)$
R_{17}	$([*:1]\#[*:2].[*:3]\#[*:4])>>([*:1]-1=[*:2]-[*:3]=[*:4]-1)$
R_{18}	$([*:3]-[*:4]-[*:1]\#[*:2])>>([*:3]-[*:2]=[*:1]=[*:4])$
R_{19}	$([*:3]-1-[*:4]-[*:1]\#[*:2]-1)>>([*:3]=[*:2]=[*:1]=[*:4])$

Table 2.3 **List of the 19 SMARTS rules that were used in this study.**

a_{12}	a_{34}	a_{23}	a_{14}	**Rule or argument for space reduction**
1	1	0	0	Rule 1
1	1	0	1	Rule 4
1	1	0	2	Rule 10
1	1	1	1	Rule 7
1	1	1	2	Rule 13
1	1	2	2	Rule 14
2	1	0	0	Rule 2
2	1	0	1	Rule 5
2	1	0	2	Rule 11
2	1	1	1	Rule 8
2	1	1	2	Double bonds in 4 membered ring
2	1	2	2	Double bonds in 4 membered ring
2	2	0	0	Rule 3
2	2	0	1	Rule 6
2	2	0	2	Rule 12
2	2	1	1	Rule 9
2	2	1	2	Double bonds in 4 membered ring
2	2	2	2	Double bonds in 4 membered ring
3	1	0	0	Rule 15
3	1	0	1	Rule 18
3	1	0	2	Valence is limited to 4
3	1	1	1	Rule 19
3	1	1	2	Valence is limited to 4
3	1	2	2	Valence is limited to 4
3	2	0	0	Rule 16
3	2	0	1	Double bonds in 4 membered ring
3	2	0	2	Valence is limited to 4
3	2	1	1	Double bonds in 4 membered ring
3	2	1	2	Valence is limited to 4
3	2	2	2	Valence is limited to 4
3	3	0	0	Rule 17
3	3	0	1	Double bonds in 4 membered ring
3	3	0	2	Valence is limited to 4
3	3	1	1	Double bonds in 4 membered ring
3	3	1	2	Valence is limited to 4
3	3	2	2	Valence is limited to 4

Table 2.4 **Solution space reduction arguments or rules** corresponding to bond configurations in section Isomer enumeration. This is shown for the 36 rules after reduction of the solution space by problem symmetry arguments.

3

RetroPath3.0: Similarity-guided Monte Carlo Tree Search for metabolic engineering

This work has been submitted for publication by Mathilde Koch, Thomas Duigou and Jean-Loup Faulon.
Only minor modifications to the submitted paper have been introduced in the Chapter below. Most notable modifications of the submitted paper consisted in moving section on detailed golden set analysis, database sped-up calculations, extension of previous searches to the Supplementary for space constraints.

Detailed Contribution to this thesis

In this Chapter, I present the implementation of a novel bio-retrosynthetic algorithm that overcomes some of the limitations presented in Chapters 1 and 2. The formalism used to derive reaction rules from knowledge databases and apply them to compounds, as well as the way promiscuous reactions are encoded, is mostly identical except for small technical details. However, my main contribution was the adaptation of the Monte Carlo Tree Search algorithm to our case, guiding it with a score that encompasses both chemical relevance of the applied reaction and likelihood that an enzymatic sequence exists to catalyze it. I developed the RetroPath 3.0 software built around this algorithm, and validated it on two different datasets. First, on a small and manually curated dataset of 20 compounds, the capacity of the algorithm to find pathways described in the literature was evaluated, in order o assess the biological relevance of generated pathways. Then, its capacity to find pathways for compounds that are described in a dataset of successful metabolic engineering projects was also evaluated. I also developed other features that allow for faster searches for frequent users, making it relevant for metabolic engineering from a practical standpoint. I then showcased its modularity by including toxicity prediction in the algorithm, to try and

avoid toxic intermediates. Therefore, this software, that overcomes some algorithmic limitations of previous retrosynthetic tools, becomes a better design tool for designing metabolic pathways. This is particularly useful in the context of development of synthetic metabolic circuits, which use metabolism to perform computation: such circuits require an efficient way to explore the metabolic space around detectable compounds.

Full reference

Not available yet.

Contributions as stated in the article

M.K., T.D. and J.-L. F. designed the study. M.K. developed the MCTS algorithm. T.D. developed rule extraction and application code, chassis analysis and LASER extraction. Both M.K. and T.D. tested and validated the software. M.K., T.D. and J.-L. F. wrote the paper.

3.1 Abstract

Metabolic engineering aims to produce chemicals of interest from living organisms, to advance towards greener chemistry. However, the research and development process is still long and costly and efficient computational design tools are required to explore the chemical biosynthetic space. We provide RetroPath 3.0, an open-source, modular command line tool that explores the bio-retrosynthesis space using a similarity-guided Monte Carlo Tree Search. We validate it on a dataset of manually curated experimental pathways as well as on a larger dataset of successful metabolic engineering projects. Moreover, we provide a novel feature, that suggests potential media supplements to complement the enzymatic synthesis plan.

3.2 Introduction

Efficient computational tools are required for metabolic engineering to achieve its true potential as a game-changer in the bioeconomy. Such tools include pathway design tools, to assist the metabolic engineer in finding new pathways for production of a target of interest. While some tools restrict

themselves to reactions already present in databases [186, 185], others allow the generation of de novo reactions, using retrosynthesis algorithms [252, 138, 188, 137, 210, 191, 190, 187, 136]. At its core, a retrosynthesis algorithm is simple: break down a target molecule into simpler molecules that can be combined chemically or enzymatically to produce it, and iterate recursively until all required compounds are either commercially available or present in the chassis organism of choice. Current bioretrosynthesis tools suffer from a number of limits. First of all, they are usually accessible by a web-server and not locally, limiting an expert user's capacity to tune them. Secondly, a number of parameters are often included within the pathway search, decided by the software designer and with limited capacity for a user to incorporate his own knowledge, as the retrosynthesis tool solves for both optimization of those parameters and actual retrosynthetic pathway search. Some examples include enzyme performance [138], predicted yield [187, 188, 190, 212, 253], thermodynamics or cofactor usage [252]. Moreover, those tools rarely include the latest advances in combinatorial search space exploration, pioneered in the field of Artificial Intelligence.
In order to address those limitations, we present RetroPath 3.0, which is released as an open source python package freely available on Pypi and GitHub for community contributions. RetroPath 3.0's search algorithm relies on Monte Carlo Tree Search (MCTS), which has already revolutionized the field of Artificial Intelligence, as illustrated by the stunning victory of Google's AI (AlphaGo) against a Go master in 2016 [104, 82, 83]. An interesting application used this algorithm combined with neural networks in chemical retrosynthesis, but acknowledging that natural compounds synthesis was beyond their scope [87]. Our aim with RetroPath 3.0 is to provide a command line software using MCTS, while allowing a number of augmentations and features to be used or developed by experts users to tune it to their needs. Lack of open-source computer-assisted pathway design tools is currently pointed out as one of the major limits faced by metabolic engineering projects [254, 255]. Therefore, RetroPath 3.0 is a timely software that will hopefully contribute to metabolic engineering realizing its true potential for green chemistry.

3.3 Theoretical background

3.3.1 Reaction rules for representing enzymatic reactions

We use reaction rules that describe the changes in bonding patterns when a set of substrates is transformed into a set of products to describe enzymatic

reactions. An important feature of rules for retrosynthetic applications is that they need to be generalisable, so that they can be applied to a new substrate that was not from among the substrates the rules were learned on. Moreover, using generalized reaction rules is the first step towards predicting promiscuous reactions, as those reactions are often missing from metabolic databases. Modeling promiscuity is a key feature in metabolic engineering, as it has for example been estimated that 37% of *E. coli* K12 enzymes have a promiscuous activity on structurally similar substrates [130]. In our data-driven approach, we learn rules at various levels of specificity around the reaction present in the database, by keeping in the described pattern of the rule a varying number of atoms around the reaction center. We select those atoms using a number we call diameter that represents the distance in bonds around the reaction center: a rule at diameter 2 will include atoms at a distance of 1 around the reaction center, while a rule a diameter 10 will include atoms at a distance of 5 around the reaction center. Therefore, the rule at diameter 2 can apply to more diverse substrates and therefore encode more promiscuity than the rule at diameter 10. A more detailed description of reaction rules can be found in [138, 116] and in the Methods section, as well as in Chapters 1 and 2.

3.3.2 Necessity of ranking reactions

Once we extracted our rules, we sought to find out how many rules on average applied to a compound. If this number is low enough, an exhaustive search can be considered, but in the other case, applying chemical reaction rules iteratively on substrates and their products leads to a combinatorial explosion. Our encoding of reaction rules at various diameters allows us to select the degree of promiscuity, which has a high impact on those statistics. Results for rule sets used in this study are presented in Table 3.1, and for individual diameters in Supplementary Table 3.2. We can see from this Table that the branching factor (the average number of rules that apply to a substrate) is around 100 when using rules at diameters 6, 10 and 16 (slightly promiscuous, medium and very specific) and drastically increases with the level of promiscuity (adding rules at diameter 2 that are highly promiscuous lead to a branching factor of 900). Therefore, the more promiscuity we allow, the higher our branching factor becomes.

Set of diameters used	6, 10, 16	2, 6, 10, 16	All (2 to 16)
Average number of applicable rules	100	900	1183

Table 3.1 **Average branching factor for chosen sets** Average number of applicable rules that on a compound according to the set of diameters used. Information on average number of rules at all individual diameters are available in Supplementary Table 3.2.

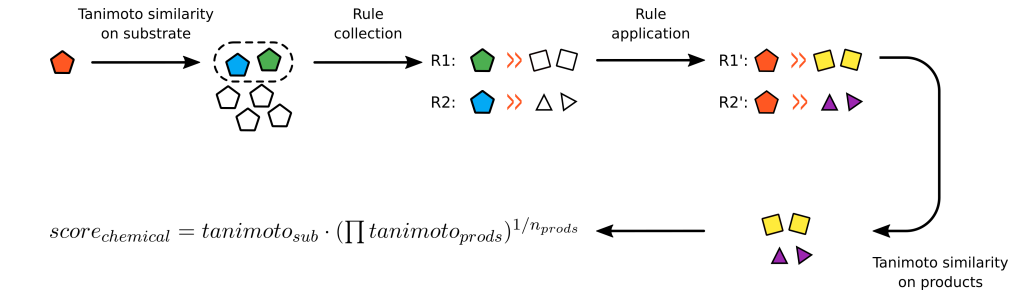
Such a high branching factor is comparable to the Go game (branching factor of 250) and much higher than chess (35), and the reason why using algorithms that were successful in this domain could also be of interest for retrosynthesis [104, 87]. We therefore use an algorithm (MCTS) that can effectively handle this combinatorial explosion, and a heuristic (chemical similarity) to guide the search.

3.3.3 Chemical similarity and sequence availability for reaction ranking

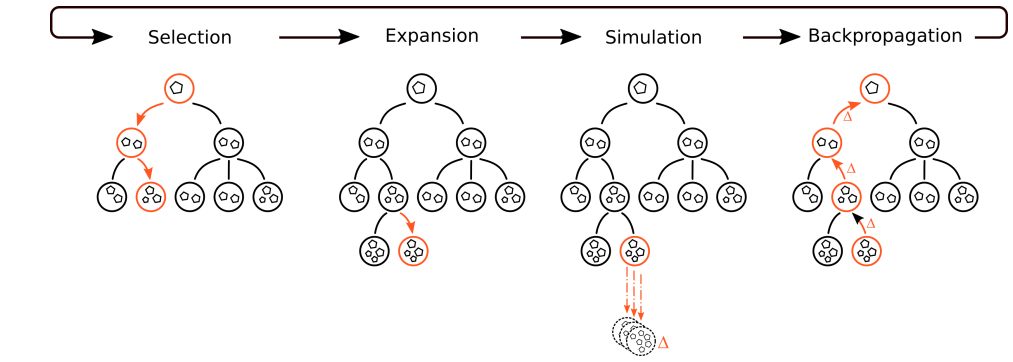
Chemical similarity between query (applied on a new substrate) and the native chemical transformation has been used in various studies [212, 256, 257, 187]. We adapted the strategy from Coley *et al.*, that proceeds in a 2 step evaluation of the reaction. In a first step, before rule application, similarity between query and native substrates is calculated. After rule application, similarity between query and native products is also calculated. This allows accounting of similarity in a manner straightforward to use with monocomponent reaction rules (Figure 3.1A). Using this metric allows us to select chemical reactions similar to the ones present in metabolic databases, increasing our chances that this predicted reaction can be catalyzed.
However, in [212] [256] [257], the authors are interested in chemical retrosynthesis, whereas we are interested in enzyme-catalyzed reactions. Therefore, we combine (through multiplication) this chemical score with a scoring scheme that we developed previously [138]. Briefly, this biological score characterizes our confidence that a sequence exists to catalyze the desired enzymatic rule. We updated this scoring scheme to be a normalized score, between 0 and 1, to be in the same range as the chemical score. This biological score has the useful property that rules at low diameter (i.e.: more promiscuous) are usually ranked lower than rules at high diameters (i.e.: more specific and trustworthy).
In RetroPath 3.0, this combined biochemical score is used for ranking and

excluding reaction rules that are not considered trustworthy (similarity too low to the original reaction, or sequence availability too low). We found cut-offs of 0.3 provide a good trade-off between allowing promiscuity and keeping realistic rules.

3.3.4 Integrating rule ranking with Monte Carlo Tree Search



(a)



(b)

Figure 3.1 **Presentation of the algorithm and the chemical scoring scheme employed** **a** Chemical score computation and rule selection scheme. We start by selecting substrates within the rule collection that are similar to the query substrate. We then apply those rule templates and check similarity of those products to the products the rules were learned on. **b** Monte Carlo Tree Search algorithm. Circles represent nodes, and pentagons molecules.
Detailed explanations are in the main text and in the Methods section.

A Monte Carlo Tree Search (MCTS) algorithm proceeds in 4 phases, repeated until a resource budget (time or number of iterations) has been exhausted (Figure 3.1B).

(Selection) Starting from the root node (here, a chemical state containing the target compound), the best children nodes according to the selection policy are iteratively chosen until a leaf node is reached. We use a selection policy guided by the biochemical score of the reaction rule, unless otherwise stated.

(Expansion) Possible transformations are selected based on the ranking scheme presented above and the node is expanded (with a predefined max-

imum number of children).
(Simulation or rollout) This is an iterative procedure that starts by checking the status of the state. If it is terminal, a reward (or penalty) is returned according to the rewarding policy (detailed in the methods section). If it is not terminal, a transformation is randomly sampled from available transformations and the process is repeated. This is performed until a maximum number of rollout steps or the maximal depth of the tree is reached.
(Backpropagation or update) The score obtained after exploring this node is returned to its parents to update their values and visit counts.
The biochemical rule ranking is used both in expansion and rollout phases. For evaluation of a state, we check whether compounds belong to a sink (chassis organism of interest for bioretrosynthesis but buyable compounds in retrosynthesis).

3.4 Results and Discussion

3.4.1 Evaluating RetroPath 3.0 with a golden dataset

We first evaluate our tool on a manually curated dataset of 20 compounds (see Methods for selection) to identify the best settings for a retrosynthetic search on 20 compounds (chosen compounds and rationale for selection are available in the Methods section). RetroPath 3.0 provides a number of features for expert users (See Supplementary Note 1) and we wanted to compare them on this golden dataset to select the best parameters possible. While various metrics could be available to describe what a good bioretrosynthesis algorithm should do, there is no obvious consensus. Should such an algorithm be fast? Return a lot of pathways? Return fewer but more reliable pathways? We use three criteria for comparing algorithms. First, it should return pathways for as many compounds as possible. Second, results should include the chosen literature-described experimental pathway (exact intermediates are found). For parameter sets with identical results on those two criteria, the third criteria is that a better parameter combination should return the experimental pathway in less iterations.

The best parameters set we found used chemical and biological thresholds of 0.3, and a maximum of 10 allowed children per node (detailed configuration is available as Supplementary Table 3.4 and effects of various parameters were investigated and presented in Supplementary Note 1). Results comparing RetroPath 3.0 and RetroPath 2.0 (run with the same set of rules at diameters 6, 10 and 16 and a timeout of 3 hours) on the golden dataset are

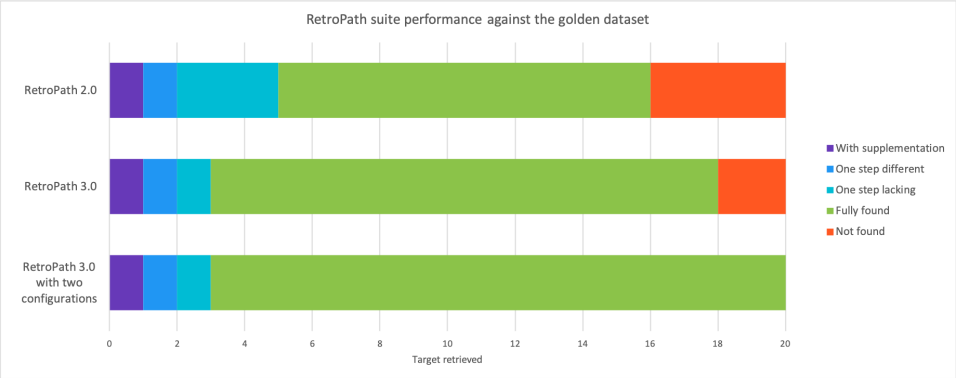


Figure 3.2 **Results of the RetroPath suite against the golden dataset to identify the experimental pathway.** We compared results of RetroPath 2.0, the default configuration of RetroPath 3.0 and a combination of results between the default configuration and a more tolerant one on the used scores with a timeout of 3 hours. With supplementation (purple) means a supplement has to be provided in the media to identify the correct experimental pathway. One step different (dark blue) means only one step differs from the described pathway, for example by using a different co-substrate. One step lacking (light blue) means the search algorithm found a pathway identical to the experimental one, except one step which was shortcutted. Fully found (green) means the experimental pathway was found without restriction. Not found (orange) means the experimental pathway was not found.

presented in Figure 3.2. For all compounds, at least one pathway was found with those settings with RetroPath 3.0, while one compound had no pathway with RetroPath 2.0. For 2-amino-1,3-propanediol, the same core pathway was found, but the identified co-substrate in the first step was different (D-alanine for RetroPath instead of D-glutamate for the experimental pathway, the main substrate being dihydroxyacetone phosphate) (one step different in Figure 3.2 - dark blue). For four compounds (TPA, N-methylpyrrolinium, 1,4 BDO and protopanaxadiol), the experimental pathway was not found for different reasons. For TPA, the described experimental pathway in our golden dataset starts from a compound added to the mix, xylene. Running our workflow adding this compound to the sink allows us to find the experimental pathway (Figure 3.2 - purple). For protopanaxadiol and N-methylpyrrolinium, we ran the MCTS using a different set of parameters, allowing both to explore more reactions (15 instead of 10) and more tolerance on the scores (cut-offs of 0.15 instead of 0.3). With those new settings we found the experimental pathway for both compounds. For 1,4 BDO, the experimental pathway was not found with these new settings either, but a similar pathway (lacking only one enzymatic step) was found with the default configuration. It transforms 4-hydroxybutyryl Coa into 1,4 BDO without using 4-Hydroxybutyraldehyde, supposedly catalysed by EC number (1.2.1.84: alcohol-forming fatty acyl-CoA reductase, without the alcohol dehydrogenase step from the literature example). The rest of this pathway is identical to the experimental pathway (one step lacking in Figure 3.2-

light blue).

In conclusion, our workflow found at least a pathway for each compound of our golden dataset (100% success), and found the experimental pathway 75% of the time with strict settings, and 95% of the time when trying more tolerant settings on failed compounds, media supplementation or using another cofactor (lacking one step is considered a failure). We can see this is better than our previous algorithm, proving that RetroPath 3.0 suggests more experimentally relevant pathways.

3.4.2 Importance of our scoring schemes

Although all parameters of interest are evaluated (with a timeout of 4 hours) in Supplementary Figures 3.6 to 3.16 in Supplementary Note 1, we detail here the impact of the scoring scheme. As mentioned in the theoretical background section, we use a biochemical score, based on both chemical similarity and estimation of enzyme sequence availability. We analyzed algorithm behavior when guided only by similarity, biological score or no scoring scheme (classical algorithm). The results are presented in Figure 3.3 and validate our approach. We can see the score contributing most to our biochemical score' success is the chemical similarity component, while the biological score mainly ensures experimental relevance.

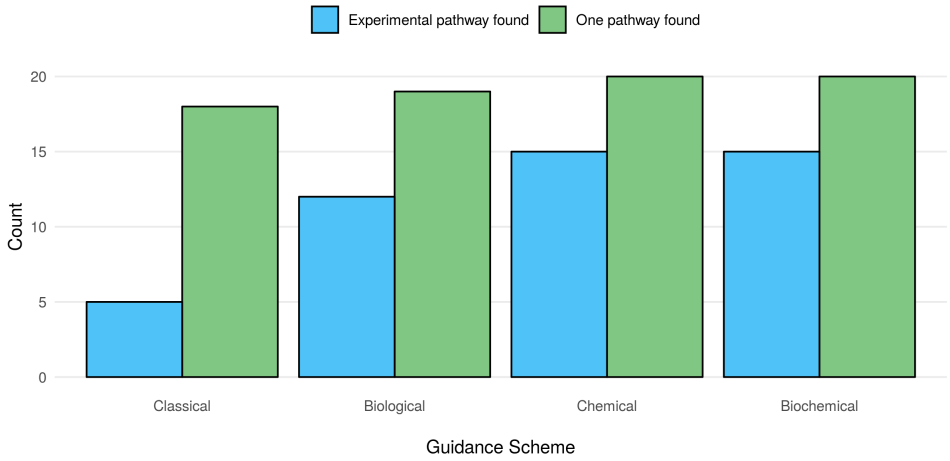


Figure 3.3 **Impact of guidance scheme on retrieval performance of RetroPath 3.0** We compared results between guiding the search based on the Classical UCT formula, a formula guided by Biological scoring, Chemical scoring or Biochemical scoring. One pathway found means that at least one pathway have been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

3.4.3 Evaluating RetroPath 3.0 on successful metabolic engineering projects

After validating relevance of predictions, we tested RetroPath 3.0 on a larger dataset. Our previous tool [138] was tested on the LASER database [258, 259] that compiles successful metabolic engineering projects, completed with compounds taken from the Metabolic Engineering journal (see Methods, available as Supplementary Data 1 available online). Given the curation level of this dataset, we checked the number of compounds for which we could find a pathway, and not the exact experimental pathway.
We ran our RetroPath 2.0 software using the same RetroRules rules (diameters 6, 10 and 16) and solved 77.6% compounds (118 out of 152), finding a median number of pathways of 4.5. With RetroPath 3.0 and a score cut-off of 0.3, we solved 121 compounds (79.6%) with a median number of pathways of 11.5. Without score cut-off, we obtain 6 more compounds, yielding a success rate of 83.6% (results are available as Supplementary Data 2 available online). The main advantage of RetroPath 3.0 over RetroPath 2.0 is its capacity to find longer pathways. Indeed, the memory requirements of the exhaustive search performed by RetroPath 2.0 essentially limit it to 5 step pathways, while RetroPath 3.0 can explore longer pathways and therefore find more solutions given the same allowed time. Its focus on promising areas of the search space also lead RetroPath 3.0 to propose more solutions for the same compound.

3.4.4 Supplement finder for media supplementation

The literature pathway that we identified for TPA used xylene for media supplementation, i.e.: xylene is not a metabolite from the microbial strain but was necessary for producing TPA and experimentally added to the strain's growth media. Knowing this allowed us to add xylene to the list of starting compounds for retrosynthetic search. Media supplementation is commonly done in metabolic engineering but rarely integrated into retrosynthesis tools. We therefore developed a RetroPath3.0 feature that analyses search trees and suggests potential media supplements, available from the chemical provider Sigma. We first extracted supplement information from LASER, as well as two TPA experimental pathways known from the literature. Filtering out gene inducers, antibiotics or early precursors, we obtain a list of 8 curated pathways with supplementation (available as Supplementary Data 3 available online). In 5 cases out of 8, we retrieve the compound that was used for supplementation in the described experimental pathway.

For 2 cases, we did not find the experimentally described supplement but suggest other possibilities. Here, we used availability in the Sigma-Aldrich database as a criterion on whether a compound could be an interesting supplement. However, this feature can be used with any database of interest, for example with in-house compounds of a laboratory. One could include criteria such as capacity to cross membranes, solubility, toxicity, cost or any other feature of interest and select compounds that are biologically relevant for their application of interest.

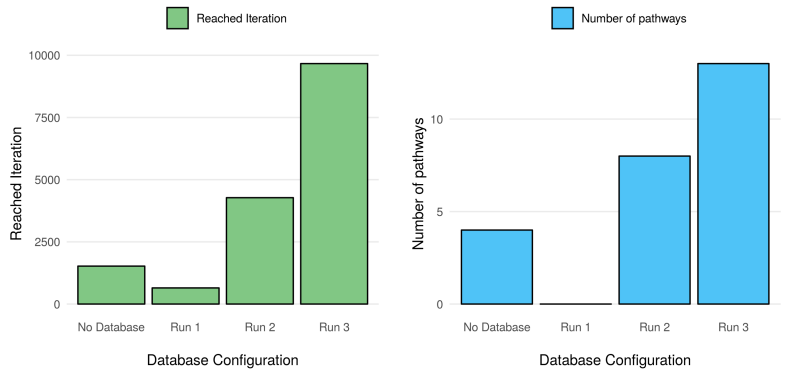
3.4.5 Custom use of RetroPath 3.0: avoid toxic intermediates

We sought to make our tool as modular and flexible as possible, therefore allowing expert users to input their knowledge. We showcase this by implementing a toxicity score to bias the search away from toxic compounds [211, 188]. This toxicity score is negative (between 0 and -10 in our training dataset) for toxic compounds, and set to 0 otherwise. The strength of using bias in MCTS is that it favors preferred routes (here, avoiding toxic intermediates) but can still find results if those preferred routes are not successful, in contrary to other algorithms that would exclude those intermediates altogether. While implementing this feature did not change our results on the golden dataset (i.e.: the experimental pathway was identified for the same compounds), the order as well as the total number of returned pathways was impacted (Supplementary Table 3.3).
While we tested it with toxicity, biasing the search could also be used to encourage pathways to be found from a set of privileged metabolites (core metabolism), by cost, availability in the cell or any other metric the advanced user wishes to use, making MCTS ideal to incorporate biological knowledge into retrosynthetic search.

3.4.6 Database sped-up calculations

When running a rule-based retrosynthetic algorithm, the most time and power consuming steps are rule application steps which require sub-graph matching, as this is an NP hard sub-graph isomorphism problem. We implemented a NoSQL database that allows for a frequent user to store rule application calculations. To allow for fair comparison between runs and algorithms, it was not activated in the results presented previously. However, when this feature is active, the results of the first rule application on a compound is stored in the database. When the same calculation is encountered

in a later run of RetroPath 3.0, results are retrieved from the database. This allows for faster runs of the algorithm and therefore larger and deeper exploration within the same time budget. For example, we ran the TPA retrosynthetic search 4 times: without the database, and with the database for the first, second and third time, allowing 1 hour and 100 000 iterations at each step. We present in Figure 3.4 the number of iterations performed in 1 hour, as well as the number of pathways found. While we can see the first run with the database is not as efficient as the run without it (reaches less iterations and does not find a pathway), we can see that having filled the database allows for more exploration of the tree in runs 2 and 3, where in the same allocated time more iterations are performed and more pathways found.



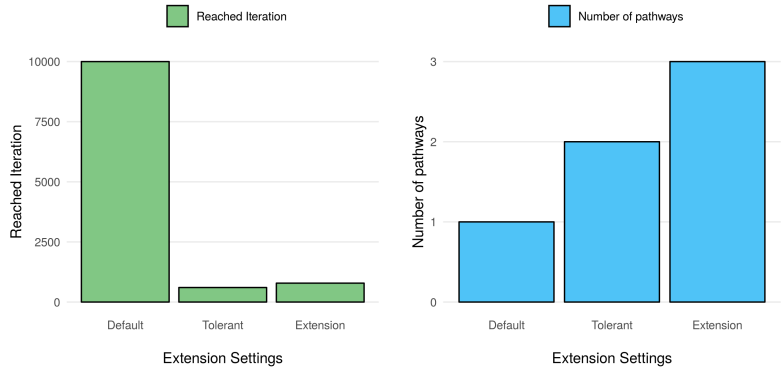
(a) (b)

Figure 3.4 **Database sped-up retrosynthetic search** We compared results between a classical run where computation is performed on the fly versus storing results in a Database. **a** Reached iteration is the number of iterations performed by the algorithm in the given time-frame (1hour). **b** Found pathways is the number of found pathways per run.

3.4.7 Extending a previous search

Another feature of interest for expert users is the possibility to extend a previously run tree. For example, considering the example of protopanaxadiol, the experimental pathway was not found with our default settings, but was with more tolerant settings (15 children instead of 10 and a score cut-off of 0.15 instead of 0.3). However, instead of starting from scratch and losing the previously made calculations, it is possible to restart the search from a saved tree, with more tolerant settings. The results after running for 4h and 10 000 iterations are presented in Figure 3.5. With the default settings, the iteration budget is spent finding only 1 pathway (not the experimental one). Using more tolerant settings is slower (604 iterations are performed in the allotted time) and 2 pathways are found. Extending the original tree

allows for performing more iterations (786) and finding one more pathway, when compared to starting from scratch.



(a) **(b)**

Figure 3.5 **Extending a previous search** We compared results between default settings, more tolerant settings and extending the saved searched. **a** Reached iteration is the number of iterations performed by the algorithm in the given time-frame (1hour). **b** Found pathways is the number of found pathways per run.

3.5 Conclusion

We provide here RetroPath 3.0, an open-source Python package for retrosynthesis. RetroPath 3.0 uses Monte Carlo Tree Search, a combinatorial search algorithm that changed the field of artificial intelligence when playing board games. While MCTS has already been implemented for chemical synthesis [87], RetroPath 3.0 is to our knowledge the first application of this algorithm to metabolic engineering. It is a versatile, modular command line tool, build for metabolic engineers, that takes as input a compound of interest, a chassis organism as sink and a set of reaction rules.

While RetroPath 3.0 can take as input any set of chemical rules in SMARTS, giving great freedom to users to use reactions from their pathways or microbial strain of expertise, the tests presented in this article use rules from our RetroRules database [116]. RetroRules was built using a data-driven approach showcased in RetroPath 2.0 [138], allowing promiscuity encoding. Using both reaction rules at various diameters and chemical similarity scoring, we can tune the allowed promiscuity within our search, which is an advantage over building reaction rules from EC numbers as is commonly done in metabolic engineering. Another advantage of a data-driven approach is that it is bound to get more precise and information-rich as metabolic databases expand.

We validated RetroPath 3.0 by verifying on a manually curated dataset that a literature-described pathways could be found for 20 different compounds.

The fact that the experimental pathways are found for 75% of compounds using default settings and 95% of the time using more tolerant ones, with much better results when biochemically guiding the search, confirms the ability of RetroPath 3.0 to suggest experimentally viable pathways for metabolic engineers.

Moreover, following the standards we set in our RetroPath 2.0 paper, we tested RetroPath 3.0 on a larger dataset and found a pathway for 83.6% of compounds that were results of successful metabolic engineering projects. These results confirm that RetroPath 3.0 generalizes well to metabolic engineering compounds outside the golden dataset. Moreover, RetroPath 3.0 suggests more pathways (median number of 11.5 versus 4.5), given metabolic engineers more suggestions with which to exercise their expertise.

While a restricted number of chassis was provided with RetroPath 3.0, the user can provide his own chassis or supplement the media with a given compound of interest. For example, when supplementing xylene to the chassis an experimentally described pathway to TPA [260] is found, and when adding thebaine to the media, a pathway for morphine production is also found [261]. A novel feature that should be welcomed by the metabolic engineering community is the ability of RetroPath 3.0 to suggest media supplements to complement the enzymatic synthesis plan. This feature, tested on 8 pathways, found the experimentally described supplement for 5 of them, and also suggests others supplements to test.

We showcase RetroPath 3.0's modularity by biasing the search towards less toxic intermediates, and also provide expert users 2 strategies to speed-up the retrosynthetic search, either by storing results in a database or extending a search from previously run search trees. Another feature of interest is the ability to use RetroPath 3.0 for biosensor design, as we also demonstrated with our previously developed tools [77, 262]. This can be used in conjunction with a dataset of detectable compounds [76] to allow for design of Sensing Enabling Metabolic Pathways.

Despite the advantages presented above, RetroPath 3.0 still presents some limits, mostly related pathway ranking. The authors of this article believe modular design, with a downstream analysis of selected pathways, to be a more appropriate course of action than ranking within the tool: so many genome or growth conditions modifications can increase a pathway yield that including all ranking schemes into one to come up with a 'best' pathway neglects both metabolic engineers' expertise and years of lessons from the industry. However, most bio-retrosynthesis developers present all integrated tools that perform pathway search and ranking together. Ranking schemes in such tools can involve accounting for enzyme integration into the chassis, kinetics, toxicity, carried flux, thermodynamics or preferred cofactor usage [252, 253]. Our approach provide two advantages over integrated

tools. First, ranking the pathways separately allows for a much more modular integration, and the ability to integrate new ideas into ranking much more easily. Moreover, as shown in the custom use section, for advanced users attached to using these schemes, it is possible to integrate them easily given RetroPath 3.0's modular code and the ability of the Monte Carlo Tree Search algorithm to bias the search towards or away from properties of interest.
While stereochemistry has been described as one of Nature's most interesting advantages compared to the traditional chemical industry [263], it is not used in results presented here, mainly due to current technical limitations to the way stereochemistry is handled by the cheminformatics packages we used at the moment. However, since our formalism uses rules SMARTS which are stereo-aware [108], and our standardisation schemes can leave or remove stereochemistry, users wishing to use stereochemistry in RetroPath 3.0 can do so.
Another major advance that could be included in RetroPath 3.0 would be to guide reaction selection steps through learned values instead of similarity. For example, this was implemented in [87] or [264]. However, the authors learned values from Reaxys which contains 12.4 million single-step reactions (compared to around 80k in Metanetx, including reactions without chemical structures [202]). Therefore, using learned values in bio-retrosynthesis seems out of reach for the moment, but could become available in the coming years due to the intense curation efforts under way in the community. Moreover, those learned values in bio-retrosynthesis would have to be chassis-dependent, as a intermediate compound's value for bio-retrosynthesis depends highly on the chassis of interest.
In conclusion, we present here a highly modular tool using one of the latest tree search algorithms for bio-retrosynthesis. This tool is modular enough for expert users to input their expert knowledge, and has been thoroughly tested on datasets of interest to the community.

3.6 Materials and methods

All chemical operations were performed using RDKit release 2019.03.1.0 and Python 3.6.

3.6.1 Compound standardization

All compounds were standardized using the following steps:

1. sanitizing chemical depictions using RDKit's SanitizeMol method
2. removing isotope
3. neutralizing charges
4. removing stereo
5. converting back and forth to InChI [265] to ensure tautomerism consistency

3.6.2 Reaction rule encoding

Our reaction rules are generated as presented in our RetroRules database [116]. Briefly, we extracted known biochemical reactions from the MetaNetX [202] database version 3.1 and filtered out incomplete reactions. We identified the reaction centers based on atom-atom mappings we performed using the Reaction Decoder software [195](version 2.1.0). We decomposed multi-substrates reactions into mono-substrate components by considering one substrate per component and only the subset of products that share at least one atom with the substrate. Mono-components involving a typical co-factor (such as CO2, ATP, NADH, ...) as substrate were excluded. Finally, each mono-component reaction is encoded into a collection of reaction rules using the SMARTS [108] formalism for a diameter ranging from 2 to 16 around the reaction center.
The main difference with the RetroRules procedure is that we use implicit hydrogen notation in the reaction rules instead of explicit, which allows much faster computation of standardization of compounds and reaction rule application through RDKit.
To validate the reaction rules, we checked that applying the reaction rules on the substrates used as templates produce the products described in the template reactions. The success rates range from 99.4% for reaction rules at diameter 2 to 99.8% for those at diameter 16. We also performed a cross-radius consistency check by ensuring that sets of products produced by large diameter reaction rules (e.g. reaction rules generated with a diameter of 10) are always subsets of products produced by reaction rule at any smaller diameter (e.g. reaction rules for diameter 8), which show a success rate of 99.3%. Only reaction rules passing successfully both procedures have been retained and used, which represent almost 146k distinct reaction rules (available for download on RetroRules website, download page, release 20190524) and model more than 18k biochemical reactions in both directions.

3.6.3 EMS generation for branching factor calculation

We generated what we called the Extended Metabolic Space (EMS) at 1 step, i.e. the metabolic space that can be reached by applying our reaction rules once on all compounds of MetaNetX [202] being used as template for generating at least one reaction rule. We filtered out substrates having a molecular weight greater than 1 kDa. We set a timeout cut-off of 1 second on rule application. We then analyzed the results presented in Tables 3.1 and 3.2 using the NetworkX python software [266].

3.6.4 Chemical score calculation

After compound standardization, we calculate a 1024 binary Morgan fingerprints vector using the RDKit method GetMorganFingerprintAsBitVect at diameter 4.
For substrate score, the query substrate is compared to all native substrates a rule was learned on using Tanimoto score [196, 267], and the maximum score is kept.
For products and substrate scoring, the procedure is as follows: For each native (n_{sub}, $n_{products}$) couple:

1. calculate Tanimoto of query and native substrate.
2. generate all combinations of native and query products (below 1000 to avoid a combinatorial explosion. The number of combinations is $n!$ where n is the number of products generated by the rule application)
3. for each combination, calculate the geometric mean of the Tanimoto scores of products
4. Keep the highest combination score.
5. The score of this native combination is the product of the substrate score and the highest combination score.

The score of the rule is the highest score of all native substrates and products the rule was generated from. Rules generating a different number of products from the template receive a score of -1.

3.6.5 Biological score calculation

A penalty is calculated as presented in our RetroPath 2.0 paper [138]. Briefly, we clustered reaction rules according to the EC number annotations inherited from the template reactions. Independently, we clustered

enzyme sequences collected from UniProt [268] (release 2019.04) according to sequence similarity using the cd-hit software [204] (version 4.6.8). We establish the sequences to reaction rules associations based on Rhea [269] (v98), MetaCyc [270] (v21.5) and Reactome [271] (v66) cross-links. The penalty score was then computed as

$$penalty_{score}(rule) = log10(n_{rule})$$

with n_{rule} the number of distinct clusters that contains sequences associated to the rule.
In addition, we then normalized this penalty into a score comprised between 0 and 1 using the following function:

$$score(rule) = \frac{1}{1 + alpha.weighted_{penalty}(rule)}$$

with

$$weighted_{penalty}(rule) = \frac{penalty_{score}(rule)}{\sqrt{radius(rule)}}$$

and $alpha = \frac{1}{median(weighted_{penalty})}$.

3.6.6 Sinks construction

Except when stated, sink compounds have been extracted from genome-scale metabolic models by only collecting the chemicals that lie in the cytosol compartment. In addition, we filtered out "dead-end" compounds, i.e. compounds that cannot be produced by any reactions in a steady-state metabolic model, that we detected by performing a Flux Variability Analysis using the COBRApy package [272] (v0.15.3).
Chemical structures have been obtained using cross-links from the models to metabolic databases. In case no cross-link or not any valid structure was found, the PubChem [273] database was examined using compound names as query. Finally, all chemical structures were standardized as described in the Compound standardisation section.

3.6.7 Available sinks

The available sinks provided with our software are iML1515 [274], iJO1366 [216] and core *E. coli* metabolism [275], as well as Bacillus Subtilis iYO844 model [118] and our set of detectable compounds for biosensor design [76]. The genome scale models were obtained from the BiGG Models database [276]. By default, we used sinks from the iML1515 model.

3.6.8 RetroPath 2.0 configuration

For all tests made with RetroPath2.0, we perform all executions on a recent work station. We use the sink extracted from the iML1515 model, and we set the maximum pathway length to 5, the maximum number of structures to keep for next iteration to 1000, and a 3 hours per execution time budget. We then use the rp2paths software available on GitHub to extract the pathways from RetroPath2.0 output [138].

3.6.9 Monte Carlo Tree Search implementation

The aim of Markov decision processes is to model sequential decision processes of an agent in an environment [277]. Its most notable components are states (representing positions in a game) and actions (allowed transformations from the state). In RetroPath 3.0, following the method developed by [87], we consider states to be a set of molecules. The initial state contains only the target compound one desires to produce. Actions are transformation of any molecule of the state that do not transform the compound into itself nor produce a non-sink compound that has already been produced before in the synthesis plan so as to avoid loop searches. A state is considered terminal if all compounds are in the sink, if no move can be applied to this state or if the maximum allowed depth has been reached. Monte Carlo Tree Search is a reinforcement learning approach that builds a search tree and stochastically explores search space to bias search towards most promising regions of combinatorial space, following steps presented in Figure 3.1B and detailed below.

Selection Starting from the root node, the best children nodes according to the selection policy are iteratively chosen until a leaf node is reached. The formula we used is:

$$Value = \frac{Node_{score}}{Node_{visits}} + UCTK.chemical_{score}.biological_{score}.\sqrt{\frac{Parent_{visits}}{1 + Node_{visits}}}$$

where $Node_{score}$ is the cumulative score from rollouts, $Node_{visits}$ is the number of visits to this node, $UCTK$ is the UCT (Upper Confidence Trees) constant used (balances between exploitation and exploration), chemical and biological scores are the scores of the move leading to this node and parent visits is the number of visits of this Node's parents.
Other policies have been developed, notably with only chemical, biological score, or no scoring. In our implementation, grand-children of a node can

only be explored if all his children have had at least $minimal_{visits}$ number of visits. This allows mandatory rollout on different branches at least once to favor exploration.

Expansion For each compound that is not in the sink, its n best moves are identified and stored (with n the maximal number of children allowed for the node). Then, the n best moves overall on the state are selected and children created iteratively (one at the first visit of the node, the next at the next visit and so on) for each of these moves and a rollout is performed.

Rollout Rollout is an iterative procedure that starts by checking the status of the state. Is it is terminal, a reward (or penalty) is returned according to the rewarding policy. If it is not terminal, a transformation is randomly sampled from available transformations and the process is repeated. This is performed until a maximum number of rollout steps or the maximal depth of the tree is reached. The function for random sampling used throughout this study gives weight chemical * biological score to moves, therefore giving more probability of being sampled to higher scoring moves according to our scoring scheme.

Rewarding policy A state is rewarded as follows:

1. receives a penalty of -1 when no compound is solved
2. receives a bonus of 5 when the state is fully solved.
3. receives a score of $\frac{number_{found}}{total_{number}}$ when only a fraction of molecules in the state are solved.

Update The node and its parents update their value and node counts according to the results obtained from the above rewarding scheme after rollout.

3.6.10 Returning complete pathways

Each time a full pathway is found during the tree expansion, the pathway is returned, and an additional bonus of 10 is received by the node, to allow for biasing towards similar successful pathways. At the end of search, the most visited pathway is returned ("best"), and all pathways are returned ranked in order of decreasing biochemical score.

3.6.11 Rule calculation cache using a NoSQL database

Rule calculation can be optionally cached into a NoSQL database in order to optimize the running time of RetroPath 3.0. We released this cache system as an optional python package named rp3_cache that is also available on GitHub. Technically, the cache system relies on the Mongo DB database that is embedded into a Docker container to make the implementation agnostic of the operating system.

3.6.12 Tree extension

The tree extension procedure starts with a tree containing search results. If n children were allowed in the first run and the extension allows m more children, up to $n + m$ children can be found for a given node. Node scores and visit counts are re-initialized, and nodes are first flagged for extension then extended when they are first visited in the new search.

3.6.13 Golden dataset construction

In order to perform golden dataset curation, we focused on articles where pathways were explicitly described (i.e.: no missing steps, and available intermediate compounds). We retrieved compound structures in PubChem [273] and EC numbers from BRENDA [133] based on enzyme name as given in the article. We selected pathways of strictly more than 1 step. We then verified for each step that we had chemical rules available in our RetroRules [116] database to encode the described transformations. This ensures a fair comparison between tools using our publicly available reaction rules, in order to evaluate separately retrosynthesis tools and the underlying chemical rules. The detailed list with references is available as Supplementary Table 3.5, and the pathways as Supplementary Data 1 available online.

3.6.14 Experimental pathway comparison

In order to compare the pathways found by RetroPath 2.0 or 3.0 and the experimental pathways described in the literature, we have two types of information: compound identity and EC number identity. We consider the reaction EC number to be equal if it is identical up to 3 digits (1.1.1.x is identical to 1.1.1.y). Since spontaneous reactions do not have an EC number,

we used compound identity for comparison, and EC number as additional information.

3.6.15 Laser retrieval and Metabolic Engineering completion

We build the LASER dataset by parsing target molecule and chassis information from the LASER database published by Winkler *et al.* which provides a curated list of more than 600 successfully implemented metabolic designs [258, 259]. When available, we store the chemical structure provided by MetaCyc (v23.0), otherwise we query the PubChem database based on target compound names. We augmented this list with target compounds reported in the Metabolic Engineering journal in 2016 (volumes 33 to 38) and published in RetroPath 2.0 [138]. All chemical structures have been standardized using the procedure described in the Compound standardisation section. The final dataset used in the present paper contains 211 unique structures that are provided as Supplementary Data 2 available online.

3.6.16 Extracting supplements from LASER

LASER provides a 'Media' line that contains addition to the media, extracted using Natural Language Processing. This can include antibiotics, promoter inducers or supplements of interest required to build the pathway. We removed all compounds that did not concern pathway supplements, and removed early precursor supplementation. Structures were obtained from PubChem. We obtain a list of only 6 pathways satisfying these requirements, and 8 when we also add two pathways for TPA from literature [278, 260]. This list is available as Supplementary Data 4 available online.

3.6.17 Supplement finder feature

The Supplement finder functions as follows:

1. load search tree in memory
2. explore all nodes and keep compound structures that complete of a chemical state (i.e.: all other compounds of the state are solved)
3. compounds that allow for completion of more than N states (which would complement N pathways) are kept. Here N= 1: any compound that can complement a pathway is kept.

4. compounds are filtered according to presence in a Database of interest. Here, we filtered according to presence in the Sigma catalogue.
5. We keep the N best suggestions (according to number of pathways that are completed). Here, we returned up to 20 potential supplements.
6. All completed pathways are extracted for future analysis.

3.6.18 Toxicity implementation

We used data from EcoliTox [211] and XTMS [188] to build a QSAR model (using as input features 1024 binary Morgan fingerprints vector calculated with the RDKit method GetMorganFingerprintAsBitVect at diameter 4), predicting log(IC50) of the compounds. We train our model using scikit-learn (version 0.19.1) [279]. The model is a multi-layer perceptron trained with the default parameters from scitkit-learn except the following parameters: maximum iteration of 20000, adaptive learning rate, adam solver, early stopping and the following layers: (10, 100,100, 20). This model has a Leave-One-Out (LOO) score of 0.81. In prediction mode, toxicity was used only if the predicted log(IC50) <= 0. The modified UCT function that was used is:

$$Value = \frac{Node_{score}}{Node_{visits}} + bias_k . \frac{toxicity}{1 + Node_{visits}} + UCTK . chemical_{score} . biological_{score} . \sqrt{\frac{Parent_{visits}}{1 + Node_{visits}}}$$

3.6.19 Hardware

We ran our tests on 2 calculation clusters and 1 personal computer. Tests involving using the NoSQL Database and media supplementation were run on a desktop computer with the following characteristics: CPU is Intel(R) Xeon(R) W-2145 CPU @ 3.70GHz and 32G of RAM. Tests for the golden dataset evaluation were performed on the Migale cluster (2.0 to 2.4 GHz CPUs). Allocated resources per job were set to 1 vCPU and 20 GBs. Tests for the Laser database evaluation were run on the IFB cluster (2.3GHz CPUs). Allocated resources per job were set to 1 vCPU and 40 GBs. Cluster tests were run using snakemake 5.4.0 [280].

3.7 Supplementary Tables

Diameters used	2	4	6	8	10	12	14	16
Average number of applicable rules	797	227	73	35	20	14	11	8

Table 3.2 **Average branching factor for individual diameters** Average number of rules that apply to a compound according to the diameter at which the rule is used. This shows that more promiscuous rules (at lower diameters) generate a bigger combinatorial space.

Compound	**Total pathway number**		**Iteration for experimental pathway**		**Rank of experimental pathway**	
	Default	*Toxicity*	Default	*Toxicity*	Default	*Toxicity*
3-methyl-1-butanol	1	*1*	4	*4*	1	*1*
1,4-Butanediol	NA	*NA*	NA	*NA*	NA	*NA*
2-amino-1,3-propanediol	1	*1*	1	*1*	1	*1*
2,5-DHBA	34	*31*	88	*97*	4	*4*
benzyl alcohol	13	*22*	1815	*2213*	4	*8*
caroten	4	*4*	6508	*6443*	1	*1*
cis,cis muconate	11	*11*	159	*176*	6	*6*
glutaric acid	61	*74*	1114	*571*	13	*7*
lycopene	106	*103*	101	*96*	1	*1*
mesaconic acid	14	*14*	9	*9*	1	*1*
naringenin	38	*40*	408	*1011*	1	*9*
N-methylpyrrolinium	9	*10*	NA	*NA*	NA	*NA*
p-hydroxystyrene	59	*61*	34	*33*	1	*1*
piceatannol	34	*38*	8581	*8996*	32	*36*
pinocembrin	25	*25*	66	*55*	2	*2*
protopanaxadiol	1	*2*	NA	*NA*	NA	*NA*
styrene	43	*45*	14	*13*	2	*2*
TPA	9	*10*	NA	*NA*	NA	*NA*
vanillin	34	*36*	169	*171*	4	*4*
violacein	3	*3*	158	*158*	3	*3*

Table 3.3 **Toxicity biased results** Comparing results between the default configuration and the one using the toxicity score to bias the search. Total pathway number is the total number of pathway found with the given time and iteration budget. The iteration and rank for the experimental pathway refer to the pathway described in the golden dataset.

Parameter name	Golden Default	Golden Rescue	Laser Default	Laser Rescue
Itermax	10000	10000	50000	10000
Expansion width	10	15	10	10
Time budget (s)	10800 when comparing to RP2, 14400 otherwise	10800 when comparing to RP2, 14400 otherwise	10800	10800
Max depth	7	7	10	7
UCT Policy	Biochemical	Biochemical	Biochemical	Biochemical
UCTK	2	2	2	2
Rollout Policy	Uniform on biochemical score	Uniform on biochemical score	Uniform on biochemical score	Uniform on biochemical score
Max rollout	2	2	2	2
Chemical scoring	Substrate and product	Substrate and product	Substrate and product	Substrate and product
Virtual visits	0	0	0	0
Rule diameters	6, 10, 16	6, 10, 16	6, 10, 16	2, 6, 10, 16
Biological score cut-off	0.3	0.15	0.3	0
Substrate score cut-off	0.3	0.15	0.3	0
Chemical score cut-off	0.3	0.15	0.3	0
Minimal visits count	1	1	1	1
Fire timeout (s)	1	1	1	1
Penalty	-1	-1	-1	-1
Reward	5	5	5	5
Full pathway reward	10	10	10	10
Seed	42	42	42	42

Table 3.4 **RP3 configuration** Detailed configuration data for RetroPath 3.0 runs on validation datasets.

Name	InChI	Reference
3-methyl-1-butanol	InChI=1S/C5H12O/c1-5(2)3-4-6/h5-6H,3-4H2,1-2H3	[281]
1,4-Butanediol	InChI=1S/C4H10O2/c5-3-1-2-4-6/h5-6H,1-4H2	[282, 191]
2-amino-1,3-propanediol	InChI=1S/C3H9NO2/c4-3(1-5)2-6/h3,5-6H,1-2,4H2	[283]
2,5-DHBA	InChI=1S/C7H6O4/c8-4-1-2-6(9)5(3-4)7(10)11/h1-3,8-9H,(H,10,11)	[284]
benzyl alcohol	InChI=1S/C7H8O/c8-6-7-4-2-1-3-5-7/h1-5,8H,6H2	[285]
caroten	InChI=1S/C40H56/c1-31(19-13-21-33(3)25-27-37-35(5)23-15-29-39(37,7)8)17-11-12-18-32(2)20-14-22-34(4)26-28-38-36(6)24-16-30-40(38,9)10/h11-14,17-22,25-28H,15-16,23-24,29-30H2,1-10H3/b12-11+,19-13+,20-14+,27-25+,28-26+,31-17+,32-18+,33-21+,34-22+	[286]
cis,cis muconate	InChI=1S/C6H6O4/c7-5(8)3-1-2-4-6(9)10/h1-4H,(H,7,8)(H,9,10)/p-2/b3-1-,4-2-	[287]
glutaric acid	InChI=1S/C5H8O4/c6-4(7)2-1-3-5(8)9/h1-3H2,(H,6,7)(H,8,9)	[288]
lycopene	InChI=1S/C40H56/c1-33(2)19-13-23-37(7)27-17-31-39(9)29-15-25-35(5)21-11-12-22-36(6)26-16-30-40(10)32-18-28-38(8)24-14-20-34(3)4/h11-12,15-22,25-32H,13-14,23-24H2,1-10H3/b12-11+,25-15+,26-16+,31-17+,32-18+,35-21+,36-22+,37-27+,38-28+,39-29+,40-30+	[289]
mesaconic acid	InChI=1S/C5H6O4/c1-3(5(8)9)2-4(6)7/h2H,1H3,(H,6,7)(H,8,9)/b3-2+	[290]

naringenin	InChI=1S/C15H12O5/c16-9-3-1-8(2-4-9)13-7-12(19)15-11(18)5-10(17)6-14(15)20-13/h1-6,13,16-18H,7H2	[291]
N-methylpyrrolinium	InChI=1S/C5H10N/c1-6-4-2-3-5-6/h4H,2-3,5H2,1H3/q+1	[292]
p-hydroxystyrene	InChI=1S/C8H8O/c1-2-7-3-5-8(9)6-4-7/h2-6,9H,1H2	[293]
piceatannol	InChI=1S/C14H12O4/c15-11-5-10(6-12(16)8-11)2-1-9-3-4-13(17)14(18)7-9/h1-8,15-18H/b2-1+	[294]
pinocembrin	InChI=1S/C8H8/c1-2-8-6-4-3-5-7-8/h2-7H,1H2	[295]
protopanaxadiol	InChI=1S/C30H52O3/c1-19(2)10-9-14-30(8,33)20-11-16-29(7)25(20)21(31)18-23-27(5)15-13-24(32)26(3,4)22(27)12-17-28(23,29)6/h10,20-25,31-33H,9,11-18H2,1-8H3/t20-,21+,22-,23+,24-,25-,27-,28+,29+,30+/m0/s1	[97]
styrene	InChI=1S/C15H12O4/c16-10-6-11(17)15-12(18)8-13(19-14(15)7-10)9-4-2-1-3-5-9/h1-7,13,16-17H,8H2/t13-/m0/s1	[296]
TPA	InChI=1S/C8H6O4/c9-7(10)5-1-2-6(4-3-5)8(11)12/h1-4H,(H,9,10)(H,11,12)	[278, 260]
vanillin	InChI=1S/C8H8O3/c1-11-8-4-6(5-9)2-3-7(8)10/h2-5,10H,1H3	[297]
violacein	InChI=1S/C20H13N3O3/c24-10-5-6-15-12(7-10)14(9-21-15)17-8-13(19(25)23-17)18-11-3-1-2-4-16(11)22-20(18)26/h1-9,21,24H,(H,22,26)(H,23,25)/b18-13+	[298, 299]

Table 3.5 **Golden dataset structures** This dataset contains the compounds, structures and references used for experimental pathway analysis presented in Results - golden set.

3.8 Supplementary Note 1: Parameters' Role and Effects

The aim of this supplementary note is to detail the different parameters available in RetroPath 3.0 and their roles and effects.

A number of methods and ideas were taken or inspired from the following master thesis, which uses Monte Carlo Tree Search against a computer game [300].

3.8.1 Rule selection parameters

Biological score: As described in the main text, this scores characterizes our confidence that a sequence exists to catalyze the reaction of interest. It is normalized between 0 and 1. Using a cut-off on this score removes less trustworthy reaction rules. We see in Supplementary Figure 3.6 that results do not vary between using a cut-off from 0 to 0.3, and we start losing pathways of interest when the cut-off is superior or equal to 0.5. A cut-off of 0.3 therefore seems to be a good trade-off between confidence in the existence of a sequence and obtaining enough retrosynthesis results.

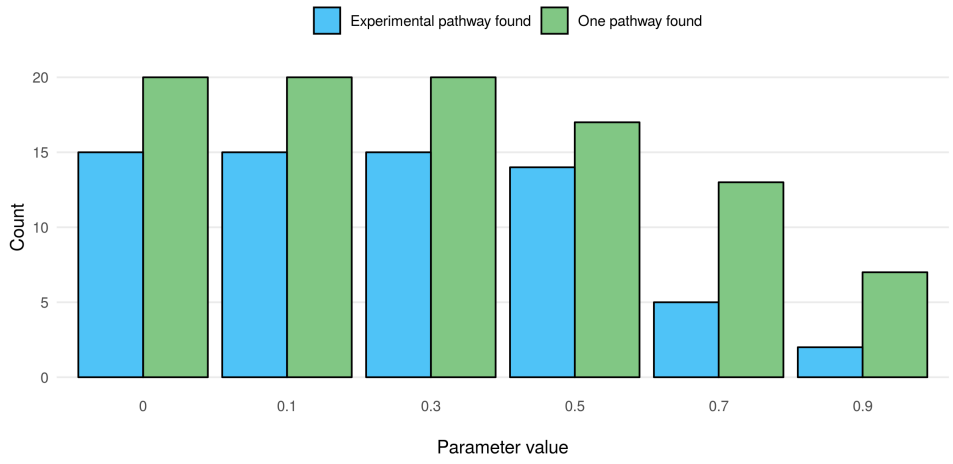


Figure 3.6 **Impact of biological score cut-off on retrieval performance of RetroPath 3.0** We compared results between using a biological score cut-off varying between 0 and 0.9. One pathway found means that at least one pathway have been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

Chemical score: As described in the main text, this score characterizes our confidence that the reaction rule learned on a substrate and product from a database of interest can truly be applied to a new substrate, based on

similarity between substrates and products of the native reaction versus the query reaction. We can see in Supplementary Figure 3.7 that allowing reactions that are too different leads the tree to explore too diverse pathways, while being too conservative leads to loss of useful reactions. 0.3 therefore seems to be a good cut-off between confidence that the reaction rule does apply to the compound and allowing exploration of the metabolic space.

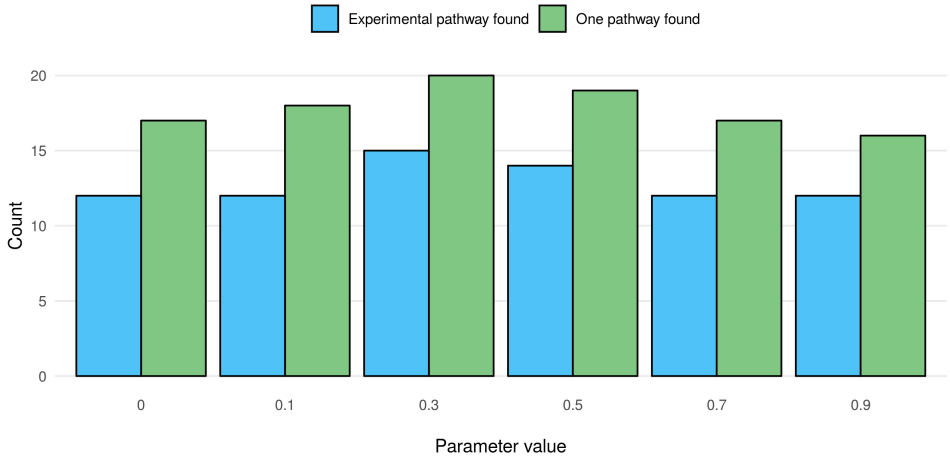


Figure 3.7 **Impact of chemical score cut-off on retrieval performance of RetroPath 3.0** We compared results between using a chemical score cut-off varying between 0 and 0.9. One pathway found means that at least one pathway have been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

Biological and chemical score: Here we varied both the chemical and biological scores, set at the same value. We can see in Supplementary Figure 3.8, as in Supplementary Figures 3.6 and 3.7, that cut-offs of 0.3 provide the best trade-off between exploration and confidence.

Diameters: Diameters characterize the degree of promiscuity we allow in a reaction rule: higher diameters are more specific, while lower diameters are more promiscuous. We found a good trade-off was to allow rules at different levels of promiscuity, using diameters of 6, 10 and 16 (low, medium and high specificity), as shown in Supplementary Figure 3.9.

UCT policy: While these policies are used to tune the exploration/exploitation balance, we modified it to guide the search and see the importance of that guidance on finding results. We can see in Figure 3.3 the best UCT policy to guide our exploration of the metabolic space is our formula including the biochemical score.

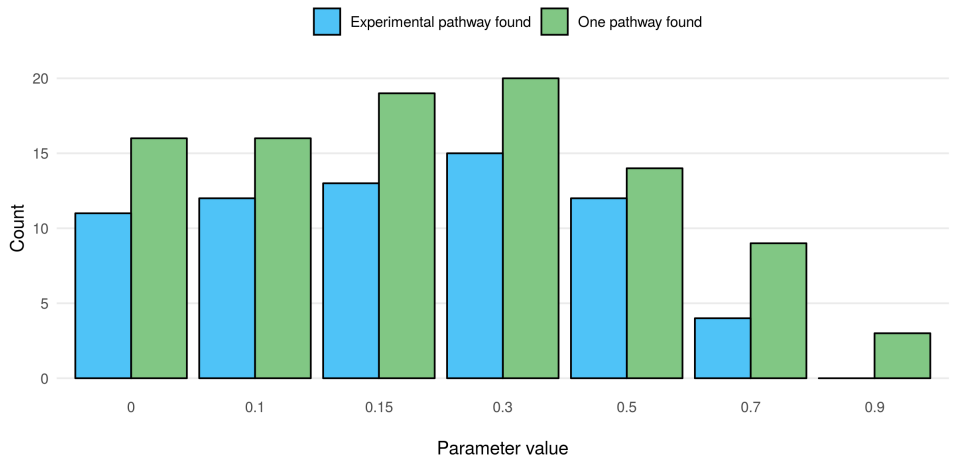


Figure 3.8 **Impact of biochemical score cut-off on retrieval performance of RetroPath 3.0** We compared results between using chemical score cut-off and biological score cut-off (set at the same value) varying between 0 and 0.9. One pathway found means that at least one pathway have been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

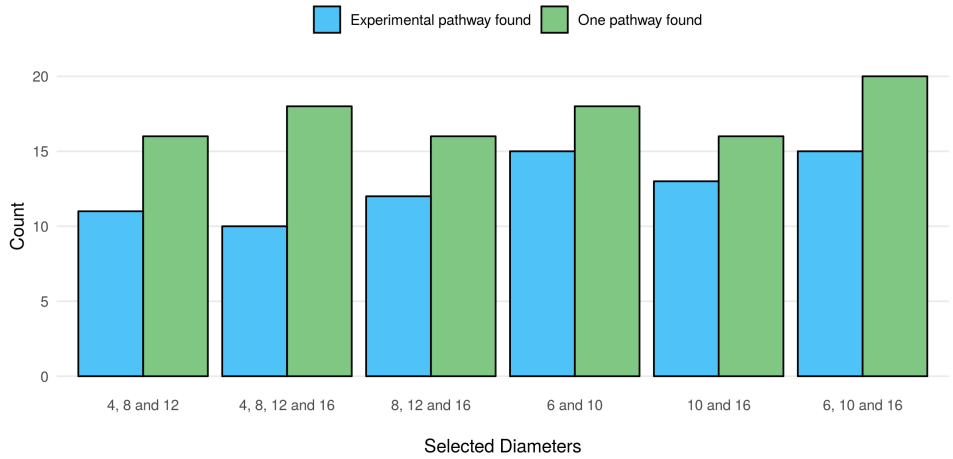


Figure 3.9 **Impact of allowed rule diameters on retrieval performance of RetroPath 3.0** We compared results between using different diameter sets. One pathway found means that at least one pathway have been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

3.8.2 Exploration parameters

Expansion width: It is the number of children a node is allowed to have. We found 10 and 15 provided a good trade-off between exploration and exploitation, as shown in Supplementary Figure 3.10. We usually tested with 10 children, and expanded to 15 for failed compounds.

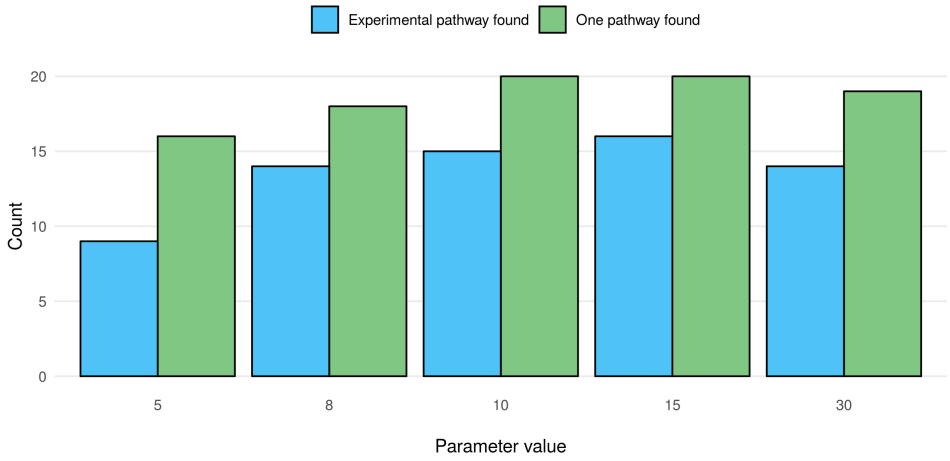


Figure 3.10 **Impact of expansion width on retrieval performance of RetroPath 3.0** We compared results between using different expansion width (number of allowed children per node). One pathway found means that at least one pathway have been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

Expansion width: In our implementation, grand-children of a node can only be explored if all his children have had at least $minimal_{visits}$ visits, where this value was set at 1 in the default settings. This allows mandatory rollout on different branches at least once to favor exploration. We can see from Supplementary Figure 3.11 that results are similar when not forcing this exploration with a parameter set to 0.

Rollout: This is the rollout depth: the number of reactions performed before analyzing the state and returning the state's reward or penalty. Supplementary Figure 3.12 shows that rollout depth does not impact our capacity of finding experimental results on the golden dataset. However, un-shown results (taking into account the iteration at which those results are found) suggest best rollout values are either 2 or 3.

UCTK: This constant balances the trade-off between exploration and exploitation in the UCT formula. We can see in Supplementary Figure 3.13 that the value allowing the best retrieval rate from the golden dataset is a constant of 2.

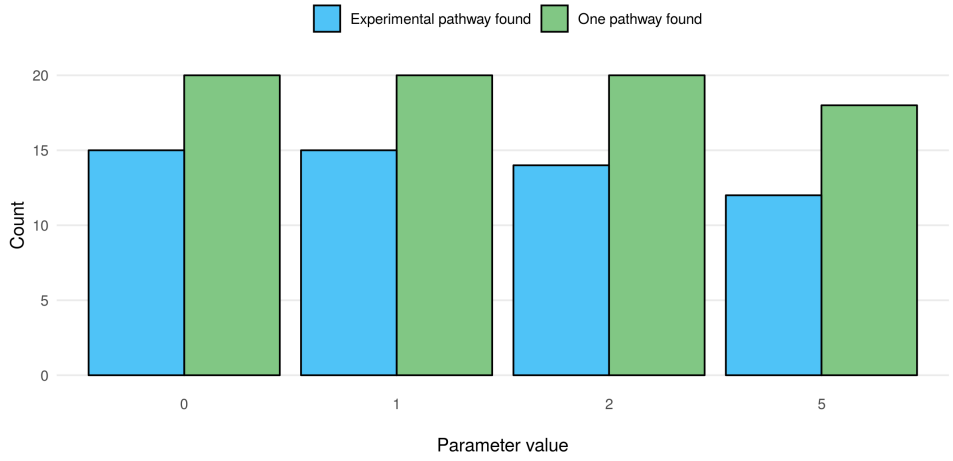


Figure 3.11 **Impact of minimal visits count on retrieval performance of RetroPath 3.0** We compared results between using different minimal visit count values. One pathway found means that at least one pathway have been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

Virtual visits: This is the number of visits a new node starts with. The concept of virtual visits is that giving an initial value to a node will return more stable rollout results as they will be smoothed by a number less close to 0 rather than being very stochastic at low values. We can see this strategy did not give better results in our MCTS for bioretrosynthesis in Supplementary Figure 3.14.

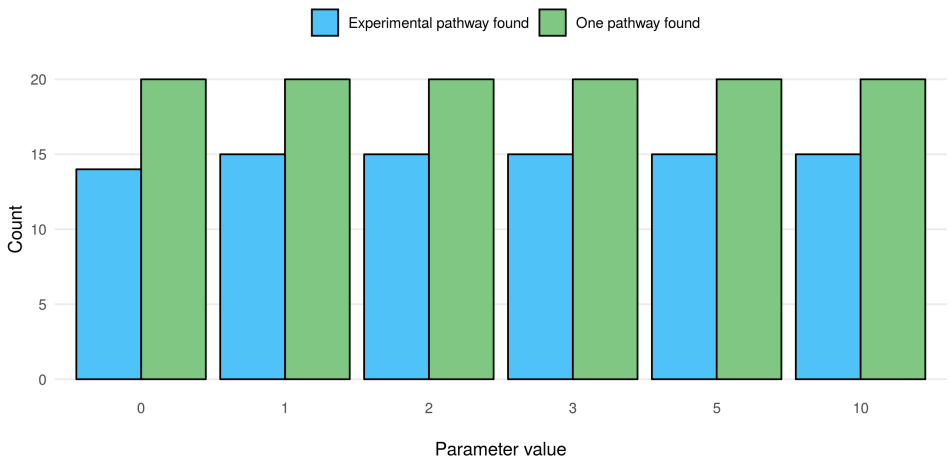


Figure 3.12 **Impact of rollout depth on retrieval performance of RetroPath 3.0** We compared results between using different rollout depths. One pathway found means that at least one pathway have been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

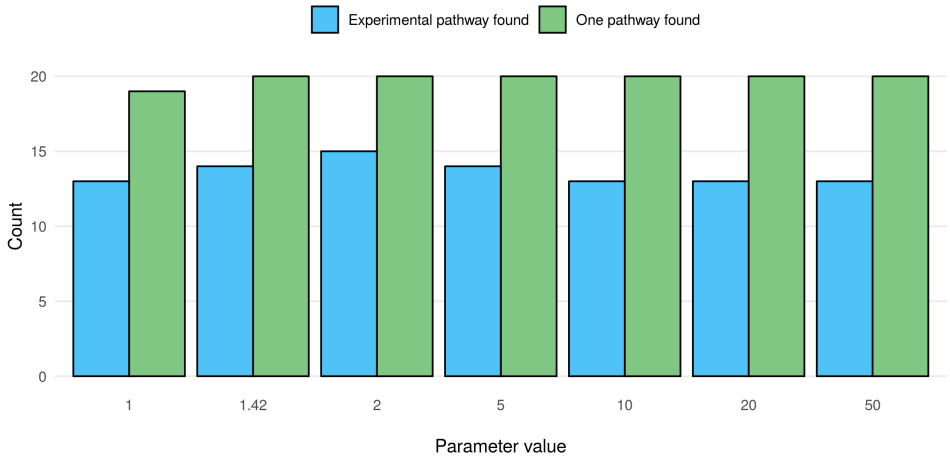


Figure 3.13 **Impact of exploration constant value (UCTK) on retrieval performance of RetroPath 3.0** We compared results between using different exploration constant values (UCTK). One pathway found means that at least one pathway have been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

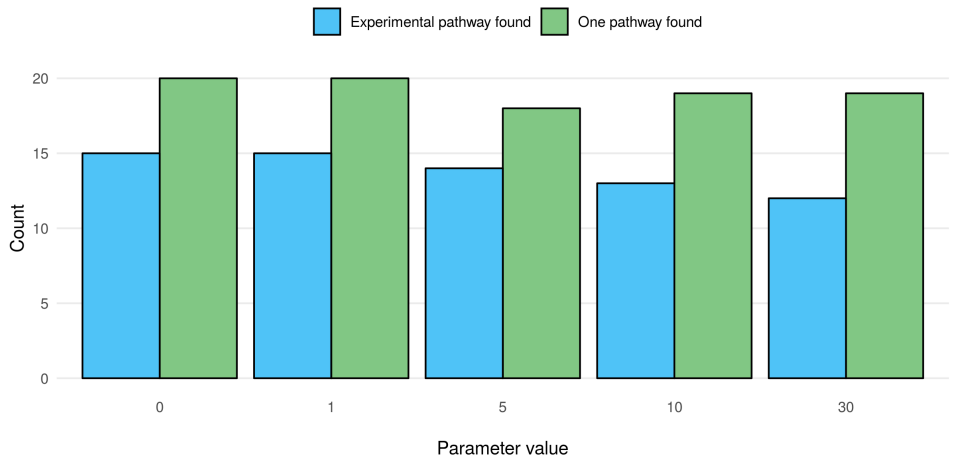


Figure 3.14 **Impact of virtual visits on retrieval performance of RetroPath 3.0** We compared results between using different virtual visits numbers. One pathway found means that at least one pathway have been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

3.8.3 Solution rewarding

Penalty: This is the value returned when no compound of the state is within the chassis (including at the end of rollout). We can see in Supplementary Figure 3.15 that increasing penalty does not yield better results in our case, and a value of -1 penalizes enough the unsuccessful rollout results.

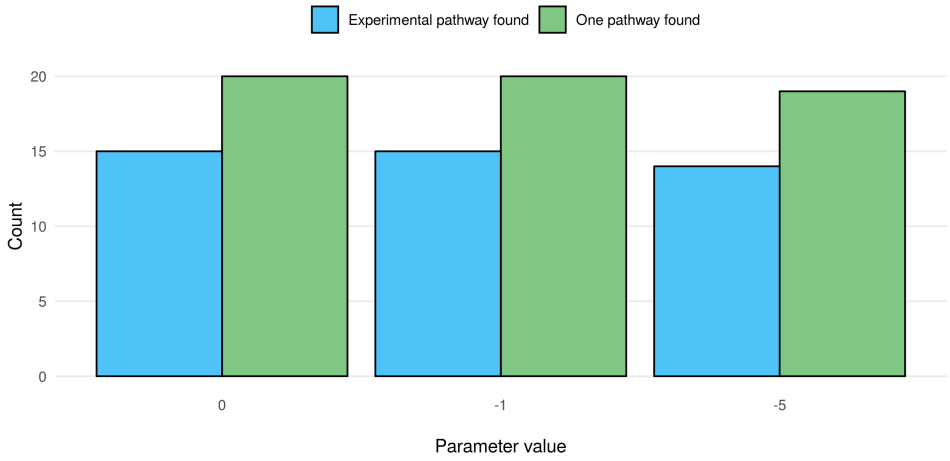


Figure 3.15 **Impact of penalty on retrieval performance of RetroPath 3.0** We compared results between using different penalties. One pathway found means that at least one pathway have been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

Reward: Reward is the value returned when all compounds of the state are solved, to encourage exploration of the same area of the Tree. We can see from Supplementary Figure 3.16 that a value of 5 provides a good trade-off between exploration of other areas of the tree and exploitation of promising regions.

3.8.4 Other parameters - for other applications

Heavy saving: Saves search Tree state during the search instead of only at the end. Used to analyze Search evolution.

Stop at first result: The search stops once a single result is found.

Fire time-out and standardization time-out: Timeouts on rule application on a compound.

Organism name: Choose another organism from our predefined list of sinks.

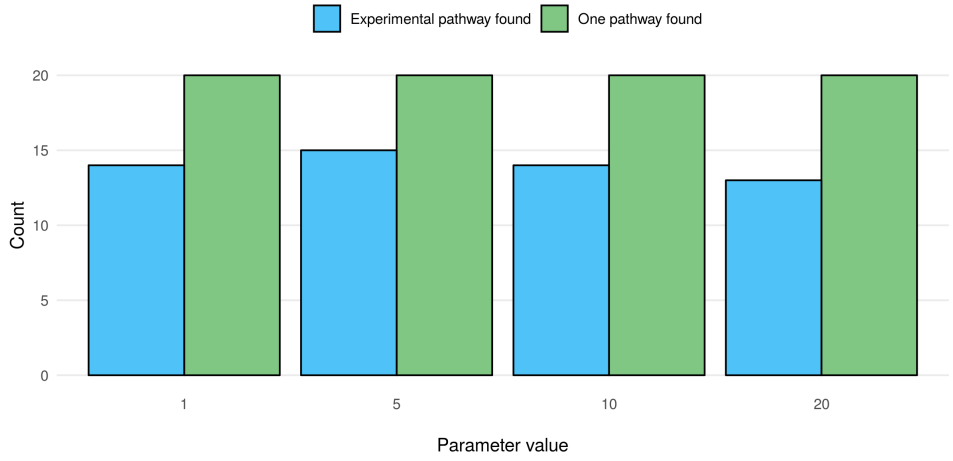


Figure 3.16 **Impact of reward on retrieval performance of RetroPath 3.0** We compared results between using different rewards. One pathway found means that at least one pathway have been predicted. Experimental pathway found means that the experimental pathway is from among the predicted pathways.

Complementary sink: Adding compounds to the sink as supplements. Can also be used to provide an entirely new sink following the required format.

3.8.5 Other parameters - exploratory

Remark: No detailed comparison was performed in this article on those parameters, contrary to the parameters presented above.

k_{RAVE}: For balancing of Rapid Action Value Estimation. The idea is to provide moves with results from rollouts elsewhere in the Tree, to give them an initial value. This will decrease in importance as the node itself is visited, but provides a fast initial value.

$Bias_k$: When using bias (for example towards toxicity), how to weight this value.

Progressive bias:: When used in conjunction with $bias_k$, can give an initial value to a node based on various policies: high reward, current state reward, no reward... This also helps initial estimation of the node value rather than rely only on costly rollouts.

Progressive widening: Allow a number of children different according to the number of visits of the node. This is to avoid expanding too much in spaces of the tree search that are actually not interesting.

4

A dataset of small molecules triggering transcriptional and translational cellular responses

This work was published in Data In Brief by Mathilde Koch, Amir Pandi, Baudoin Delépine and Jean-Loup Faulon.
Only minor modifications to the published paper have been introduced in the Chapter below.

Detailed contribution to this thesis

The aim of this Chapter within the context of development of synthetic metabolic circuits is the following: in combination with retrosynthetic algorithms as presented in Chapters 1 and 3, a dataset of detectable compounds can be used to find an enzymatic pathway to convert an undetectable compound into a detectable one. This strategy has been used multiple times in the second part of this PhD, in Chapters 9 and 10. This study was initially designed by myself, B.D. and A.P. Both other authors contributed to the literature review, and I assembled the final dataset, extended the literature review and wrote the manuscript.

Full reference

Koch M., Pandi A., Delépine B., Faulon J.-L. (2018) A dataset of small molecules triggering transcriptional and translational cellular responses *Data in Brief*, 10.1016/j.dib.2018.02.061.

Contributions as stated in the article

Not available.

4.1 Abstract

The aim of this dataset is to identify and collect compounds that are known for being detectable by a living cell, through the action of a genetically encoded biosensor and is centered on bacterial transcription factors. Such a dataset should open the possibility to consider a wide range of applications in synthetic biology. The reader will find in this dataset the name of the compounds, their InChI (molecular structure), the publication where the detection was reported, the organism in which this was detected or engineered, the type of detection and experiment that was performed as well as the name of the biosensor. A comment field is also provided that explains why the compound was included in the dataset, based on quotes from the reference publication or the database it was extracted from. Manual curation of ACS Synthetic Biology abstracts (Volumes 1 to 6 and Volume 7 issue 1) was performed as well as extraction from the following databases: Bionemo v6.0 [301], RegTransbase r20120406 [302], RegulonDB v9.0 [303], RegPrecise v4.0 [304] and Sigmol v20180122 [305].

Specifications Table

Subject area	Biology
More specific subject area	Synthetic biology
Type of data	Table
How data was acquired	Database extraction from Bionemo v6.0 [301], RegTransbase r20120406 [302], RegulonDB v9.0 [303], RegPrecise v4.0 [304] and Sigmol v20180122 [305] as well as manual curation ACS synthetic biology abstracts (Volumes 1 to 6 and Volume 7 issue 1)
Data format	Analyzed
Experimental factors	Not applicable
Experimental features	Not applicable
Data source location	Github
Data accessibility	Data is with this article and on Github

Table 4.1 Detectable compounds dataset specifications

Value of the data

- This dataset provides a basis for the development of new biosensing circuits for synthetic biology and metabolic engineering applications, e.g. the design of whole-cell biosensor, high-throughput screening experiments, dynamic regulation of metabolic pathways, transcription factor engineering or creation of sensing-enabling pathways
- This dataset provides a unique source of a broad number of compounds that can be detected and acted upon by a cell, increasing the possibility of orthogonal circuit design from the few usual compounds used in those applications
- The manually curated section provides information on where the biosensor has been first reported and successfully used, enabling the reader to select trustworthy information for his application of choice
- Detectable compounds can be searched by both by name and chemical similarity
- This dataset is an update of 10.6084/m9.figshare.3144715.v1 [77]

4.2 Data

The aim of this dataset is to identify and collect compounds that are known for being detectable by a living cell, through the action of a genetically encoded biosensor and is centered on bacterial transcription factors. The dataset should allow the synthetic biology community to consider a wide range of applications. The reader will find in this dataset the name of the compounds, their InChI (molecular structure), the publication where the detection was reported, the organism in which this was detected or engineered, the type of detection and experiment that was performed as well as the name of the biosensor. A comment field is also provided that explains why the compound was included in the dataset, based on quotes from the reference publication or the database it was extracted from. Manual curation of ACS synthetic biology abstracts (Volumes 1 to 6 and Volume 7 issue 1) was performed as well as extraction from the following databases: Bionemo v6.0 [301], RegTransbase r20120406 [302], RegulonDB v9.0 [303], RegPrecise v4.0 [304] and Sigmol v20180122 [305].

This dataset is available online on GitHub to allow for further updates as well as community contributions.

4.3 Experimental design, materials and methods

4.3.1 Manual curation of ACS synthetic biology (Volume 1 to 6 and Volume 7 issue 1)

All abstracts of ACS Synthetic biology (Volume 1 to 6 and Volume 7 issue 1) were read and information relevant to this dataset was extracted from those abstracts. The aim of this manual curation was to establish a list of detectable compounds whose detection method was already successfully implemented in a synthetic circuit, providing a good basis for further implementation for synthetic biologists.

4.3.2 Bionemo v6.0 [301]

The Structured Query Language (SQL) request used to create this dataset is:

```
SELECT DISTINCT substrate.id\_substrate, minesota\_code,
 name FROM substrate
    INNER JOIN complex\_substrate ON complex\_substrate.id\_substrate =
   substrate.id\_substrate
    INNER JOIN complex ON complex.id\_complex =
      complex\_substrate.id\_complex
    WHERE activity='REG';
```

4.3.3 RegTransbase r20120406 [302]

The SQL request used to create this dataset is:

```
SELECT DISTINCT a.pmid, e.name, r.name
    FROM regulator2effectors AS re
    INNER JOIN exp2effectors AS ee ON ee.effector_guid=re.effector_guid
    INNER JOIN dict_effectors AS e ON e.effector_guid=ee.effector_guid
    INNER JOIN regulators AS r ON r.regulator_guid=re.regulator_guid
    INNER JOIN articles AS a ON a.art_guid=ee.art_guid
    ORDER BY e.name;
```

RegTransbase was not maintained anymore at the time of writing of this manuscript.

4.3.4 RegulonDB v9.0 [303]

The SQL request used to create this dataset is:

```
SELECT c.conformation_id, c.final_state, e.effector_id, e.effector_name,
               tf.transcription_factor_id, tf.transcription_factor_name,
                                    p.reference_id, xdb.external_db_name
    FROM effector AS e
    INNER JOIN conformation_effector_link AS mm_ce ON
                                         mm_ce.effector_id=e.effector_id
    LEFT JOIN conformation AS c ON
                                 c.conformation_id=mm_ce.conformation_id
    LEFT JOIN transcription_factor AS tf ON
                   tf.transcription_factor_id=c.transcription_factor_id
    LEFT JOIN object_ev_method_pub_link AS x ON
                                       x.object_id=c.conformation_id  OR
                               x.object_id=tf.transcription_factor_id OR
```

```
                                                x.object_id=e.effector_id
LEFT JOIN publication AS p ON p.publication_id=x.publication_id
LEFT JOIN external_db AS xdb ON xdb.external_db_id=p.external_db_id
WHERE c.interaction_type IS Null OR c.interaction_type!='Covalent';
```

4.3.5 RegPrecise v4.0 [304]

The RegPrecise website was accessed (version v4.0) and all relevant data was extracted from the effector pages of the website.

4.3.6 Sigmol v20180122 [305]

Sigmol was accessed on 16/02/2017 and all effector data was retrieved from the unique Quorum Sensing Signaling Molecule page. In the "detected by" column, we provide the class of signaling compounds the compound belongs to. The comment field reads 'Extracted from Sigmol v20170216 – Uniq_QSSM_"number"'.

4.3.7 Data overview

In Table 4.2 are presented some characteristics of each data source: number of compounds without a structure from this source, total number of compounds with a structure from this source and number of compounds with a structure found only in this source. The last column in particular shows that around half the compounds are found in more than one data source.

Source	Compounds without structure	Compounds with structure	Unique compounds with structure
RegPrecise	136	418	73
BioNemo	5	499	8
RegTransBase	683	2057	63
RegulonDB	12	245	23
Sigmol	2	175	135
ACS synthetic biology	44	287	73
All sources	**882**	**3681**	**729**

Table 4.2 **Contribution of each data source** The first column contains the data source, the second column the number of compounds found without a structure in that source, the third column the number of compounds with a structure (InChI) and the last column the number of compounds with a structure found only in that source.

Figure 4.1 shows the repartition of the type of experiment (*in vivo*, unspecified or other), as well as the repartition of Biosensor type (Transcription factor, riboswitch or unspecified) in the full dataset and the manually curated dataset from ACS synthetic biology.

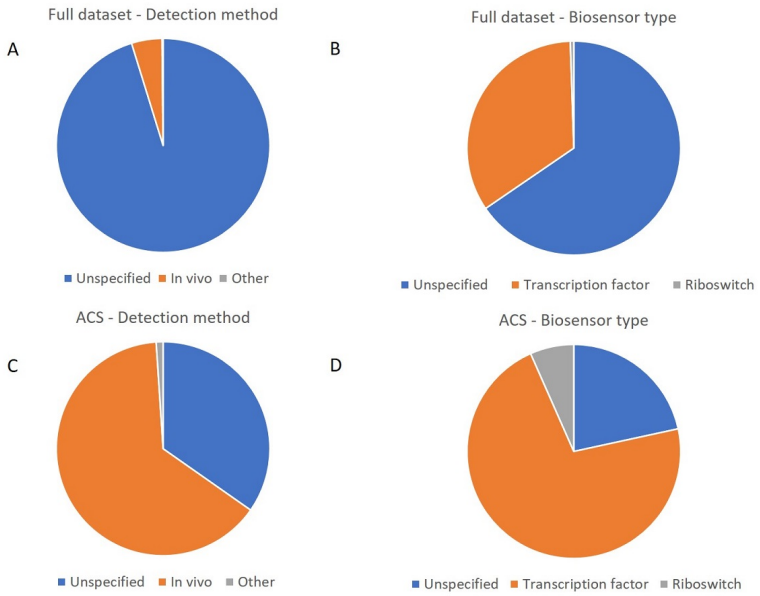


Figure 4.1 **Type of experiment and biosensor type in the full dataset and the manually curated dataset.** **A**: Full dataset – detection method. **B**: Full dataset – biosensor type. **C**: ACS dataset – detection method. **D**: ACS dataset – biosensor type. **A** and **C**: other in detection method corresponds to *in silico*, *in vivo* and cell-free detections. **B** and **D**: ACS dataset is the dataset obtained from manual curation of ACS synthetic biology with compounds that have available structures.

5

Large scale active-learning-guided exploration to maximize cell-free production

This work was submitted for publication by Olivier Borkowski *, Mathilde Koch *, Agnès Zettor, Amir Pandi, Angelo Cardoso Batista, Paul Soudier and Jean-Loup Faulon. It is available on BioArchive.
Only minor modifications to the submitted paper have been introduced in the Chapter below.
* stands for equal contributions.

Detailed contribution to this thesis

Within this Chapter, I tackled another aspect of circuit design and combinatorial space exploration. The question was the following: given a protein of interest (but this could be a metabolite, as long as it is detectable in a high throughput manner), how do we select components of the cell-free mix to maximize this production? While the combinatorial space is much too large to be explored exhaustively, active learning algorithms, which suggest the next round of experiments to optimize a metric of interest are perfectly adapted to such a problem. Therefore, I developed an active learning method, coupled with liquid handling robots for experiments, to optimize protein production in cell-free extracts. The conceptual question is similar to the problem of retrosynthesis when both are considered as algorithms for combinatorial space exploration, as presented in the introduction. Moreover, from the practical standpoint, improving lysate quality leads to better experimental results when testing the designed circuits in cell-free systems.

Full reference

Borkowski O.*, Koch M.*, Zettor A., Pandi A., Cardoso Batista A., Soudier P. and Faulon J-L. (2019) Large scale active-learning-guided exploration to maximize cell-free production *BioArchive* 10.1101/751669
* stands for equal contributions.

Contributions as stated in the article

O.B, M.K and J-L F designed experiments. O.B performed experiments. M.K developed and performed model simulations and liquid handler programming. O.B and M.K performed data analysis. O.B and A.Z collected data. O.B, A.P, A.C.B and P.S provided lysates. A.P cloned and maxi prep the plasmid. O.B, M.K, A.Z and J-L F wrote the paper. All authors approved the manuscript.

5.1 Abstract

Lysate-based cell-free systems have become a major platform to study gene expression but batch-to-batch variation makes protein production difficult to predict. Here we describe an active learning approach to explore a combinatorial space of $\simeq$ 4,000,000 cell-free compositions, maximizing protein production and identifying critical parameters involved in cell-free productivity. We also provide a one-step-method to achieve high quality predictions for protein production using minimal experimental effort regardless of the lysate quality.

5.2 Results and Discussion

Cell-free systems, especially lysate-based systems, are major platforms for both prototyping of genetic circuits and understanding of fundamental processes [306, 74, 307, 308, 67, 309, 310]. They provide fast gene expression kinetics, low reaction volumes, allowing high-throughput measurements and simplified gene characterization via decoupling protein production from host physiology [44, 311, 174, 68, 312]. Cell-free systems could disseminate among laboratories and be standard methods for molecular biology if efficient and predictable protein productions were guaranteed. Ribosomes, native poly-

merases and cofactors concentrations remain arduous to control as they are provided by the lysate [64, 313], making the efficiency of cell-free systems variable. A great challenge is to develop a lysate-specific optimization method for cell-free composition to maximize protein production. Using a Design of Experiment approach, Caschera *et al* [64] explored cell-free compositions by varying one compound concentration at a time and obtained a 10 fold increase of protein production in a lysate-based cell-free system. Such results reveal the considerable margins of improvement of protein expression in such systems.

Here, we use an active learning approach [95, 314] to explore, optimize and understand the impact of cell-free composition on protein production in cell-free systems. We demonstrate that sufficient amount of data can be obtained to train a machine learning algorithm, achieve high quality predictions and increase protein production by 34 times. We next show that only 20 informative compositions are enough to train our machine learning model and obtain accurate predictions. This approach enables to maximize protein production on different cell lysates with minimal experimental effort.

To study cell-free systems productivity, we developed an automatable strategy coupling an acoustic liquid handling robot (Echo 550, Labcyte, USA) and a plate reader (Infinite MF500, Tecan, USA) to measure $\simeq$4000 cell-free reactions (including controls and triplicates) and provide data to train a machine learning model. The lysate was obtained by sonication and supplemented with compounds described in Figure 5.1a. The reference concentrations is based on the protocol developed by the Noireaux laboratory [68] (see Methods, Supplementary Figure 5.3). We fixed 4 concentration levels for each of the 11 compounds leading to a combinatorial space of 4,194,304 possible compositions (Figure 5.1a). Protein production was measured using the fluorescence level from the expression of sfgfp under control of a constitutive promoter (Figure 5.1b, Supplementary Table 5.1). In order to compare measurements between plates, we maximized a relative fluorescence level named yield hereafter (Figure 5.1b). The yield is defined as the ratio of the fluorescence produced with a chosen composition divided by the fluorescence obtained with the reference composition (Figure 5.1b). To explore our vast combinatorial space, we used an active learning strategy [95], combining both exploration and exploitation to increase the yield and reduce model uncertainty (Figure 5.1c). Each iteration started with 102 new cell-free compositions to be tested. The fluorescence level was measured in a plate reader and fed to an ensemble of neural networks (Figure 5.1c, see Methods). Our active learning loop was initiated with a training set of 102 cell-free compositions (see methods: 22 chosen and 80 random compositions, Figure 5.1c). The first iteration already led to a maximum of 10 fold improvement of the yield (Figure 5.1d). As expected, the prediction

accuracy was very low (Figure 5.1e). After 7 iterations, we reached a maximum for both the yield (Figure 5.1d) and the prediction accuracy (Figure 5.1e). Eventually, we stopped at 10 iterations as we were not able to increase neither the yield nor the prediction accuracy of our model (Figure 5.1d, Figure 5.1e, see methods). Throughout our workflow, we measured fluorescence levels in 1017 cell-free compositions and validated the efficiency of our method with a high quality predictions score ($R^2 = 0.93$) and a maximum of 34 fold increase of the yield. The 1017 cell-free compositions were sorted, from low to high yields, to observe the relationship between yield and composition (5.1f). An increase of Mg-glutamate, K-glutamate, Amino Acids and NTPs concentrations and a decrease of cAMP, spermidine and 3-PGA concentrations can be noticed with increasing yield (5.1f). We used a mutual information analysis (see Methods) to reveal the dependence between our 11 compounds concentration and the yield. Mg-glutamate, K-glutamate, Amino Acids, cAMP, spermidine, 3-PGA and NTPs exhibit a score between 0.25 and 0.75, confirming that a variation of their concentrations strongly impacts protein production (5.1g). Variation of tRNA, CoA, NAD and Folinic Acid concentrations have little impact on the yield (5.1g).

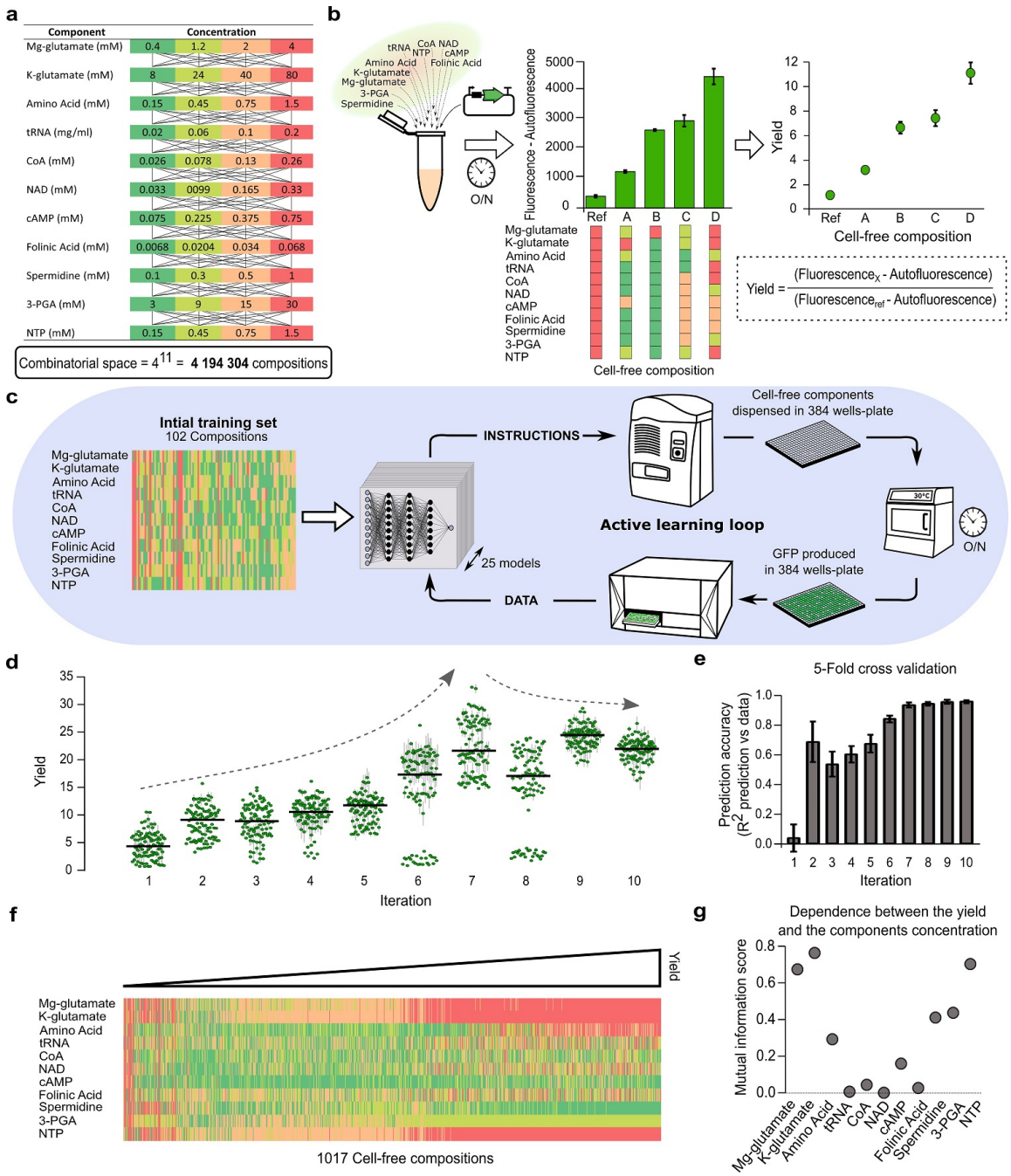


Figure 5.1 **Active learning loop to explore the composition of a cell-free system** **a** List of chemicals added to the cell-free mix in addition to PEG-8000, HEPES and the lysate. Four concentrations have been chosen for each chemical. The concentration in red is the highest concentration, then orange, light green, dark green stand for 50%, 30% and 10% of the highest concentration. **b** An example of fluorescence obtained using 4 cell-free compositions with our plasmid (10 nM). The autofluorescence value is measured with the reference composition without DNA and subtracted from every measurement in the plate. The yield is the ratio between the fluorescence of a composition x and the reference composition. **c** Illustration of the active learning approach used to explore the combinatorial space of cell-free composition and trained an ensemble of 25 machine learning models. **d** Yield evolution among 10 iterations. The green dots are the mean yields of 3 replicates obtained in the same plate with the same composition. The vertical grey lines stand for the standard deviation of the 3 replicates. The horizontal black line is the median value of all the yields obtained during an iteration. The Arrows represent the evolution of the maximal yields value. **e** Quantification of the predictive accuracy of the model using a 5-fold cross validation. **f** Cell-free compositions tested in the study sorted by yield level. A row stand for one mix composition, the color code is the same as in panel **a**. **g** Results of a mutual information analysis, using the 1017 compositions, of the relationship between the yield and each chemical compound.

Next, we investigated whether protein production in cell-free using lysates made in other conditions (different experimentalists, using a different strain or supplemented with antibiotics) could be quickly predicted with a one-step method (Figure 5.2a). We selected 102 cell-free compositions representative of the 1017 already tested with the original lysate (see methods, Supplementary Figure 5.4a). Among the 102 compositions, 20 were used to train the model and 82 to test its predictive accuracy (Figure 5.2a). The challenge lies in the model's ability to accurately predict a large diversity of yields based on a small training dataset. The 20 compositions (magenta dots in Figure 5.2) were chosen to be highly informative (see methods, Supplementary Figure 5.4b). We used the same 20 and 82 compositions to train and test our model with all the lysates used in Figure 5.2. With new lysates prepared by other experimentalists (labeled $lysate_{PS}$ and $lysate_{AB}$), similar cell-free compositions led to different yields but the compounds exhibiting a high impact on protein production remained the same (Figure 5.1g and Supplementary Figure 5.5). The maximum yield among the 102 tested compositions differs from one lysate to another, with a maximum yield at 23 and 26 for the $lysate_{PS}$ and $lysate_{AB}$ respectively (Figure 5.2a,b). The 102 yields obtained with the original lysate, labeled $lysate_{ORI}$, are presented in Supplementary Figure 5.6a. The yield used previously is a relative measurement (Figure 5.2 and Supplementary Figure 5.7) which does not allow absolute comparison between our cell-free systems. We calculated a global yield (calculated with the $lysate_{ORI}$ as a global reference, Supplementary Figure 5.6b) and observed a maximum global yield 1.5 times higher with $lysate_{PS}$ than $lysate_{AB}$ (Supplementary Figure 5.6c). These results highlight the variability in lysates quality even when they are prepared in the same laboratory with the same strain and protocol. Despite these differences, we achieved high quality predictions with both lysates (Figure 5.2 a,b). We obtained a $R^2 \simeq 0.9$ for both lysates and linear fits with intercepts of 0.2 / 0.1 and slopes of 0.8 / 1.01 with $lysate_{PS}$, $lysate_{AB}$ respectively (Figure 5.2 a,b). These results validate our approach to both maximize protein production and accurately predict protein production regardless of the experimentalists who prepared the lysate.

We then challenged our method by interfering with the transcription or translation processes to mimic lysates of lower quality. By adding rifaximin (Figure 5.2c) or spectinomycin (Figure 5.2d) to the cell-free mix, we interfered with the transcription or translation apparatuses respectively. The two antibiotics led to a strong decrease in absolute protein production (Supplementary Figure 5.6c) but opposite behaviors can be observed (high versus low room for yield improvement, Figure 5.2 c,d, Supplementary Figure 5.7c,d). When the transcription process is impaired, we obtained a predic-

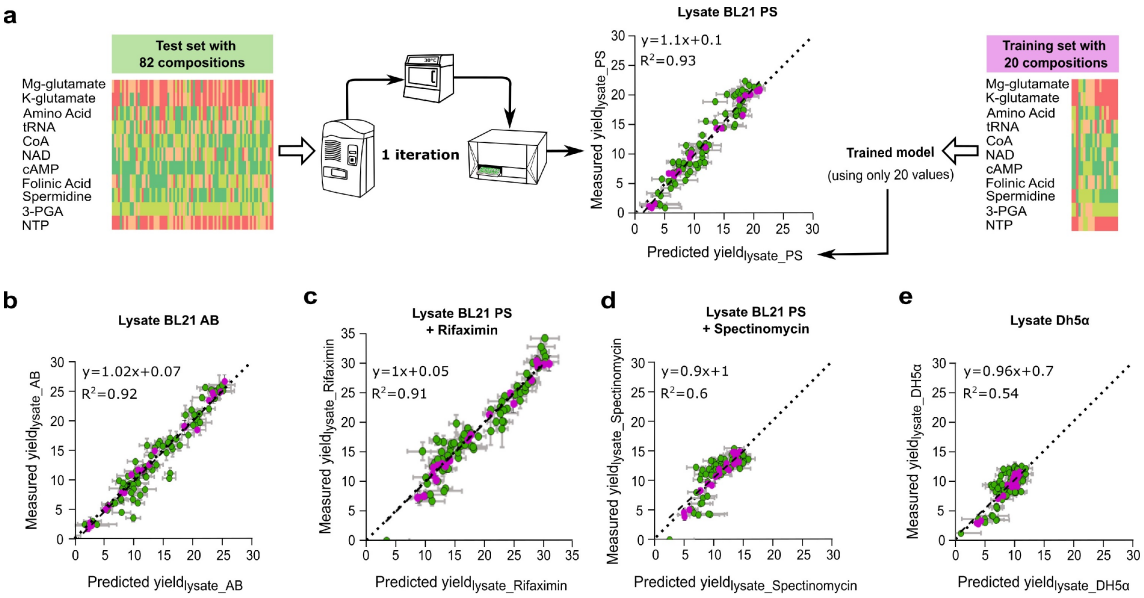


Figure 5.2 **One-step method to predict protein yield in cell-free systems. a** Illustration of the method used to predict the yield of protein expression with a new lysate, labeled PS, made by another experimentalist. The training of the model is based on yield measurements of 20 compositions (Magenta circles). The choice of the 20 combinations leading to the best predictions is described in the methods. The yield obtained with 82 compositions (Green circles) were measured and compared to the model predictions to test its accuracy (R^2 value). The yield is specific to each lysate as the reference composition used the same chemicals concentration as in Figure 5.1 but with different lysate. **b** Comparison of the yields obtained with the lysate AB (made by a third experimentalist) vs the model predictions. **c** Comparison of the yields obtained with the lysate, of panel **a** supplemented with $0.25\,\mathrm{mg}\cdot\mathrm{mL}^{-1}$ of rifaximin vs the model predictions. **d** Comparison of the yields obtained with the lysate, of panel **a**, supplemented with $0.5\,\mathrm{mg}\cdot\mathrm{mL}^{-1}\mathrm{L}$ of spectinomycin vs the model predictions. **e** Comparison of the yields obtained with a lysate obtained from the stain $DH5\alpha$ vs the model predictions. In all panels, the model predictions are based on a model trained with the same 20 compositions and the same test set of 82 compositions only lysate differs. In all panels, the horizontal grey lines stand for the standard deviation of 3 replicates. The vertical grey lines stand for the standard deviation of 25 predictions.

tion of high accuracy with a R^2 of 0.91 and linear fit intercepts of 0.2 and slopes of 0.9 (Figure 5.2c). The cell-free containing rifaximin exhibits a high leeway for yield improvement (Figure 5.2c and Supplementary Figure 5.7c) with a maximum yield of 35 among the 102 cell-free compositions. When the translation process is impaired, the yield is capped to a maximum improvement of 15 (Figure 5.2d, Supplementary Figure 5.7d). The R^2 value observed in Figure 5.2d is lower but the linear fit exhibit an intercept of 0.1 and slopes of 0.9. Thus, we obtained accurate prediction for the low and high yields value but the intermediate yields remains difficult to estimate. Such predictions stay powerful to maximize protein production as extreme values are captured and provide precious information concerning the lysate quality (Supplementary Figure 5.8, Supplementary Note 1).

Eventually, we tested our method with a lysate prepared using the strain $DH5\alpha$. As observed with the lysate supplemented with spectinomycin, the R^2 value is low but the linear fit of the data exhibits an intercept of 0.07 and slopes of 0.96. The maximum global yield obtained with this lysate was low, as expected for a strain not optimized for protein production [315] (Supplementary Figure 5.6c). Nevertheless, with half of the tested cell-free compositions, the $Lysate_{DH5\alpha}$-based cell-free exhibits a high global yield (Supplementary Figure 5.6c). The yield exhibits a similar behavior as the lysate supplemented with spectinomycin, suggesting an impaired translation process (little room for yield improvement, Figure 5.2d,e, Supplementary Figure 5.7d,e), but with a higher level of protein production.
Our method enables a fast lysate-specific optimization of the cell-free composition to predict and maximize protein production (Figure 5.2, Supplementary Figure 5.9, Supplementary Note 2). Our results suggest that the optimization of the cell-free composition mainly improves the efficiency of the translation apparatus as we observed a limited improvement with an impaired translation. On the contrary, a damaged transcription machinery can be balanced by the optimization of the cell-free composition. Our approach gives precious information about the room for protein production improvement of a home-made cell-free system, the impact of each compound on cell-free productivity and the efficiency of the transcription and translation processes. Our method, based on the measurements of green fluorescent protein (GFP) production with the same 20 cell-free compositions used in this work to train the model provided, can be easily extended to any other bacterial-based cell-free [308, 316, 317] to investigate cell-free optimization beyond *E. coli* cell-free systems. As our model is not based on mechanistic hypotheses, our method can be extended to cell-free systems using other organisms as yeast, insect, plant or human cells after performing new explorations to find the 20 most informative compositions for those cell-systems.

5.3 Methods

5.3.1 Bacterial strains and DNA constructs

Strains BL21 DE3 and $DH5\alpha$ were used to prepare the different lysates in this study. Our sfgfp plasmid was obtained by modification of the RBS of the plasmid pBEAST-J23101-sfGFP9. We used PCR amplifications using the reverse primer GCGGTCTCACATCTACTATTTCTCCTCTTTCTCTACTAGCTAGC and forward primer GCGGTCTCAGCTTACTTTATCTGAGAATAGTC with the backbone, and reverse primer p CCGGTCTCAAAGCTTATCATCATTTG-

TACAGTTCATCC and GCGGTCTCAGATGCGTAAAGGCGAAGAG forward primer with the sfgp sequence. The PCR amplifications was followed by a golden gate assembly using BsaI and T4 ligase (New England Biolab) and transformed into chemically competent *E. coli* top10.

5.3.2 Plasmid preparation

We noticed with preliminary experiments that the same cell-compositions gave different results when we used plasmid DNA from miniprep done on different days using the same kit. The whole project was done using aliquots from the same initial batch of *sfgfp* plasmid. The plasmid was extracted from a 600 ml LB of *E. coli* top 10 using the Plasmid DNA purification NucleoBond Xtra Maxi of Macherey-Nagel. The 500 µL aliquots were stored at -80 °C. The whole project was done using aliquots from the same initial batch of DNA. The final *sfgfp* plasmid concentration in every reaction was 10 nM.

5.3.3 Cell-free reagents preparation

As the reagents preparation can have a significant impact on cell-free efficiency [318], all our reagents except spermidine and Mg-glutamate (we run out of those two compounds during the study) came from aliquots of the same initial batch. We did not see an impact on our control when the spermidine and Mg-glutamate were renewed.

5.3.4 Cell lysate mix preparation and reactions

The cell lysate preparation is based on the protocol of Sun *et al.* [68]. Briefly, the protocol of Sun *et al.* [68] is a 5 days protocol in three phases: harvest cells (colonies grow on plate overnight at 37 °C, 50 mL preculture at 37 °C for 8h, 12 liters of cultures at 37 °C until OD_{600}= 1.5), lysate preparation (multiple pellet washing followed by beads-beating to obtain an lysate). The protocol was modified by using sonication instead of use of a bead beater to obtain BL21 or DH5α cell lysates. After washing the cells as following the sun *et al.* protocol (Day 3 step 18) with S30A buffer (14 mM Mg-glutamate, 60 mM K-glutamate, 50 mM Tris, 2 mM DTT, pH 7.7), the cells were centrifuged 2000g for 8 min at 4 °C.

The pellet was re-suspended in S30A (pellet mass (g) x 0.9 mL). The solution was split in 1.5 mL aliquots in 2 mL Eppendorf tubes. Eppendorf tubes were placed in a cold block and sonicated using Vibracell 72408 (from Bioblock scientific) using the following procedure: 20s ON - 1 min OFF - 20 s ON - 1 min OFF - 20 s ON. Output frequency 20 kHz, amplitude 20%. The remaining protocol followed the procedure of the Sun *et al.* protocol for day 3, step 37. mRNA and protein synthesis are performed by the molecular machinery present in the lysate, with no addition of external enzymes. Reactions take place in 10.5 µL volumes at 30 °C in

384-well plate. Note that we kept the 50 mM HEPES and 2% PEG-8000 fixed in every reaction.

$Lysate_{ORI}$, $Lysate_{PS}$ and $Lysate_{AB}$ were obtained from the same *E. coli* strain BL21 in the same laboratory with the same sonicator and centrifuge. The $Lysate_{ORI}$ came from one-batch prepared from 12 Liters of BL21 culture. The 12 liters culture were separated in 4 Liters culture. The culture were inoculated, grown and their pellets were washed on different 3 days then freeze and stock at -80 °C. Then, the pellets were weighed, re-suspended in S30 buffer, pooled, sonicated, centrifuged, mixed and aliquoted on an extra day. The Lysates $Lysate_{PS}$, $Lysate_{AB}$ and $Lysate_{PS}$ and $Lysate_{DH5\alpha}$ were each obtained from 2 liters culture. For the $Lysate_{spectinomycin}$ and $Lysate_{rifaximin}$, the final concentration of rifaximin and spectinomycin were $0.25\,\mathrm{mg}\cdot\mathrm{mL}^{-1}$ and $0.5\,\mathrm{mg}\cdot\mathrm{mL}^{-1}$ respectively. They were added to the cell-free reactions using $Lysate_{PS}$.

5.3.5 sfGFP purification

The *sfGFP* was produced in *E. coli* culture. After a 10 min centrifuge at 4000g, the pellet was re-suspended in 20 mM Tris (Ph8), 0.2 M NaCl and sonicated (Output frequency 20 kHz, amplitude 40 % with the Vibracell 72408). After sonication, the solution was centrifuged (400g, 15 min). The proteins in the supernatant were purified and fractionated using ammonium sulfate. The *sfGFP* was isolated at more than 70% saturation. The solution was centrifuged (4000g, 15 min) and the pellet re-suspended in 20 mM Tris (Ph8), 100 mM NaCl. The solution was dialysed overnight in 20 mM Tris (Ph8), 100 mM NaCl. Eventually, for the last step of purification, we used a Mono Q anion exchange chromatography column (GE Healthcare) and obtained a solution of 90 % *sfGFP*. The final solution dialyse in a solution 0 mM Tris (Ph8), 100 mM and 50% glycerol leading to a final concentration of $7.62\,\mathrm{mg}\cdot\mathrm{mL}^{-1}$. To obtain an absolute quantification of the protein production in cell-free, we measured the *sfGFP* fluorescence in wells containing 10.5 µL of *sfGFP* solution at different concentration. The G-yield values are calculated as described in Supplementary Figure 5.6b with the fluorescence measured from *sfGFP* and no autofluorescence divided by the cell-free mix $lysate_{ORI}$ autofluorescence and the reference fluorescence obtained from our plasmid in a cell-free mix with $lysate_{ORI}$.

5.3.6 myTXTL commercial kit

We used the commercial kit: myTXTL from Arbor Biosciences (Sigma 70 Master Mix Kit, (USA). We used both our plasmid (10 mM final concentration) and the control plasmid, pTXTL-P70a(2)-deGFP (20 nM final concentration) provided by Arbor Biosciences. The 2 plasmids were expressed with the reactions provided with myTXTL kit and with the optimized cell-free reaction with the $lysate_{ORI}$ Supplementary Figure 5.9b.

5.3.7 Fluorescence quantification

We used a plate reader Infinite MF500 (Tecan) to measure fluorescence in 384-well plates (Nunc 384-well optical bottom plates, Thermo-Scientific). The excitation wavelength was fixed at 425 nm, the emission at 510 nm and the gain at 50. We measured 5 fluorescence values for each well as a quality control of the plate reader measurements. The fluorescence was measured from the top of the 384-well plates with no lid.

5.3.8 Echo liquid handler

We used the Echo software Cherry Pick to program the Echo 550 liquid handler. The software was programmed using CSV (comma separated values) files that gave machine-readable instructions: namely the well it had to take liquid from (containing pure reagents), the well the liquid was destined to and the volume that was to be taken. It allows us to program the content of each individual well separately. We calculated the concentrations of our chemical compounds stocks so the final volumes sent to the destination well were multiples of 2.5 nL (the droplet size managed by the Echo machine). The scripts generating the CSV file are presented below in concentrations to instructions workflow. We chose our stock volumes so that the minimal volume to transfer was 12.5 (=5 droplets). The chemical compounds were dispensed using BP2 fluid class except for K-glutamate and 3PGA (CP fluid class).

5.3.9 General script descriptions

All scripts mentioned below were written in Python (version 3.6.5), executed in Jupyter notebooks (version 1.0.0). Scripts are available online on github. The libraries numpy and csv were used to handle files between different scripts. We used scitkit-learn [279] (version 0.19.1) for all model training.

5.3.10 Concentrations to instructions workflow

The details of those scripts are described in the READme file of the ECHO-handling-scripts of our code. Roughly, it proceeds in 4 steps:

1. **Complete concentrations:** Taking as input a file containing only concentrations of interest for the machine learning algorithm, it adds information of values that are constant across all conditions, such as the lysate quantity.
2. **Concentration to volume:** This file converts a csv file concentrations -to a file of volumes one wants to test (in triplicates). This is due to the fact that the ECHO liquid handler needs volumes as inputs.

3. **Optional:** we sorted those volume files according to water content. This allows us later to manually pipet important water volumes so that the robot only adjusts small volumes.
4. **Volume to echo:** This file converts a set of transfer volume quantities to the csv file expected by the ECHO liquid handler (instructions files). It also provides a file containing the name of the wells with their corresponding transfer volumes. This file is used to match the well compositions with the fluorescence measurements obtained later with the plate reader. The amino acids and water were pipetted manually (for volumes $> 1\,\mu L$).
5. **Named volumes to concentrations:** maps the volumes and the associated well name to a concentration file with the associated well name, for integration with the fluorescent plate reader at the next step.

The script matched the named concentration with the yield value as described in see Data analysis of those methods.

5.3.11 Data analysis

We provide a script to map the fluorescence quantification (see fluorescence quantification above) to the tested concentrations with well names (last step of concentrations to instructions workflow above). We performed outlier removal based on the following criteria: if the coefficient of variation, among 3 replicates, was higher than 30%, we removed the value farthest from the other 2. This concerned 27 values of our 1017 values tested during the active learning. Those are identified in the online data on Github with the third value of fluorescence is set to -1. This script also outputs csv files allowing for visualization of where the outliers are, in order to spot potential border effects. It also separately outputs the outliers for further analysis.

5.3.12 Data normalization

We normalized using the following equation:

$$Yield_{composition} == \frac{Fluorescence_{composition} - autofluorescence}{Fluorescence_{reference} - autofluorescence}$$

Where $autofluorescence$ is the fluorescence measured in the cell-free reaction supplemented with water and using the reference composition. The yield exhibited in Figure 5.2 used a cell-free reaction with the new lysate to measure the autofluorescence and the fluorescence with the reference composition. In supplementary Figure 5.6, all the yields are calculated with $autofluorescence$ and reference fluorescence of the $Lysate_{ORI}$.

5.3.13 Quality controls

In every 384-well plates we measured 13 control compositions (in triplicate) including the reference composition with and without DNA. In each 384-well plate, we used 2 rows of controls: A and P. The controls in row A never changes. The controls in row P changed throughout the workflow. We used the compositions leading to higher yields in the previous iteration. When analyzing our controls, we checked whether the yields were identical from plate to plate ($R^2 > 0.75$ between new plate and all previous plates on yield of controls). Plates with R^2 >0.75 when compared to all previous plates, or systematically above or below other plates are discarded and the same combinations were tested again.

5.3.14 Initialization of the Machine learning

For the first plate of the active learning, we proceeded as follows. We chose 22 concentrations that we wished to test: fixing all reagents at the maximum allowed concentration, except one which was at the lowest (11 combinations) and fixing all reagents at the minimum allowed concentration except one which was at the highest (11 combinations). The rest (80 compositions) was filled randomly.

5.3.15 Model training

The models were trained as follows.

1. **Input data is normalised:** each component maximum concentration is 1, and the other values take discrete values of 0.1, 0.3, or 0.5 as described in the legend of Figure 5.1. While unnormalized inputs could be used, we strongly encourage normalization due to scale differences between the inputs.
2. **We train an ensemble of n models** where n = 25. For each model, we train it 10 times ($models_number$) using the whole dataset at the moment (e.g. 3x102 values at the 3rd iteration). Training the model multiple times allows for optimizing for random weight initialization of the model. We keep the best model (with the highest regression from scitkit-learn R^2 score).
3. Multilayer perceptrons give the best results (random forests and linear regressions were also investigated early on). They are trained with the default parameters from scitkit-learn except the following parameters: maximum iteration of 20000, adaptive learning rate, adam solver, early stopping and the following layers: (10, 100,100, 20)
4. We obtain mean and standard error from our predictions by taking the mean and standard error from the n results generated by our ensemble of n models.

5.3.16 Active learning

The workflow used the data from all the available plates as an input. It trains an ensemble of 25 models and returns instructions for the following round. Here is the detailed process:

1. For N times (N= 100,000):
    a) Randomly sample a composition in the composition space (Figure 5.1a)
    b) If a composition was drawn previously (either in a previous experiment or during current selection), neglect it.
2. Predict mean and standard deviation for all 100,000 points using the ensemble of 25 models previously trained.
3. Select the best set of compositions, according to the following Upper Confidence Bound (UCB) formula: $exploitation * yield_{pred^{mean}} + exploration * yield_{pred^{std}}$, with exploitation = 1 and exploration = 1.41. Our scripts output the best 500 compositions based on the mean and std predictions of the yields. A high std value stands for an uncertain yield value. We output compositions for full exploitation, full exploration and maximization of the above formula but use the third option for the rest of the workflow. We are therefore querying points with both high yield and uncertainty.

5.3.17 Model statistics

For model statistics presented in Figure 5.1e, we used the same models as described in the active learning section above, but using 5 fold cross validation instead of the whole training set. The full dataset is separated into 5 subsets then the 25 models are trained on 4 subsets, and used to predict the 5th, where scores are obtained. This is done 5 times, once on each subset. The scores presented in Figure 5.1e are the mean and standard deviation of those 5 scores.

5.3.18 Mutual information calculation

Mutual information is a method to quantify the mutual dependence between two variables. This concept is intrinsically linked to the concept of entropy and is especially useful to quantify non-linear relationships between variables. More information on the theory behind this method can be found in the review estimation of mutual information [319] and in sci-kit learn documentation [279]. It was calculated using the feature-selection.mutual-info-regression function from scitkit-learn [279] (version 0.19.1) between each feature and the yield ($compounds.effect.analysis/mutual.information.analysis$ Jupyter notebook) with default parameters.

5.3.19 Identification of informative points

To identify the most informative points, we proceeded in the following manner: We did 1000 iterations of the following procedure:

1. Randomly sample n combinations from the dataset (n=20 out of a dataset of 102 values for Figure 5.2)
    a) Train models on those points using the strategy presented in model training for each lysate
    b) Predict on the other points (82 for Figure 5.2) for each lysate
    c) Obtain the average score on all lysates
2. Keep those combinations if this average is better
3. Note: Data is saved every 100 iterations

5.3.20 Maximization of the protein production for future users

Users must do the following experiments:

1. Maxiprep a LB culture of our plasmid (or MyTXTL plasmid)
2. Measure the yields (or absolute fluorescence) in the 20 cell-free compositions described in Figure 5.2a

Then, in order to apply our method to a new extract, a Jupyter notebook called $predict_for_new_lysate$ is available. It takes as input a csv file containing the 20 tested concentrations and the 20 corresponding yields and standard error values. It provides as an output a file to maximize exploration, exploitation or a combination of both as in the active learning loop. For obtaining the highest possible yield, it is recommended to take the exploitation results, which contain the highest predicted yields. It must be noticed that several cell-free compositions can be predicted to reach maximum yield or values in the same range. The algorithm provides mean yields value with standard deviation errors and so several yields will be equivalent to the maximum value. During this study we provided yield values to our training algorithms but absolute fluorescence can also be used if a user does not need to compare fluorescence values measured on different 384-well plates.

5.4 Supplementary Data

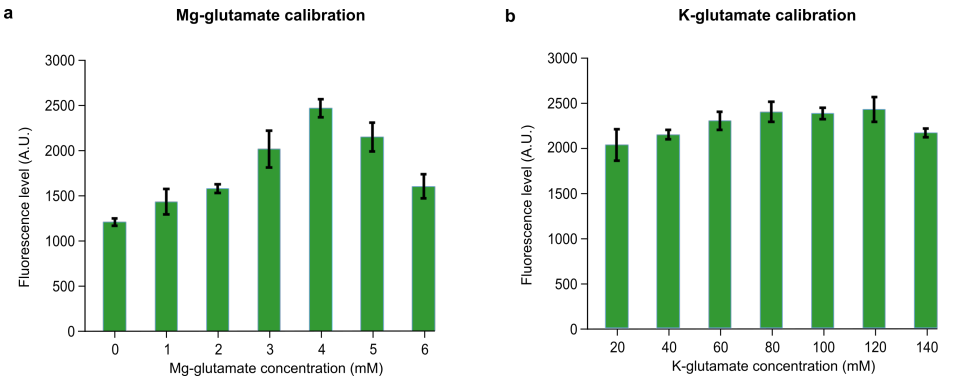


Figure 5.3 **Preliminary calibration of the cell-free composition** The lysate is usually only calibrated for Mg-glutamate, K-glutamate levels. Here we show the end point after overnight cell-free reactions with the $lysate_O RI$ used in Figure 5.1. Then, we fixed the maximum concentration for: a, Mg-glutamate concentration at 4 mM and b, K-glutamate at 80 mM. The error bars stand for the standard deviation of 3 replicates performed on the same day.

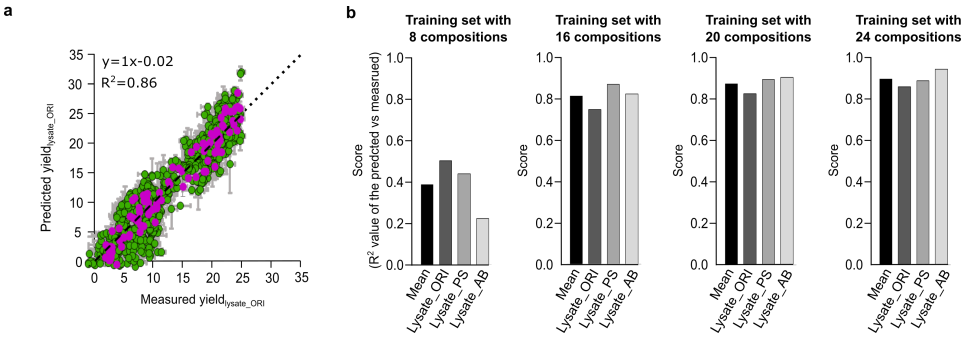


Figure 5.4 **The choice of 102 cell-free compositions for training and testing of our model. a** Distribution of the yields obtained with the 102 training cell-free compositions along the 1017 cell-free compositions tested in Figure 5.1. The 102 cell-free compositions were chosen based on the highest R^2 obtained by training on 102 points and predicting on the 915 remaining points. The vertical error bars stand for the standard deviation of 3 replicates. The horizontal error bars stand for the standard deviation of 25 predictions. **b** Comparison of the prediction efficiency of the model when trained with a training set of 8, 16, 20 or 24 cell-free compositions, for prediction on the reminder of the 102 points. The training set is chosen among the 102 cell-free compositions fixed in panel **a**. The training set leading to the highest mean R^2 among the 3 lysates has been selected.

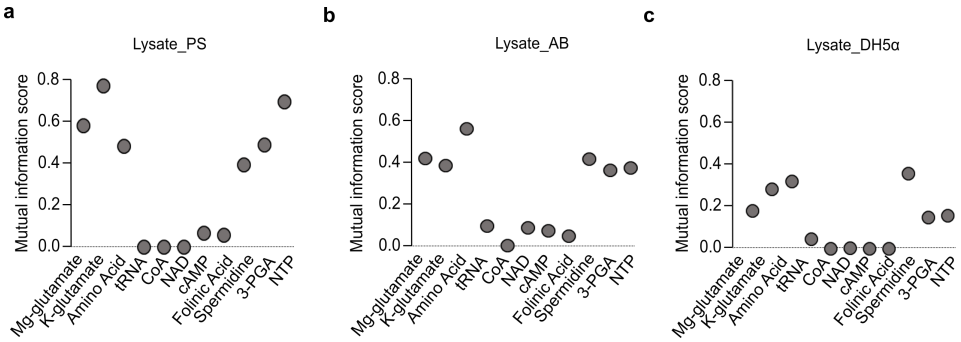


Figure 5.5 **Mutual Information analysis based on the 102 compositions tested with $lysate_{PS}$, $lysate_{AB}$ and $lysate_{DH5\alpha}$** Mutual information analysis of the relationship between the yield and each chemical compound, using the yields measured in cell-free reactions using 102 cell-free compositions and **a**, $lysate_{PS}$, **b**, $lysate_{AB}$, **c**, $lysate_{DH5\alpha}$.

Promoter J23101	tttacagctagctcagtcctaggtattatgctagc
RBS B0034	aaagaggagaaa
sfgfp	atgcgtaaaggcgaagagctgttcactggtgtcgtccctattctggtggaac tggatggtgatgtcaacggtcataagttttccgtgcgtggcgagggtgaagg tgacgcaactaatggtaaactgacgctgaagttcatctgtactactggtaaa ctgccggtaccttggccgactctggtaacgacgctgacttatggtgttcagt gctttgctcgttatccggaccatatgaagcagcatgacttcttcaagtccgc catgccggaaggctatgtgcaggaacgcacgatttcctttaaggatgacggc acgtacaaaacgcgtgcggaagtgaaatttgaaggcgataccctggtaaacc gcattgagctgaaaggcattgactttaaagaagacggcaatatcctgggcca taagctggaatacaattttaacagccacaatgtttacatcaccgccgataaa caaaaaaatggcattaaagcgaattttaaaattcgccacaacgtggaggatg gcagcgtgcagctggctgatcactaccagcaaaacactccaatcggtgatgg tcctgttctgctgccagacaatcactatctgagcacgcaaagcgttctgtct aaagatccgaacgagaaacgcgatcatatggttctgctggagttcgtaaccg cagcgggcatcacgcatggtatggatgaactgtacaaatga
rrnB T1 terminator	ccaggcatcaaataaaacgaaaggctcagtcgaaagactgggcctttcgttt tatctgttgtttgtcggtgaacgc tctc

Table 5.1 **Sequence of the plasmid used in this study** The sfgfp is under control of the promoter J23101 and RBS B0034. The plasmid contains the gene of ampicillin resistance and the origin of replication PBR322.

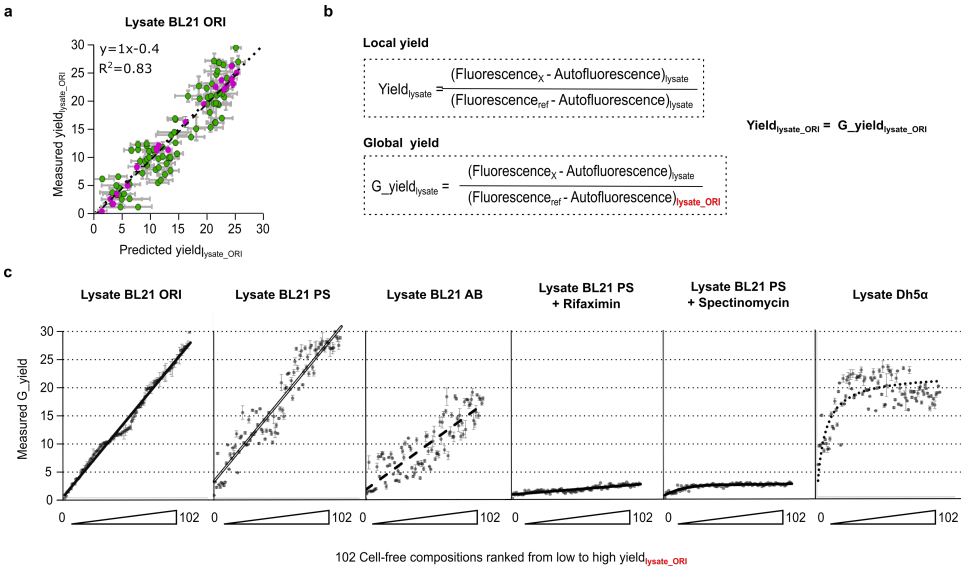


Figure 5.6 **Global comparison between the yields obtained with different lysates a** Comparison of the yields obtained with the lysate original (same as Figure 5.1) vs the model predictions for the 102 cell-free compositions used in Figure 5.2, **b** Formula of the global yield compared to the local yield. In contrary to the Yields presented in Figure 5.2, the Global yield always use the same reference yield from the lysate of Figure 5.1 named $Lysate_{ORI}$. The Global yield, noted G_{yield}, allows comparison between yields obtained with our different lysates. **c**, The 102 cell-free compositions were ranked from low to high values based on the yields obtained with the $Lysate_{ORI}$. The same ranking of the same 102 cell-free compositions was used for each lysate. Linear fit is used for $Lysate_{ORI}$, $Lysate_{PS}$, $Lysate_{AB}$ and $Lysate_{PS}$ + Riflaximin. Michaelis-Menten like fit is used for $Lysate_{PS}$ + Spectinomycin and $Lysate_{DH5\alpha}$.

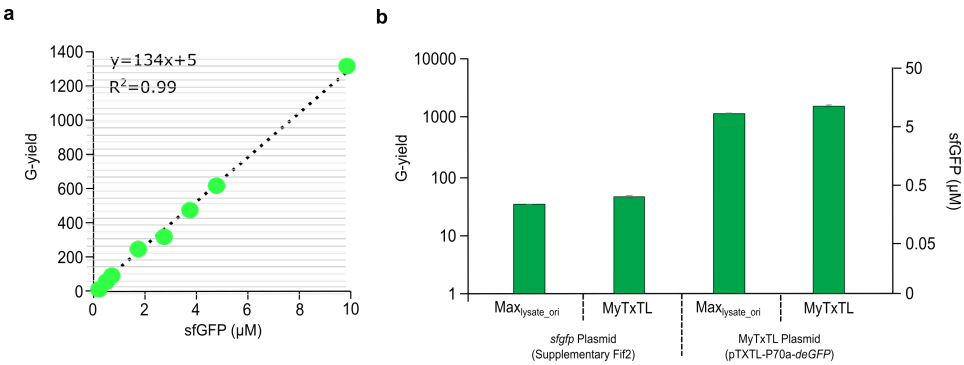


Figure 5.7 **Comparison between the behavior of the local yields measured with different lysates and the yields measured with the** $lysate_{ORI}$ Comparison between the yields measured with $lysate_{ORI}$ and **a**, $Lysate_{PS}$. **b**, $Lysate_{AB}$. **c**, $Lysate_{PS}$ + rifaximin. **d**, $Lysate_{PS}$ + spectinomycin. **e**, $Lysate_{DH5\alpha}$. The blue lines stand for linear fit and the dot lines stand for the perfect correlation (intercept 0 and slope 1). We used the same 102 cell-free compositions for all the measurements. The error bars stand for the standard deviation of 3 replicates.

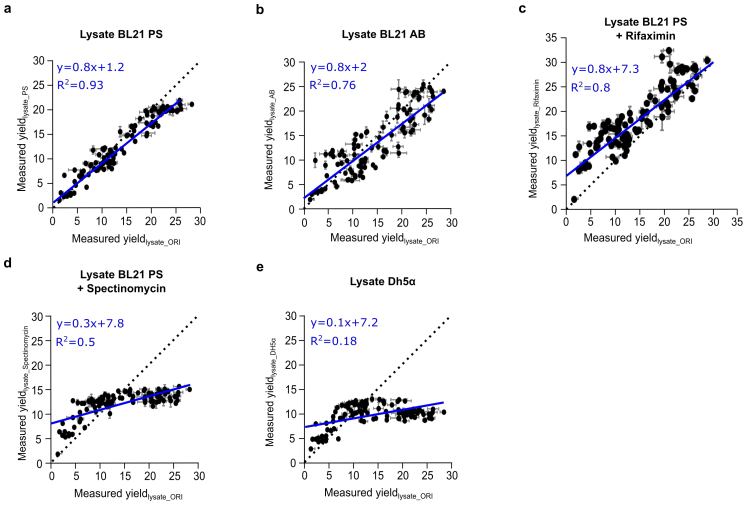


Figure 5.8 **A decrease in ribosome availability is sufficient to explain the saturation of the yields with** $Lysate_{Spectinomycin}$ **a** Comparison between the yield obtained with $Lysate_{PS}$ and the yield obtained with $Lysate_{PS}$ supplemented with Spectinomycin (same data as Supplementary Figure 5.7d). We used a Michaelis-Menten like function to fit the data. **b**, We used the well described Michaelis-Menten [141] like relationship between translation efficiency and available ribosomes concentration (Rfree). We assumed that a change in cell-composition impact the translation efficiency via a change of V_{max} and K_M. At a fixed Rfree concentration (blue arrow), an increase of V_{max}, K_M values lead to an increasing translation efficiency. **c**, As the spectinomycin binds to the 30S sub-unit of the ribosome to inhibit the translation process, its activity can be represented by a decrease in Rfree concentration (red arrow). The impact of less ribosomes will lead to a decrease in translation efficiency (blue vs red line in the second plot). **d**, Relationship between a translation efficiency with spectinomycin versus a translation efficiency without spectinomycin (see supplementary note 1). The yield as the protein production results from the translation but also the transcription process. The relationship between Translation efficiency and yields is described in supplementary note 1.

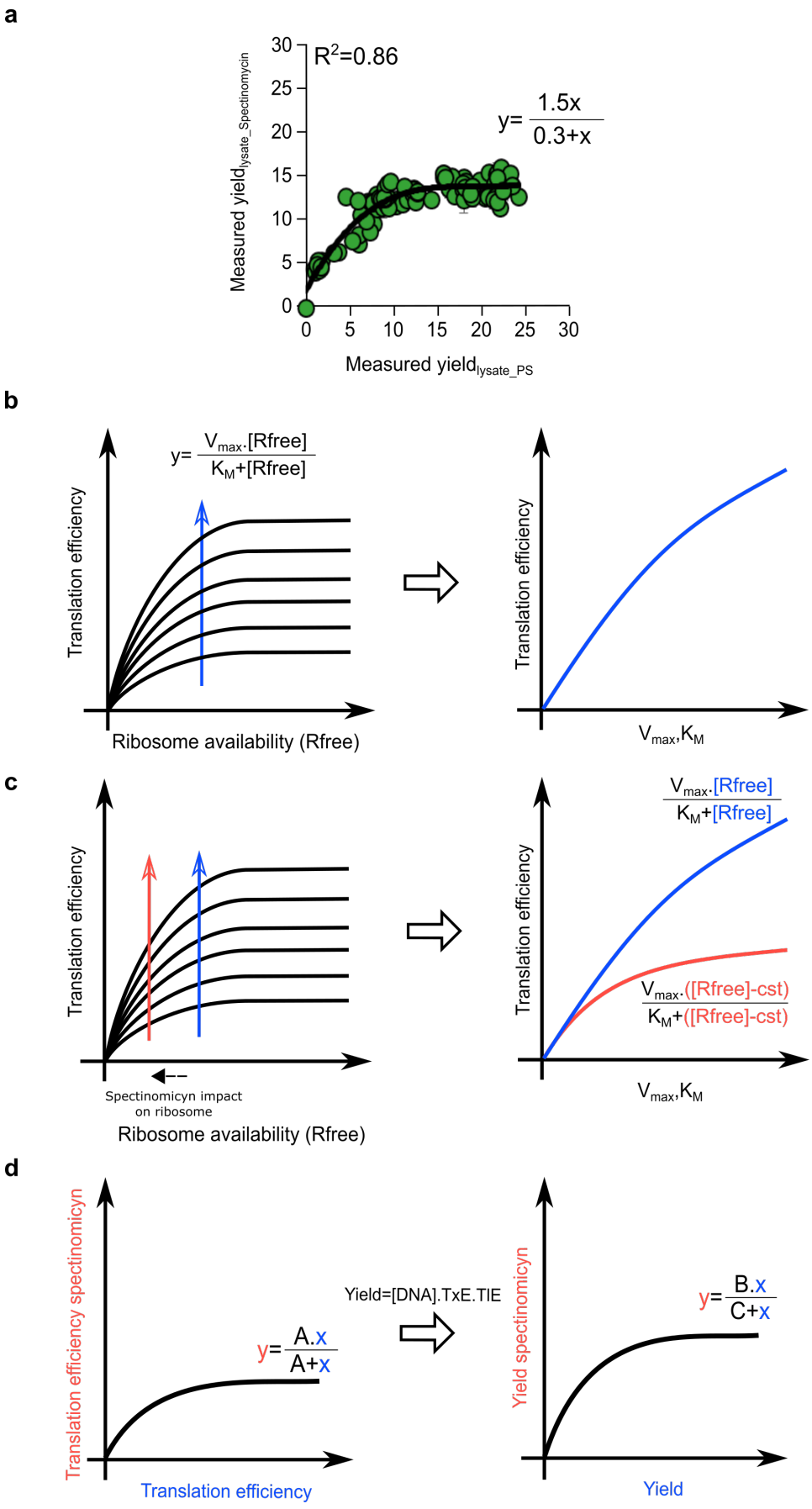


Figure 5.9 **Absolute measurements in cell-free reaction a** Relationship between purified sfGFP and the Global yield. (See supplementary note 2). **b** Comparison between the yield obtained with our best cell-free composition with $lysate_{ORI}$ and the commercial kit myTXTL from Arbor (pTXTL-P70a(2)-deGFP). We used both our plasmid and myTXTL plasmid. Noted that the y-axis with sfGFP concentration is used only for the measurements with our plasmid as myTXTL plasmid produce deGFP and not sfGFP.

5.4.1 Note 1: Deterministic model of protein production behavior in cell-free system with an impaired translation process (Supplementary Figure 5.8)

Assumption 1: Adding spectinomycin lead to a similar impact on the translation process as a decrease in concentration of the available ribosome. Spectinomycin binds to the 30S sub-unit stopping protein synthesis. Thus, a subset of ribosomes should be unavailable for translation.

$$[Rfree]_{spec} = [Rfree] - cst$$

Assumption 2: We simplified our calculation by considering that a variation in cell-free composition has a similar impact on both V_{max} and K_M.

$$V_{max} = cst2 * K_M$$

Assumption 3: The relationship of transcription efficiencies (noted TxE) between lysates is modeled by a linear relationship with a negligible intercept. We observed such a linear relationship (with an intercept close to 0) between yields from lysates with and without a damage transcription machinery in Supplementary Figure 5.7c.

$$TxE_{spec} = cst3 * TxE$$

Assumption 4: The variation in cell-free composition mainly impact the translation process. We observed in Supplementary Figure 5.7d that a lysate with a damaged translation machinery is poorly improved by a change in cell-free composition. The opposite is observed with a damaged transcription machinery in Supplementary Figure 5.7c suggesting that the efficiency of the translation machinery is the limiting factor for cell-free improvement and not the efficiency of the transcription machinery.

$$TxE = cst_4$$

(TxE is independent of the variations in cell-free compositions)

We used the well-defined model of the translation efficiency (TlE) based on a Michaelis-Menten equation [172].

$$TlE = \frac{V_{max}.[Rfree]}{K_M + [Rfree]} \tag{5.1}$$

$$TlE_{spec} = \frac{V_{max}.[Rfree]_{spec}}{K_M + [Rfree]_{spec}} \tag{5.2}$$

where V_{max} and K_M values depends on the RBS sequence and the cell-free composition. $[Rfree]$ stands for the concentration in available ribosomes. Using assumption 1 and equation 5.2: $[Rfree]_{spec} = [Rfree] - cst$, we obtain:

$$TlE_{spec} = \frac{V_{max}.[Rfree] - cst}{K_M + [Rfree] - cst} \tag{5.3}$$

Using assumption 2 ($V_{max} = cst_2 * K_M$) and equations 5.1 and 5.3:

$$TlE = \frac{cst_2 * K_M.[Rfree]}{K_M + [Rfree]} \tag{5.4}$$

$$TlE_{spec} = \frac{cst_2 * K_M.[Rfree] - cst}{K_M + [Rfree] - cst} \tag{5.5}$$

Thus,

$$5.4 \Leftrightarrow K_M = \frac{TlE.[Rfree]}{cst_2 * [Rfree] - TlE} \tag{5.6}$$

$$\begin{aligned}
5.5\&5.6 \Leftrightarrow TlE_{spec} &= \frac{cst_2.\frac{TlE.[Rfree]}{cst_2.[Rfree]-TlE}.([Rfree] - cst)}{\frac{TlE.[Rfree]}{cst_2.[Rfree]-TlE} + [Rfree] - cst} \\
\Leftrightarrow TlE_{spec} &= \frac{cst_2.TlE.[Rfree].([Rfree] - cst)}{TlE.[Rfree] + ([Rfree] - cst).(cst_2.[Rfree] - TlE)} \\
\Leftrightarrow TlE_{spec} &= \frac{cst_2.TlE.[Rfree].([Rfree] - cst)}{TlE.cst + ([Rfree] - cst).cst_2.[Rfree]} \\
\Leftrightarrow TlE_{spec} &= \frac{\frac{cst_2.[Rfree].([Rfree]-cst)}{cst}.TlE}{TlE + \frac{cst_2.[Rfree].([Rfree]-cst)}{cst}} \\
\Leftrightarrow TlE_{spec} &= \frac{A.TlE}{TlE + A}
\end{aligned} \tag{5.7}$$

with

$$A = \frac{cst_2.[Rfree].([Rfree] - cst)}{cst} \tag{5.8}$$

The protein production (and so the yield) is the result of the expression of sfgfp by the transcription and translation processes.

$$Yield = [DNA].TxE.TlE \tag{5.9}$$

$$Yield_{spec} = [DNA].TxE_{spec}.TlE_{spec} \tag{5.10}$$

Using assumption 3 ($TxE_{spec} = cst_3 * TxE$), and the fact that the DNA concentration is the same in every cell-free reaction ($[DNA] = cst_5$):

$$5.10 \Leftrightarrow Yield_{spec} = cst_5.cst_3.TxE.TlE_{spec} \tag{5.11}$$

Using assumption 4 ($TxE = cst_4$), we have:

$$5.9 \Leftrightarrow Yield = cst_5.cst_4.TlE \tag{5.12}$$

$$5.9 \Leftrightarrow Yield_{spec} = cst_5.cst3.cst_4.TlE_{spec} \tag{5.13}$$

Then,

$$5.12 \Leftrightarrow TlE = \frac{Yield}{cst_5.cst_4} \tag{5.14}$$

and

$$5.13 \Leftrightarrow TlE_{spec} = \frac{Yield_{spec}}{cst_5.cst_4.cst_3} \tag{5.15}$$

Then,

$$5.7\&5.15 \Leftrightarrow \frac{Yield_{spec}}{cst_5.cst_4.cst_3} = \frac{A.TlE}{TlE + A} \tag{5.16}$$

Then,

$$\begin{aligned} 5.12\&5.16 \Leftrightarrow \frac{Yield_{spec}}{cst_5.cst_4.cst_3} &= \frac{A.\frac{Yield}{cst_5.cst_4}}{\frac{Yield}{cst_5.cst_4} + A} \\ \Leftrightarrow Yield_{spec} &= \frac{cst_5.cst_4.cst_3.A.\frac{Yield}{cst_5.cst_4}}{\frac{Yield}{cst_5.cst_4} + A} \\ \Leftrightarrow Yield_{spec} &= \frac{cst_5.cst_4.cst_3.A.Yield}{Yield + A.cst_5.cst_4} \\ \Leftrightarrow Yield_{spec} &= \frac{B.Yield}{Yield + C} \end{aligned} \tag{5.17}$$

With

$$B = \frac{cst_5.cst_4.cst_3.cst_2.[Rfree].([Rfree] - cst)}{cst}$$

and

$$C = \frac{cst_5.cst_4.cst_2.[Rfree].([Rfree] - cst)}{cst}$$

Eventually, we obtained a Michaelis-Menten equation for the relationship between $Yield$ and $Yield_{spec}$ (eq. 5.17) which explain the data in Supplementary Figure 5.8a. Despite the multiple assumptions (that are difficult to verify by experimental measurements) this model gives a simple explanation of our observations.

5.4.2 Note 2: Commercial kit and absolute sfGFP measurements (Supplementary Figure 5.9)

Both plasmids (our plasmid and myTXL plasmid) led to similar yield when the $lysate_{ORI}$ with the optimized composition (max yield in Figure 5.2) and myTXTL

mix are used. This result suggests that pTXTL-P70a(2)-deGFP can also be used, instead of our plasmid to optimize cell-free composition. The higher Global yield come from the higher fluorescence obtained with this plasmid. The pTXTL-P70a(2)-deGFP seems to be a derivative of the pBEST-OR2-OR1-Pr-UTR1-eGFP-Del6-229-T500 [320] optimize for expression in cell-free reaction. We don't have access to the cell-free composition of myTXTL mix but we assumed that it was optimized to obtain a maximum protein production and that the lysate was prepared from a modified strain of *E. coli*. The quality of the result obtained with our lysate-specific optimization compared to the commercial kit is a validation of our method efficiency. The protein concentration obtained from the expression of our plasmid with $lysate_{ORI}$ is at 0.22 µM sfGFP equivalent. We can notice, with the arbor plasmid, that the the 7 µM sfGFP equivalent is irrelevant as the plasmid produce deGFP.

Part II

Analyzing and modeling metabolic circuits: from data to knowledge

6 Custom-Made Transcriptional Biosensors for Metabolic Engineering

This work was published in Current Opinion in Biotechnology by Mathilde Koch *, Amir Pandi *, Olivier Borkowski, Angelo Cardoso Batista and Jean-Loup Faulon. Only minor modifications to the published paper have been introduced in the Chapter below.
* stands for equal contributions.

Detailed contribution to this thesis

This Chapter describes advances in transcriptional biosensors for metabolic engineering. Transcriptional biosensors form an essential part of any circuit in synthetic biology, as they allow for output detection. This is the reason why I also studied those biosensors in the context of this thesis. Given the number of reviews published on the topic, my first contribution to this article was to define an outline that would point out the next challenges and opportunities in the field that are less often mentioned in other review, namely using mathematical modeling for fine tuning of biosensor properties and the importance of developing biosensors with and for the cell-free systems branch of synthetic biology. I then did the bibliographical research and wrote the text of the article, while A.P. did the figures, O.B. and A.C.B. did the cell-free experiments presented in Figure 6.3 and all authors read the manuscript and approved it for publication.

Full reference

Koch M., Pandi, A., Borkowski O., Cardoso Batista A., Faulon J.-L. (2019) Custom-made transcriptional biosensors for metabolic engineering *Current Opinion in Biotechnology*, 10.1016/j.copbio.2019.02.016.

Contributions as stated in the article

Not available.

6.1 Abstract

Transcriptional biosensors allow screening, selection or dynamic regulation of metabolic pathways, and are therefore an enabling technology for faster prototyping of metabolic engineering and sustainable chemistry. Recent advances have been made, allowing for routine use of heterologous transcription factors, and new strategies such as chimeric protein design allow engineers to tap into the reservoir of metabolite-binding proteins. However, extending the sensing scope of biosensors is only the first step, and computational models can help in fine-tuning properties of biosensors for custom-made behavior. Moreover, metabolic engineering is bound to benefit from advances in cell-free expression systems, either for faster prototyping of biosensors or for whole-pathway optimization, making it both a means and an end in biosensor design.

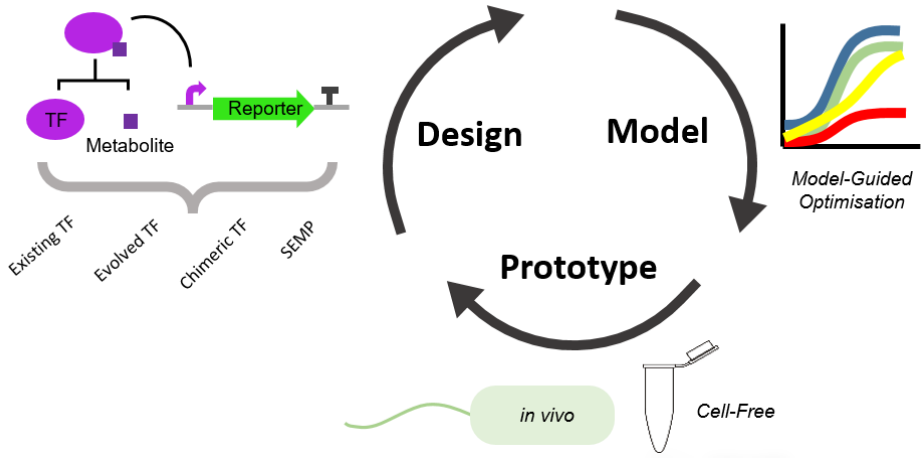


Figure 6.1 **Graphical abstract for biosensor review**

6.1.1 Highlights

- Successful examples of transcriptional biosensor implementations are presented
- Various engineering strategies extend the space of detectable chemicals
- Novel strategies exist to transform metabolite-responding proteins into biosensors
- Mathematical models of varying complexity can help tune biosensor properties
- Biosensors using or designed for cell-free systems are presented

6.2 Introduction

Metabolic engineering allows the production of value-added compounds from renewable sources, therefore making it a key discipline for a greener and more sustainable

chemistry. As the domain of synthetic biology has matured, numerous techniques have been developed and applied in metabolic engineering, allowing for cheaper and faster DNA synthesis, sequencing and assembly. It is nowadays faster to design and build constructs than to characterize them as testing often involves expensive mass spectrometry analyses. This has led to an increased interest in biosensors, which can allow fast and real-time screening, selection or dynamic regulation engineering of metabolic pathways. Cells harboring fluorescent proteins as the reporter of the biosensor allow screening of a huge number of variants, both for experimental growth conditions or genetic constructs (enzymes, RBS, promoters). Moreover, dynamic regulation can be used to monitor intermediates, final products or quorum molecules, allowing for optimal pathway balancing and resource consumption.
The advantages of using biosensors in metabolic engineering have been extensively reviewed before [321, 322, 323]and will not be detailed further. Moreover, a wide array of techniques now exists to develop biosensors, from Förster Resonance Energy Transfer (FRET) [324] to riboswitches [325, 326]: the interested reader is referred to those two excellent reviews that cover the strengths and limitations of the abovementioned technologies [327, 328]. In this review, we will focus on transcriptional biosensors in three different aspects. First, we will review techniques for discovery and engineering of transcriptional biosensors for new compounds, second, we will present how computer-assisted modeling can facilitate the tuning of biosensors for custom-made behavior, and third we will review the advances and advantages of using cell-free systems for biosensor characterization and metabolic engineering.

6.3 Designing a transcriptional biosensor to detect a compound of interest

6.3.1 Engineering allosteric transcription factors

The first step to engineer a biosensor, whether homologous or heterologous, is to identify the transcription factor (TF) and promoters that respond to it. Strategies involving transcriptional micro-arrays and identification of the up- or down-regulated genes in response to the ligand of interest provide first leads. These approaches can suffer from important limitations for metabolic engineering use: the identified genes can be either indirectly regulated by the ligand of interest, or very unspecific. This strategy has been successfully applied for 1-butanol detection [329]. Another strategy for identification of potential TF-promoter pair comes from Zhang *et al.* [330] who identified pairs that could detect lactam derivatives: the authors used a cheminformatics approach to reveal operons listed in BRENDA [133] that detected similar chemicals, and identified the gene likely coding the transcription factor. We recently published [76] (Chapter 4) a dataset of detectable metabolites (6.2a). This dataset, includes a manually curated list of experimentally validated detectable metabolites and information from databases of regulation, which contain known or putative detectable metabolites. Other strategies for mining parts have been discussed in a previous review [331].

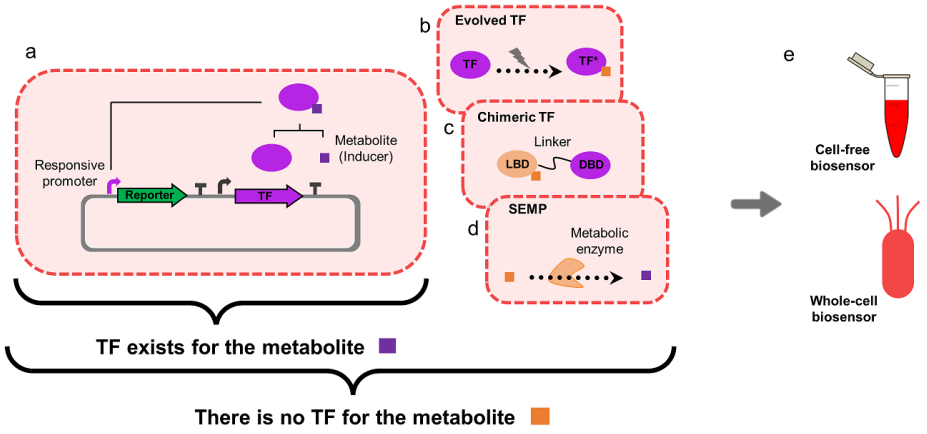


Figure 6.2 **Different strategies to develop a TF based biosensor for a given metabolite.** There is either an existing TF for a metabolite **a** or it could be engineered using evolved TF **b**, chimeric protein **c**, or a metabolic pathway (SEMP) **d** A designed biosensor could be implemented in whole-cell or cell-free system **e** Abbreviations: TF: Transcription Factor, LBD: Ligand Binding Domain, DBD: DNA Binding Domain, SEMP: Sensing-Enabling Metabolic Pathways.

Compound	Original organism	Implementation organism	Design strategy	Biosensor application	Reference
Itaconic acid	*Yersinia pseudotuberculosis*	*E. coli*	Identified TF and promoter from catabolism pathways	Used for enzyme improvement in pathway prototyping	[332]
Vanillin	*E. coli*	*E. coli*	Natural *E. coli* regulator tuned with mathematical modeling	Used for library screening	[333]
Syringaldehyde	*E. coli*	*E. coli*	Natural *E. coli* regulator tuned with mathematical modeling	Used for library screening	[333]
Muconic acid	*Acinetobacter* sp. ADP1	Saccharomyces cerevisiae	Identified from a previous publication	Used for selection of high producing strains	[334]
Pinocembrin	*Herbaspirillum seropedicae*	*E. coli*	Tuned with the help of a mathematical model	Can be used for metabolic engineering	[171] (Chapter 7)
Pamamycin	*Streptomyces albboniger*	*Streptomyces albboniger*	Improved from native genetic elements	Can be used for metabolic engineering	[335]

p-coumaric acid	*Bacillus subtilis*	*E. coli*	Identified from literature and library design of RBS to tune the repressor properties	Used for screening a producer strain in microfluidic droplets	[336]
Formaldehyde	*E. coli*	*E. coli*	Optimized from native regulatory elements.	Used to identify promising enzymes for methanol assimilation	[337]
N-acetylneuraminic acid	*E. coli*	*E. coli*	Modularization of the native biosensing system	Used for screening high-producing strains	[338]
Putrescine	*E. coli*	*E. coli*	Modularization of the native biosensing system	Used for screening high-producing strains	[339]
L-phenylalanine	*E. coli*	*E. coli*	Modularization of the native biosensing system	Used for screening high-producing strains	[340]
Shikimic acid	*Corynebacterium glutamicum*	*Corynebacterium glutamicum*	Using the promoter from native genetic elements, considering the transcription factor to be naturally expressed	Used for screening high-producing strains	[341]
Cellobiose	*Thermobifida fusca*	*E. coli*	Identified from literature and expressed in *E. coli*	Used to identify promising cellulases	[342]
Naringenin	*Herbaspirillum seropedicae*	*E. coli*	Identified from literature, modularized and expressed in *E. coli*	Can be used for metabolic engineering	[343]
Naringenin	*Acinetobacter* sp. ADP1	*Saccharomyces cerevisiae*	Identified from literature, modularized and expressed in *E. coli*	Used for pathway prototyping - screening	[344]
Various aromatic blocks	*Sphingobium* sp. SYK-6	*E. coli*	Identified from literature, modularized and expressed in *E. coli*	Used to screen for lignin degrading enzymes	[345]

Various macrolides	MphR, isolated from wastewater treatment plant	*E. coli*	Directed evolution and random mutagenesis to improve selectivity	Can be used for metabolic engineering	[346]

Table 6.1 Successful homologous and heterologous biosensor design based identified on transcription factor/promoter pairs.

Once a potential TF/promoter pair is identified, the bio-engineering workflow involves modifying the promoter, RBS and binding sites to improve selectivity, dynamic, operational range, fold change and leakiness. A number of successful biosensors have been developed in recent years, including heterologous TF despite the challenges faced to adapt the transcriptional machinery. This technology is becoming increasingly mature, as shown by the numerous examples in Table 6.1. In addition, engineering of specific biosensors for Malonyl-CoA is reviewed by Johnson *et al.* [347], while Ambri [348] describes in details an implementation of bacterial TF in yeast. Voigt's group recently published an *E. coli* strain containing twelve genomically integrated small molecule sensors, using a directed evolution strategy. It has been developed as a synthetic biology tool but the presented methods are applicable to metabolic engineering [349].
However, the above-mentioned strategies are only applicable if a natural transcription factor-biosensor pair exists for a given compound. We will now review strategies to extend the chemical scope of transcriptional biosensors.

6.3.2 Extending the chemical space for biosensors

A strategy to extend the chemical scope is to start from a known transcription factor and apply rounds of protein engineering to change its specificity (6.2.b). For example, to design a biosensor for lactulose, LacI was altered using saturation mutagenesis, with rounds of selection to ensure specificity to lactulose [350]. Taylor *et al.* [351] used computer-assisted protein design, followed by saturation or random mutagenesis to modify LacI to sense either fucose, gentiobiose, lactitol or sucralose. The promiscuous MphR transcription factor has been modified with a similar strategy to change its selectivity towards various macrolides [346]. Despite their successes, these examples still rely on well-known transcription factors and labor-intensive mutagenesis or computationally assisted protein design to change the specificity of a transcription factor to, still, a chemically similar molecule.
Several groups have tried radical approaches, fusing DNA Binding Domain (DBD) to determine ligand binding domains in different ways (6.2.c). This strategy has been successfully applied to maltose [352] and benzoate [353] by testing various linkers and DBD systematically. Another strategy, also applied to maltose, was to randomly insert the DBD into the metabolite binding protein, using transposon insertion reaction, to select constructs presenting biosensor-like behavior [354]. In a recent study [355], the authors use a ligand dependent stabilization strategy, fusing LacI (respectively MphR) to the Zif268 DBD and ribonucleic acid (RNA) poly-

merase ω-sub-unit transcription-activating domain. Those constructs are quickly degraded unless the ligand is present. The authors managed to engineer biosensors responding to Isopropyl β-D-1-thiogalactopyranoside (IPTG) and D-glucose with satisfying dose-response (respectively erythromycin with a modest response). However, to underline the difficulty of this approach, they report that in two structurally similar periplasmic binding proteins, a similar mutation did not confer ligand dependent stabilization. Another similar approach was developed recently, it uses both ligand dependent stabilization and protein dimerization: two ligand binding domains (that can homodimerize, but bind different ligands) are fused respectively to the activation domain and the DBD. Upon ligand binding, the two proteins are stable and can homodimerize, resulting in biosensing. This system allows for better range tuning and possible orthogonal biosensing of different ligands [356].
Other known metabolite responsive proteins are two-component systems, which have also been used as biosensors. By fusing the transmembrane sensing domain of another species detecting methanol with the cytoplasmic phosphorylation domain of *E. coli*, binding of methanol activates a phosphorylation cascade enabling biosensing [357]. In an elegant study, transmembrane and cytosolic receptors for caffeine were built by fusing single-domain antibodies to monomeric DBDs [358]. Different DBDs were used, proving the scalability of the method. These two platforms should allow bio-engineers to tap into the vast reservoir of two-component systems and antibodies to design new sensors.
A radically different approach to engineer the sensing scope of bacteria was coined Sensing-Enabling Metabolic Pathways (SEMP) (6.2.d). The principle of this method is to metabolically convert an undetectable ligand into an already detectable one. This method makes the most of existing biosensors as well as of the impressive accumulated knowledge on metabolic reactions. It has been successfully applied in a metabolic engineering project to produce 3-hydroxypropionate [359], and its modularity was shown by Libis *et al.* [262]. A web-server is now available to design SEMP for compounds of interest [77].

6.4 Computer-assisted fine-tuning of biosensor properties

While the scientific community agrees that biosensors need to be fine-tuned for selectivity, sensitivity and dynamic range, tuning strategies are usually based on labor-intensive and costly rounds of selection and mutagenesis. Controlling those properties is especially interesting for metabolic engineering as the specifications of a biosensor needed during various stages of the process will change, from detecting micromolar amounts before pathway optimization to $g \cdot L^{-1}$ titers in later development stages. Therefore, after engineering a biosensor with new specificity, its properties also need to be fine-tuned to match the metabolic engineer's needs.
A detailed mechanistic model of the ArsR arsenic biosensor was developed by Berset *et al.* [143], which recapitulates the sensor behavior under various circuit configurations, different ArsR alleles, promoter strengths, and presence or absence of arsenic

efflux in the bio-reporters. This model was then used to predict a circuit variant with steeper response at low arsenite concentrations. A thermodynamical model was developed in a recent study [360], which was used to tune the dynamic range of ligand-inducible promoters (mainly AraC and LasR), using binding energies calculated for different promoter sequences. Both studies proved that with sufficiently detailed models, tailoring biosensor properties for custom-made behavior can be achieved. Another interesting study based on the Lac system and involving extensive phenomenological modeling sought to find theoretical constraints for biosensor design, notably a maximum achievable dynamic range and exposing tunable parameters for orthogonal control of dynamic range and response threshold [361]. As impressive as these studies are, they are based on well-characterized and known systems and such modeling cannot be applied easily to a new biosensor.
However, a simpler formalism (Michaelis-Menten) for mathematical modeling was used to tune a biosensor used for selection of lignin transforming enzymes, giving insights on parameters influencing sensitivity, such as TF concentrations or copy number [333]. The role of plasmid copy number on sensitivity and fold-change of a pinocembrin and naringenin biosensor was investigated through a mathematical modeling [171] (Chapter 7), using the common Hill framework, allowing for better understanding of the biosensor behavior and suggestions for further tuning of properties according to desired outputs. Landry *et al.* [362] used mathematical modeling with Hill formalism to tune the detection range of a two-component system. They successfully applied it to improve their detection threshold up to two orders of magnitude. These later studies showed that simple mathematical models can help to understand and tune specific properties of a biosensor, even in less known systems.
Computer-assisted design does not always yield the expected results, as current models are often more explicative a posteriori than predictive a priori. Therefore, we believe investing the time needed to develop reliable models for a library of constructs can only be worthwhile in the long run for designing biosensors, as formalized knowledge is more easily translatable to other situations.

6.5 Custom-made biosensors' new application domain: cell-free metabolic engineering

Despite the advances presented in this review, biosensor design still necessitates rounds of trial and error. This limitation can be significantly sped up by using cell-free systems (6.2.e). Moreover, cell-free systems, are poised to become a key characterization tool in the metabolic engineering workflow before *in vivo* implementation. Cell-free systems lead to quicker responses, simpler cloning and larger combinatorial libraries screening, without requiring transformation steps. These systems can also be an appropriate platform for production because of lower noise and toxicity and absence of resource competition between pathway and cell growth. To date, cell-free systems have been applied to implement pathways for violacein [363], 4-BDO [364], polyhydroxyalkanoates bioplastics [365], mevalonate [52] [71],

n-butanol [72] and raspberry ketone [366], using either transcription-translation (TX-TL) systems, over-expressed enzymes in the crude extract or purified enzymes. Advantages and possibilities of cell-free systems for metabolic engineering has been reviewed elsewhere [367], and a methods chapter for pathway prototyping in cell-free systems has recently been published [368].

Cell-free biosensors for various applications have been reviewed elsewhere [369] and we will focus on strategies applicable to metabolic engineering. In a recent study, a vanillin biosensor was developed in cell-free systems [70]. The authors first used computational protein design and then rapid cell-free prototyping to develop a biosensor for this toxic effector, which was subsequently used in dynamic control loops *in vivo* to alleviate toxicity.

Another use of cell-free systems for biosensor engineering is discussed by Halleran *et al.* [73]: to develop complex consortia for pathway distribution among cell populations, metabolic engineers need reliable quorum sensing systems. This study characterized cross-talk between cell-free system and *in vivo* and discovered significant correlation between cell-free and *in vivo* measurements, validating the use of cell-free systems as a successful and fast characterization testbed. Recently, we have proposed a method to optimize the response of TF-based cell free biosensors. We also proved that SEMP are modularly implementable in cell-free systems, and exhibit high sensitivity, fast response times and broad dynamic range [173].

For this review, we implemented our *in vivo*-characterized pinocembrin biosensor [171] (Chapter 7) in a cell-free system (Figure 6.3.a). The cell-free biosensor exhibited a linear correlation between input concentration and fluorescence intensity as well as a wider dynamic and operational range (Figure 6.3.b) compared to its *in vivo* counterpart [171] (Chapter 7). These tools could be used for real-time screening and speed up the DBTL workflow for metabolic engineering.

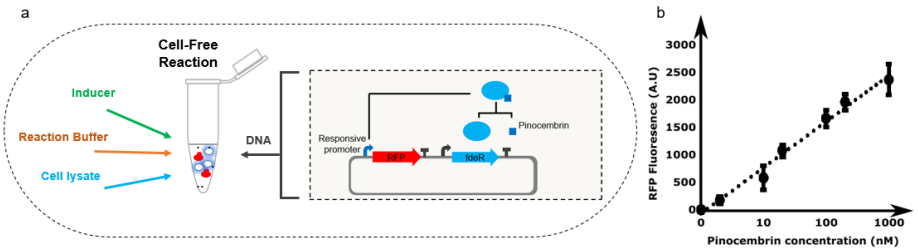


Figure 6.3 **Pinocembrin cell-free biosensor.** Cell-free reaction consists of TX-TL cell lysate, reaction buffer and DNA plus inducer for the biosensor **a**. **b** The graph shows a dose response red fluorescent protein (RFP) fluorescence after 9 hours incubation in a plate reader at 30 °C. 40 nM of biosensor plasmid is added with 0, 1, 2, 10, 20, 100, 200 or 1000 µM of pinocembrin in 10.5 µL of cell-free reaction. RFP fluorescence points and error bars are the mean and standard deviation of three measurements.

Cell-free systems provide fascinating new opportunities for metabolic engineering, both for faster biosensor development, notably for toxic products, but also for prototyping whole pathways. Cell-free based metabolic engineering can benefit

from all advantages of biosensor-based screening or dynamic regulation engineering, as does traditional metabolic engineering.

6.6 Conclusion

Thanks to extensive efforts by the research community, it has never been easier to develop transcriptional biosensors for new compounds, either from existing TF or engineering strategies. We believe the next frontier in custom-made biosensor design resides in efficient fine-tuning of properties, which is greatly advanced by modeling efforts. Moreover, metabolic engineering might be entering a new phase, with cell-free systems enabling faster prototyping of biosensors and even whole pathways. The current advances in biosensors for high-throughput screening will truly allow the field to move from the Design-Build-Test cycle to the Design-Build-Test-Learn cycle.

7

Building a minimal and generalisable model of transcription factor-based biosensors: Showcasing flavonoids

This work was published in the Journal of Biotechnology and Bioengineering by Heykel Trabelsi *, Mathilde Koch * and Jean-Loup Faulon.
Only minor modifications to the published paper have been introduced in the Chapter below. Supplementary data facilitating reading of my contributions was included where mentioned in the text, and noted as being originally Supplementary Figures.
* stands for equal contributions.

Detailed contribution to this thesis

As stated previously, biosensors are an essential part of any circuit in synthetic biology, as they allow for signal detection. This is the reason why developing, modeling and analyzing biosensors is the first step of the development of analysis tools for more complex circuits, as the signal detection layer determines all that can be understood of processes that appear upstream of the detection.
My main contribution was in the mathematical analysis and modeling of our biosensors. More precisely, I observed in our data that changing plasmid copy number of our biosensors not only modified the fold change of the biosensor, as is to be expected, but also their sensitivity. I therefore chose an adapted modeling strategy based on a modified Hill equation, that accounts for plasmid copy number through transcription factor and binding sites numbers, which is our system's degree of freedom. After fitting this model on available data and verifying parameter consistency using sampling from parameter estimations, the model was then used to suggest modifications for changing the biosensor behavior to attain desired criteria, such as higher or lower sensitivity, by modifying plasmid copy number, DNA - transcription factor binding strength or transcription factor - inducer binding strength.

Full reference

Trabelsi H *, Koch M *, Faulon J-L. (2018) Building a minimal and generalisable model of transcription factor–based biosensors: Showcasing flavonoids. *Biotechnology and Bioengineering*, 10.1002/bit.26726.
* stands for equal contributions.

Contributions as stated in the article

H. T., M. K., and J.-L. F. designed the study. H. T. designed, built, and characterized the biosensors. M. K. performed the chemical structure analysis, the mathematical analysis, and the modeling. All authors participated in the interpretation of the results and in the preparation of the manuscript.

7.1 Abstract

Progress in synthetic biology tools has transformed the way we engineer living cells. Applications of circuit design have reached a new level, offering solutions for metabolic engineering challenges that include developing screening approaches for libraries of pathway variants. The use of transcription factor-based biosensors for screening has shown promising results, but the quantitative relationship between the sensors and the sensed molecules still needs more rational understanding. Herein, we have successfully developed a novel biosensor to detect pinocembrin based on a transcriptional regulator. The FdeR TF, known to respond to naringenin, was combined with a fluorescent reporter protein. By varying the copy number of its plasmid and the concentration of the biosensor TF through a combinatorial library, different responses have been recorded and modeled. The fitted model provides a tool to understand the impact of these parameters on the biosensor behavior in terms of dose–response and time curves and offers guidelines to build constructs oriented to increased sensitivity or ability of linear detection at higher titers. Our model, the first to explicitly take into account the impact of plasmid copy number on biosensor sensitivity using Hill-based formalism, is able to explain uncharacterized systems without extensive knowledge of the properties of the TF. Moreover, it can be used to model the response of the biosensor to different compounds (here naringenin and pinocembrin) with minimal parameter refitting.

7.2 Introduction

Trends in metabolic engineering approaches to produce bio-based chemicals in cell factories are still under continuous improvements. The main developments include over-expressing the enzymes of the rate-limiting steps [370], deletion of competing

pathways [371], balancing cofactor and precursor metabolites [372, 373], implementing synthetic feedback loops [374, 375], and biosensor-based dynamic regulation [376].

One current challenging task is to set up a reliable method to screen for the best producing strains among a wide genetic diversity. The use of biosensors responsive to intracellular chemicals has opened doors to solving this pressing issue. Such sensory regulatory devices, mainly TFs, have successfully been used to detect the presence of metabolites, but also for quantification and even high throughput screening [377]. Furthermore, biosensors can also play an important role in regulating pathway fluxes by sensing the level of a key intermediate and then promoting its synthesis or its downstream conversion [376]. To overcome the limited number of naturally occurring metabolite responsive TFs available, progress has been made through their heterologous use, which includes transplantation of prokaryotic transcriptional activators into the eukaryotic chassis [344]. Additionally, it was recently shown that it is possible to expand the detection abilities by adding one or more enzymatic steps to transform a non-detectable compound into a detectable one [77, 262]. This latest tool considerably expands the scope of chemicals that can be sensed via transcriptional regulators.

One of the interesting metabolic pathways implemented with relative success is the flavonoid pathway [225]. The industrial demand for some flavonoids is increasing, and among the top promising chemicals is (2S)-pinocembrin, which is a plant secondary metabolite and the main starting point for the synthesis of other flavonoid molecules. This compound has a broad range of interesting characteristics such as antioxidant [378], antibacterial [379], antifungal [380], inhibitor of atherosclerosis [381], and neuroprotection in neurodegenerative diseases [382, 383]. To produce pinocembrin from glucose, four heterologous genes have to be implemented in *E. coli*. First, phenylalanine ammonia lyase converts phenylalanine into cinnamic acid, which is then converted by coumarate-CoA ligase into cinnamoyl-CoA. Then, chalcone synthase condensates cinnamoyl-CoA and three molecules of malonyl-CoA to produce pinocembrin chalcone, which will be then converted into pinocembrin through chalcone isomerase (Figure 7.1). As of today, pinocembrin is produced at a low titer from glucose ([384]; only $40\,\mathrm{mg}\cdot\mathrm{L}^{-1}$), and work still needs to be carried out to increase productivity, most likely through the building of combinatorial libraries with various enzyme sequences and regulatory elements (promoters, RBS). Such libraries could be quickly screened with a pinocembrin biosensor, where the level of the reporter gene (i.e., fluorescence) is proportional to the pinocembrin titer.

Chemical structure similarity considerations of detectable flavonoids led us to choose as our candidate FdeR TF, a transcriptional activator based biosensor from *Herbaspirillum seropedicae* SmR1, shown to respond to naringenin [385, 386]. Here, we have focused on developing and modeling the FdeR TF to shed light on the way we could design TF-based biosensors to overcome issues of measurable quantification of metabolite production and to monitor an adequate sensing response. We have built different constructs varying most notably in plasmid copy number, changing both the concentration of the TF and the number of binding sites for the activated complex, and modeled the impact of this varying number on the sensitivity of the

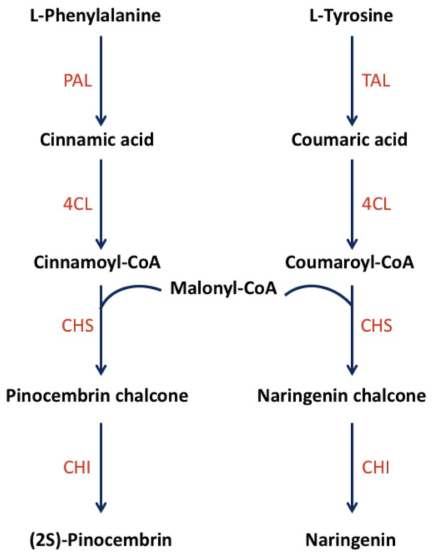


Figure 7.1 **Pinocembrin biosynthesis pathway** PAL, TAL, 4CL, CHS and CHI refer to phenylalanine ammonia lyase, tyrosine ammonia lyase, coumarate-CoA ligase, chalcone synthase, and chalcone isomerase, respectively

response. We provide a modeling strategy based on Hill functions to understand the impact of plasmid copy number and compound binding affinity to FdeR on our biosensor behavior, for both the dose-response and time curves, for a TF that has not been well characterized before.

7.3 Materials and methods

7.3.1 Plasmids and strains

All plasmids and strains used in this study are listed in the Supporting Information Table 7.5. *E. coli* strain DH5 (Life Technologies, Darmstadt, Germany) and Mach 1 strain (Invitrogen Technologies, Carlsbad, CA) were used for cloning. *E. coli* strain BL21 (DE3) was used for enzyme expression. All the primers (P1–P9) were purchased from Eurofins genomics (Ebersberg, Germany) and are listed in the Supporting Information Table 7.6. All our constructs were built by Gibson assembly using the NEBuilder HiFi DNA Polymerase Kit from New England Biolabs (Ipswich, Massachussets, MA). All plasmids were sequenced at the GATC Biotech (Konstanz, Germany). We performed all cloning and transformations as per standard protocols. Antibiotics were used at the following concentrations: ampicillin (Ap), $50\,\mu g \cdot mL^{-1}$; chloramphenicol (Cm), $25\,\mu g \cdot mL^{-1}$; kanamycin (Km), $30\,\mu g \cdot mL^{-1}$; and spectinomycin (Sp), $50\,\mu g \cdot mL^{-1}$.

7.3.2 Pinocembrin sensor library construction

Sixteen pinocembrin biosensors were constructed by varying the plasmid copy number and the RBS strength.
First, primers P1 and P2 were used to amplify the plasmid backbones with different copy numbers from pACYCDuet-1, pCDFDuet-1, pETDuet-1, and pRSFDuet-1 (Supporting Information Table 7.7). Second, the RFP under the control of the responsive promoter to pinocembrin was amplified from the plasmid pV20 (Supporting Information Table 7.5) using the primers P3 and P4. Third, the FdeR TF with its constitutive promoter J23100 was amplified also from the plasmid pV20 with the four couples of primers P5/P9, P6/P9, P7/P9, and P8/P9 to generate the FdeR fragment with an RBS sequence 1, 2, 3, and 4, respectively. Finally, the 16 possible combinations were assembled in one step by Gibson assembly and confirmed by colonies PCR and sequencing (Figure 7.2). All the constructs of pinocembrin biosensors are highlighted in Supporting Information Table 7.8.

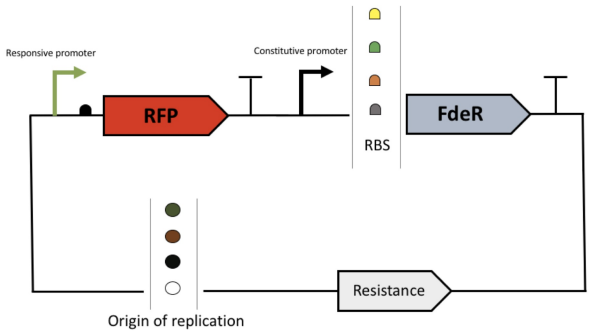


Figure 7.2 **Schematic representation of the pinocembrin biosensor module:** A promoter and a RBS precede each gene. A terminator is located downstream of each gene. Resistance refers to chloramphenicol resistance, spectinomycin resistance, ampicillin resistance, or kanamycin resistance. RFP

7.3.3 Biosensor dose-response characterization

For each biosensor strain, an isolated colony of BL21(DE3) harboring the appropriate plasmid was inoculated in 2 mL Luria Broth (LB) media containing the appropriate antibiotics and grown overnight at 37 °C. The culture was then diluted 1:100 in fresh LB containing the appropriate antibiotics as well as different concentrations of pinocembrin, naringenin, or cinnamic acid (previously dissolved in ethanol) ranging from 1 to 500 µM. All the sensor cells were grown then for 24 h with agitation at 37 °C in microplate reader BioTek. Absorbance at 600 nm and fluorescence (Exc: 580 nm /Em: 610 nm) were measured. All experiments were repeated at least three times.

7.3.4 Chemical structure similarity

Compound InChI was obtained from PubChem [387]. Chemical structure analysis was performed using the KNIME [223] analytics platform and RDKit nodes [194]. Tanimoto scores were computed using MACCS keys fingerprints [120].

7.3.5 Data normalization

RFP fluorescence reading was normalized by optical density (OD) to obtain values that are proportional to per cell fluorescence. For fold change data, values obtained with inducers were divided by values obtained without inducers:

$$fold_{change}(inducer) = \frac{RFP/OD(inducer)}{RFP/OD(inducer = 0)} \tag{7.1}$$

7.3.6 Simulation tools

All data analyses and simulations were run on R (version 3.2.3, [388]). Time evolution curves were simulated using the DeSolve package (version 1.14, [389]) and the rk4 algorithm, implementing the fourth-order Runge-Kutta method. For random parameter sampling around the best fit, values were sampled from within ±1.96 standard deviation of the parameter estimate.

7.3.7 Parameter fitting

All parameters that could be found in the literature are highlighted in Table 7.1. The other parameters (n, K_m, and Kd_{single}) were fitted using the nls (nonlinear square, from Package stats version 3.2.3) function using weighted least squares and the port algorithm [390], which allows for boundaries on the search space. The time evolution parameters (k_{deg} and α) were fitted using the optim function (from Package stats version 3.2.3, using the L-BFGS-B method implementing the Limited-Memory Broyden Fletcher Goldfarb Shanno Algorithm, which is a quasi-Newton method). Model function parameters were fitted locally to each data set of n = 3 replicates per data point unless otherwise stated. The final parameters used in the model are presented in Table 7.1 and Supporting Information Table 7.2.

Parameter name	Parameter value	Parameter description	Method of acquisition
n_{copy} for 157	10	Copy number for the 157 construct	Novagen (Supplier)
n_{copy} for 257	20	Copy number for the 257 construct	Novagen (Supplier)
n_{copy} for 357	40	Copy number for the 357 construct	Novagen (Supplier)
n_{copy} for 457	100	Copy number for the 457 construct	Novagen (Supplier)
n	1.847 ± 0.168 (dimensionless)	Cooperativity constant of the Hill model	Fitted on pinocembrin data
ratio for 157	0.142 ± 0.015 (AU)	Dynamic range of the construct divided by its copy number	Fitted on pinocembrin data
ratio for 257	0.707 ± 0.035 (AU)	Dynamic range of the construct divided by its copy number	Fitted on pinocembrin data
ratio for 357	1.767 ± 0.022 (AU)	Dynamic range of the construct divided by its copy number	Fitted on pinocembrin data
ratio for 457	0.417 ± 0.011 (AU)	Dynamic range of the construct divided by its copy number	Fitted on pinocembrin data
Correcting factor for naringenin	1.3 (dimensionless)	Correcting fold change factor	Estimated by averaging the correcting factors for individual constructs
$K_{dsingle}$	887.84 ± 120.68 (µM)	The Hill constant, K_d, for a single plasmid	Fitted on pinocembrin data
K_m for pinocembrin	1 (dimensionless)	Ratio between the binding constants of the inducer and the transcription factor	By definition
K_m for naringenin	2.142 ± 0.206 (dimensionless)	Ratio between the binding constants of the inducer and the transcription factor	Fitted on naringenin data
n_{TF}	2 (dimensionless)	The transcription factor forms dimers	Naringenin dose-response [385, 386]

Table 7.1 **Pinocembrin and naringenin parameters**: Parameters, their values, and references

Parameter name	Parameter value	Parameter description	Method of acquisition
OD_0	0.099	Starting OD	From data (100 µM of pinocembrin in the construct 357)
OD_m	1.0026 ± 0.0021	Maximum capacity	Fitted to 100 µM of pinocembrin in the construct 357
k	$0.0001743 \pm 1.342e-06$ s^{-1}	Exponential growth constant of logistic model	Fitted to 100 µM of pinocembrin in the construct 357
k_{deg}	$3.205e-05$ s^{-1}	Degradation constant	Fitted to 100 µM of pinocembrin in the construct 357
α	0.0019373 (Arbitrary Units (AU))	Term accounting for production and fluorescence	Fitted to 100 µM of pinocembrin in the construct 357

Table 7.2 Parameters, their value and references for the time-course model. Supplementary in the original article.

7.3.8 Sensitivity, fold change, and cooperativity of the different biosensors

To characterize the different biosensor dose-response curves, they were fitted to the following standard Hill function [155]:

$$fold_{change}(inducer) = \frac{RFP/OD(inducer)}{RFP/OD(inducer = 0)} \tag{7.2}$$

where I is the concentration of the considered inducer (in µM; K_d is the concentration that allows for half-maximum induction (in µM as well), also termed IC50; n is the Hill coefficient that characterizes the cooperativity of the induction system; and *ratio* is the dynamic range (in arbitrary units).

7.4 Results

7.4.1 Choice of the TF

Recently, Raman and colleagues were able to convert the intracellular presence of some flavonoids into a fitness advantage for the cell by combining the TtgR-responsive domain (a regulatory gene of the multidrug efflux pump operon, ttgABC) to a TolC membrane protein (an *E. coli* outer membrane protein) necessary for

survival under selective conditions. The strategy was successful in the screening of targeted genome-wide mutagenesis for naringenin high-producing strains [391]. It is very useful in evolution experiments looking to enrich the culture with evolved variants and counter-select the false positives but is not a first-choice strategy when planning to screen libraries and pinpoint the response of every single clone. Our objective is to combine a TF with a fluorescent response to sense pinocembrin, which has not been previously reported. The use of Sensipath webserver [77] has shown the need to transform pinocembrin to succinate or S-adenosyl-L-homocysteine to be sensed by a transcription regulator. This would not be relevant from a metabolic engineering point of view, where the main objective is to increase the titer of the final product and not to consume it in some other auxiliary reactions even for a screening purpose. Direct detection in this case is more valuable. In a previous work, [386] have already characterized FdeR and Qdor, two TFs from *H. seropedicae* SmR1 and *Bacillus subtilis* shown to be responsive in *E. coli* to naringenin and kaempferol, respectively. Since these two compounds belong like pinocembrin to the flavonoid group, we have therefore performed a chemical structure similarity search in this family of chemicals. We have shown using the Tanimoto score that naringenin is the closest detectable compound to pinocembrin (see Section 7.3.4). We then decided to use FdeR as a potential candidate to develop a pinocembrin biosensor (Table 7.3).

7.4.2 Biosensor design and construction

[385] have identified the *fde* operon, associated with the degradation of aromatic compounds, mainly naringenin. The expression of this operon, under the regulation of the FdeR TF, is induced by naringenin. Thus, we have built a plasmid containing FdeR under a constitutive promoter and an RFP under the control of the responsive promoter from the fde operon. To build our combinatorial library to identify the best biosensors, we chose to build the constructs using four different plasmid copy numbers and four different RBS sequences for the FdeR gene (Figure 7.2).

7.4.3 Biosensor characterization

To benchmark our design, *E. coli* cells harboring the different constructs were grown for 24 h in the absence and presence of increasing concentrations of pinocembrin or naringenin ranging from 1 to 500 µM, and red fluorescence was monitored in parallel with cell growth (Figure 7.3a). As expected, the different biosensor constructs were active in *E. coli* in the presence of naringenin. More interestingly, the different constructs were able to detect pinocembrin, and most of them have shown a high expression level of RFP exceeding in all cases the level of expression in the presence of naringenin. Moreover, FdeR appears to be more sensitive to pinocembrin than naringenin, as is evident from the steeper slope in Figure 7.3a. The results have shown that the minimal concentration of pinocembrin required to activate the TF ranges between 1 and 5 µM. The fold change is also shown to reach 60 folds in construct 156 for instance. In some cases, we highlighted a decrease in the fluorescence

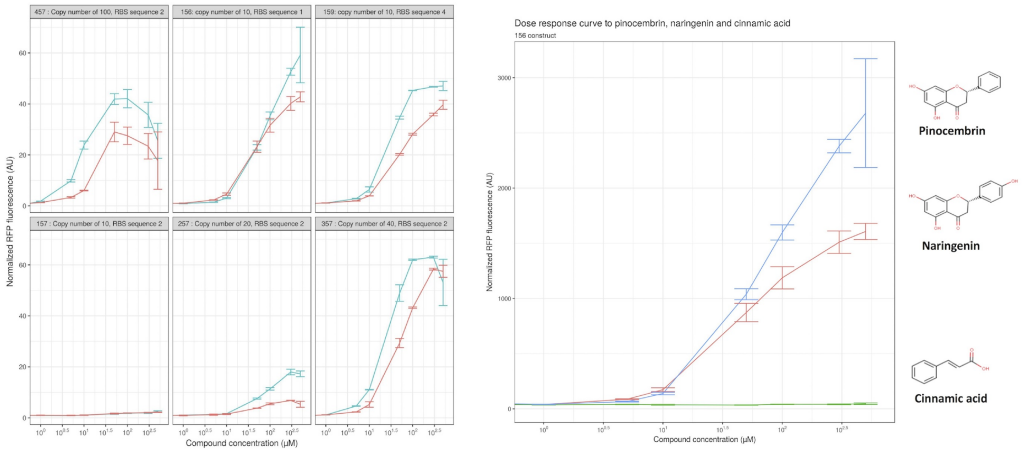
when we exceed 300 µM, which is probably due to the toxicity of the compound. This toxicity could also explain the difficulty in reaching high titer of pinocembrin in metabolic engineering experiments, where, as mentioned previously, the record is around $40\,\mathrm{mg} \cdot \mathrm{L}^{-1}$ [384].
To validate this biosensor as a potential candidate for screening purposes, we tried to evaluate the specificity of FdeR. The sensor detects pinocembrin, but what about its biosynthesis intermediates? The work of [385] has shown that this TF is not activated by phenylalanine or tyrosine, which are the precursors of pinocembrin and naringenin, respectively. Next, we investigated the effect of cinnamic acid, a key intermediate in the pinocembrin pathway. One of the most sensitive constructs (156, see list of constructs in the Supporting Information Table 7.8) was grown in the presence of increasing concentrations of cinnamic acid. The results show no detection of this compound (Figure 7.3b). As a conclusion, none of the major intermediates are detectable by FdeR. These data support our choice of using the FdeR biosensor as a tool to screen for pinocembrin- or naringenin-producing cells.

7.4.4 Choosing an adapted modeling strategy

The FdeR TF has been studied in only a few previous publications [385, 386], which means although it is characterized enough to know which inducers will or might bind to it and induce a response in *E. coli*, there is no quantitative data available on the binding strengths of the inducers to the TF or of the complex to the promoter. We had the choice among three main modeling approaches: statistical physics model [392, 393], mechanistic modeling [143], or variations on Hill modeling [153, 158].
The statistical modeling approach makes use of extensive knowledge of the promoter, its inducer, and the TF. For instance, after reviewing several published works, [392] have highlighted the need of the following constants to model a transcriptional activator: different binding energies (RNA polymerase to the promoter, TF to the promoter, binding interaction between the two, RNA polymerase to the rest of the genome) as well as knowledge of the number of binding sites on the promoter or the number of promoters. This does not include yet the effect of the binding of the inducer to the TF or of the genetic context (e.g., if there is a DNA binding loop for repression). This kind of modeling has been applied to the Lac operon but remains elusive for less-characterized systems such as our novel biosensor.
The mechanistic approach models all possible interactions in the system, or at least most, with an important number of parameters ([143]; 21 for the ArsR biosensor). This approach, although interesting, necessitates a lot of biological knowledge to minimize the number of unknown parameters, as well as knowledge of the interactions that do occur or not. This can therefore only be carried out in a relatively well-known system. Those parameters are then fitted using system biology approaches and different optimization algorithms to avoid the main issue these models face: their sloppiness. Sloppiness characterizes the fact that different sets of parameters can model the data due to high interdependency between parameters. For example, when two parameters are used to model a forward and a backward

reaction, which is actually at equilibrium given the time scale considered, an infinite number of parameters, whose ratio is the equilibrium constant of the reaction, will fit the data.
The Hill class of models does not necessitate a priori knowledge of the exact interactions between the species involved, although knowledge of the broad behavior of the interactions is necessary. This model has been, for example, extended to take into account resource competition [153], model both the binding with the inducer and complex binding to the promoter in the Lux system [158] or any switch-like behavior. Therefore, we decided to extend the Hill model to account for a key tunable parameter in synthetic biology: plasmid copy number. Our aim was to have a model with as little free parameters as possible that could account for this effect.

7.4.5 Effects of plasmid copy number that we intend to model



(a) **(b)**

Figure 7.3 **Dose responses of different biosensor constructs. a** Constructs 457, 156, 159, 157, 257, and 357 were cultivated for 24 h in the presence of increasing concentrations of pinocembrin (blue) and naringenin (red) ranging from 1 to 500 µM. Error bars are based on the standard deviation of a minimum of biological triplicate. **b** Biosensor 156 was cultivated for 24 h in the presence of increasing concentrations of pinocembrin (blue), naringenin (red), and cinnamic acid (green) ranging from 1 to 500 µM. Error bars are based on the standard deviation of a minimum of biological triplicate

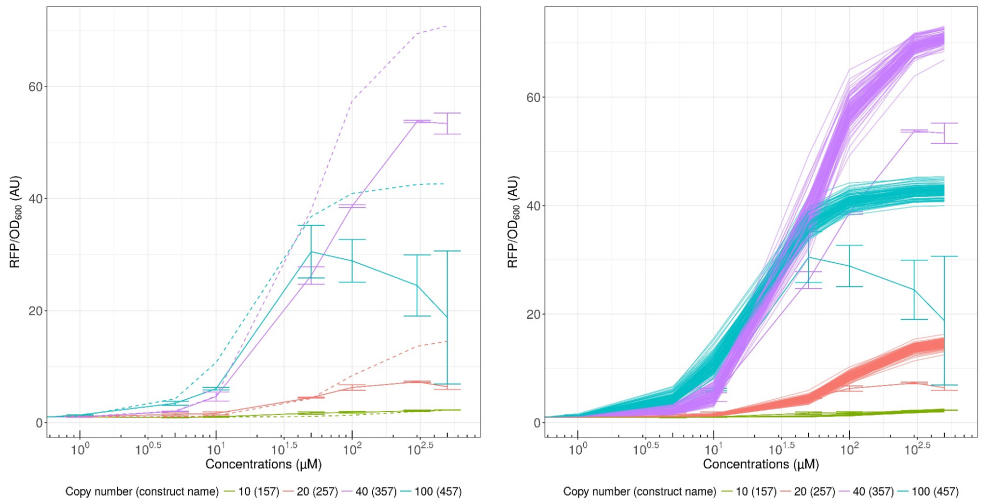


Figure 7.4 **Effect of copy number variation on data fold change for pinocembrin a and naringenin b.** In colors are represented the different constructs. The assumed copy numbers are as follows: 10 for 157, 20 for 257, 40 for 357 and 100 for 457. Concentration of the inducer are expressed in µM and vary from 0 to 500. Error bars represent standard deviation.
Supplementary Table in original article.

As can be seen in Figure 7.3a (or the Supporting Information Figure 7.4), increasing the copy number leads to increased production, as expected, except for construct 457 (very high copy number), showing a decline in production after 100 µM concentration of pinocembrin or naringenin. The constructs behave similarly for both compounds, although the biosensor is slightly more effective for pinocembrin detection than for naringenin detection, which is somewhat unexpected given that naringenin is its natural reported activator. Another interesting aspect is the effect of copy number on IC_{50} (concentration at which the biosensor reaches half-maximum induction: it corresponds to K_d in the standard Hill function). We can see that effect both in the figure where the induction starts at smaller concentrations of the inducer and in biosensor characterization (Supporting Information Table 7.4), where IC_{50} diminishes with copy number of the construct. We therefore decided to take that effect into account in our modeling effort.

Compound	**Plasmid construct**	**n**	**ratio**	IC_{50} **(or** K_d**)**
pinocembrin	156	1.63 ± 0.12	56.62 ± 2.24	71.02 ± 5.57
naringenin	156	1.49 ± 0.04	42.52 ± 0.64	49.30 ± 2.18
pinocembrin	157	1.89 ± 1.26	1.27 ± 0.24	65.87 ± 24.16
naringenin	157	1.77 ± 0.34	1.14 ± 0.05	43.47 ± 5.93
pinocembrin	159	2.04 ± 0.18	46.95 ± 0.82	27.97 ± 2.45
naringenin	159	1.51 ± 0.06	37.66 ± 0.90	50.63 ± 2.71
pinocembrin	257	1.56 ± 0.12	17.70 ± 0.80	73.68 ± 6.41
naringenin	257	1.68 ± 0.45	6.02 ± 0.37	53.25 ± 5.86
pinocembrin	357	1.96 ± 0.20	63.24 ± 1.48	23.07 ± 2.38
naringenin	357	1.51 ± 0.07	61.57 ± 1.09	58.82 ± 2.51
pinocembrin	457	1.92 ± 0.17	42.37 ± 1.38	9.62 ± 0.66
naringenin	457	1.91 ± 0.57	30.17 ± 4.59	22.53 ± 7.62

Table 7.4 **Biosensors characteristics**. Supplementary Table in original article

7.4.6 Derivation of the dose-response model: Accounting for copy number

We aim to show here a dose-response model that can account for the effect of copy number on both pinocembrin- and naringenin-responding constructs. We need to take into account two effects of plasmid copy number.

1. The number of binding sites for the TF-inducer complex increases proportionally to the plasmid copy number, meaning that intuitively, to reach half-maximum saturation, there needs to be that many more TF-inducer complexes

2. The TF is produced constitutively from the biosensor plasmid, so TF number scales with plasmid copy number.

We consider that all following processes are at equilibrium since chemical binding is a fast process compared with transcription and translation, and we are considering dose-response curves for the time being.

Formation of the TF-inducer complex

We consider that the TF forms n_{TF} multimers to derive our equations. According to the literature, FdeR forms dimers [386], which means $n_{TF} = 2$ will be used when simulating the data. Since the exact binding configuration with the induc-

ers (naringenin and pinocembrin) is not known, we will start by considering the following equilibrium (Equation 7.3). Other neglected cooperativity effects will be accounted for in the Hill cooperativity constant (Equations 7.4, 7.5 and 7.6):

$$n_{TF}T_F + I \underset{K_{dis}}{\rightleftharpoons} T_F^c \tag{7.3}$$

Ignoring the order of binding, which is not important for the final equilibrium but only for the kinetics, not considered here, given the time scales of the considered processes, we have Equation 7.4, where T_F is the concentration of the TF, T_F^c, of the TF complexes and K_{dis} is the dissociation constant of the complex:

$$T_F^c = \frac{I \times T_F^{n_{TF}}}{K_{dis}} \tag{7.4}$$

Note that K_{dis} depends on the inducer considered here, either pinocembrin or naringenin, and it is the dissociation constant of the considered reaction. We first consider a classical Hill binding equation for the induction due to the TF before improving on this equation. The classical equation is Equation 7.5, where n is the cooperativity constant and *ratio* represents the maximum induction or the dynamic range. K_d is the TF-inducer complex concentration needed for half-maximum induction:

$$P_{fold} = \frac{(T_F^c)^n}{(K_d)^n + (T_F^c)^n} \times ratio \times n_{copy} \tag{7.5}$$

However, we want to consider the fact that the plasmid copy number changes the number of binding sites for the TF (proportional to the number of plasmids in our construct, as there might be cooperativity and therefore more than one binding site per plasmid). We propose the following modification to Equation 7.5, which accounts for the fact that to reach half-maximum saturation of a higher number of binding sites, the number of binding complexes also needs to be that much higher:

$$P_{fold} = \frac{(T_F^c)^n}{(K_d \times n_{copy})^n + (T_F^c)^n} \times ratio \times n_{copy} \tag{7.6}$$

When replacing Equation 7.4 into Equation 7.6, we obtain:

$$P_{fold} = \frac{(I)^n}{(\frac{K_d \times K_{dis} \times n_{copy}}{T_F^{n_{TF}}})^n + (I)^n} \times ratio \times n_{copy} \tag{7.7}$$

Let $K_m = K_{dis}(\text{compound})/K_{dis}(\text{pinocembrin})$ in Equation 7.8 be the ratio between the dissociation constant of the compound of interest divided by the one for pinocembrin, where the dissociation constant in itself is unknown. Therefore, $K_m = 1$ for pinocembrin and $K_m = K_{dis}(\text{naringenin})/K_{dis}(\text{pinocembrin})$ for naringenin. Introducing this parameter allows us to only consider the difference of binding strength between FdeR and naringenin or pinocembrin instead of the absolute binding values, which would add one sloppy parameter to our model. Moreover, since the TF is produced under a constitutive promoter on the plasmid, we can assume it is produced proportionally to the plasmid copy number. The proportionality constant is included into the $K_{dsingle}$ constant, as well as K_{dis} (pinocembrin), leading to Equation 7.8. We can note here that given our hypothesis (number of TFs and binding sites scaling with the copy number), no effect would be obtained in our model if FdeR were not a dimer:

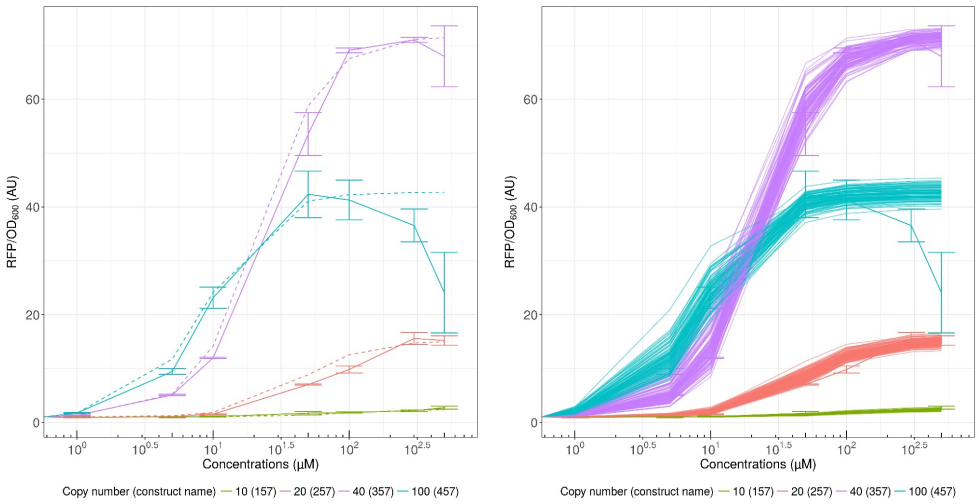
$$P_{fold} = \frac{(I)^n}{(K_{dsingle} \times K_m \times n_{copy}^{1-n_{TF}})^n + (I)^n} \times ratio \times n_{copy} \tag{7.8}$$

7.4.7 Analysis of the dose-response model

Model fitting of pinocembrin

The model was fitted to the data according to the procedure presented in materials and methods. The fitted parameters were $K_{dsingle}$, n and $ratio$, as by definition $K_m = 1$ for pinocembrin. The obtained parameters are listed in Table 7.1. Since we intend to model the effect of copy number variations, we chose to use constructions sharing the same RBS sequence for our parameter fitting: 157, 257, 357, and 457.

We chose to represent both the best fit (Figure 7.5a) and 100 simulations (Figure 7.5b), where parameters were randomly sampled from the estimated distribution of parameters (see materials and methods for more details). We can see when looking at the random parameters that there is some leeway in the estimation, allowing for a rather wide dose-response curve. However, the expected behavior is maintained, even accounting for uncertainty in the estimation of the parameters. We chose to use the same cooperativity constant n, as well as the same $K_{dsingle}$, constant, which would be the IC_{50} for a single plasmid, and hence its name. However, as mentioned in the data analysis section, the dynamic range does not scale proportionally with the plasmid copy number. For this reason, ratios varying from 0.14 to 1.76 were obtained and used in this study, instead of using a single parameter for this effect. This is due to a host of factors: higher plasmid copy number diverts more resources from the cell, the replication machinery is not the same for the different plasmids, which have different replication origins, and the cells do not divert resources to plasmids proportionally to their copies. Moreover, an interesting feature of the data is that production from the very high copy number construct (457) is initially higher than with the high copy number (357) until concentrations cross a threshold.

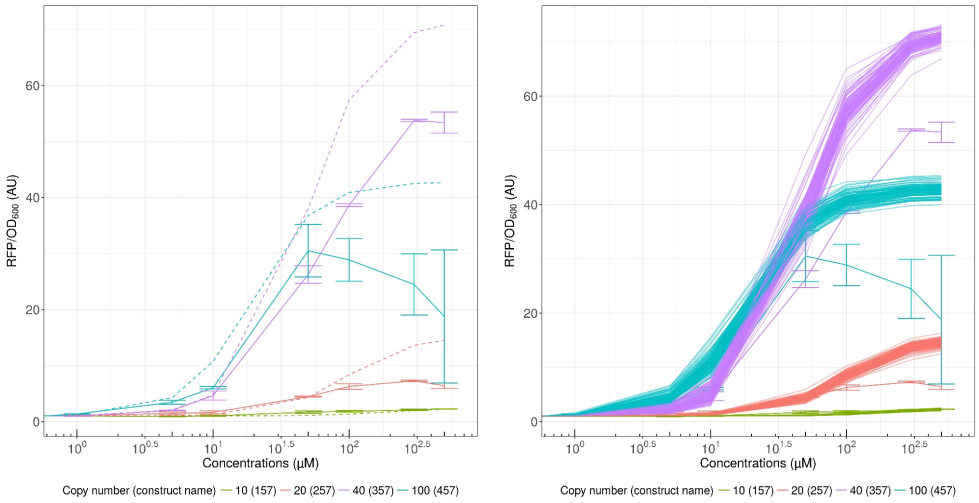


(a) (b)

Figure 7.5 **Model fitting to pinocembrin data for varying copy numbers.** **a** Best fit parameters for pinocembrin. **b** 100 random simulations from parameter fitting for pinocembrin. Error bars represent standard deviation

We can imagine that the demand on the cell from our constructs becomes too high in the 457 construct, and the cell activates a "stress response". This is observed when using both compounds for induction.

Model fitting of naringenin



(a) (b)

Figure 7.6 **Model fitting to naringenin data for varying copy number without correcting parameter.** **a** Best fit parameters for naringenin. **b** 100 random simulations from parameters fitting for naringenin. Error bars represent standard deviation.
Supplementary Figure in original article.

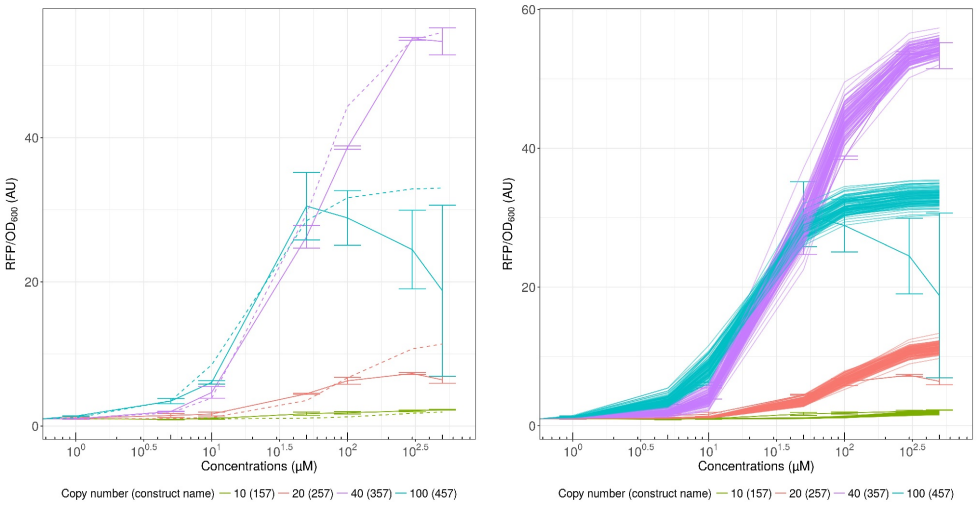


Figure 7.7 **Model fitting to naringenin data for varying copy numbers.** **a** Best fit parameters for naringenin. **b** 100 random simulations from parameter fitting for naringenin. Error bars represent standard deviation

We were interested to determine whether the model could reproduce the features observed in the naringenin data: globally lower fold change of induction than for the pinocembrin induction, but the same overall behavior on sensitivity. Our aim was to account for the compound change using only our K_m parameter, which represents the ratio between the dissociation constants of inducers to the TF (K_m $=K_{dis}$(naringenin)$/K_{dis}$ (pinocembrin)). The results of this modeling strategy, when fitting only K_m, give the results presented in Supporting Information Figure 7.6. However, as mentioned in Section 7.4.7, we chose to model fold change variation with a single parameter (multiplied by the copy number). Therefore, our model can capture changes in sensitivity due to both copy number increases and compound changes, but since this is included in our Hill function, variations of fold change at saturating amounts of substrates cannot be captured. Therefore, we added a correcting factor for all naringenin models, reducing all ratio values in our models by 1.3. This factor was chosen as a weighted average of correcting factors for the different constructs. The results obtained by this strategy can be seen in Figure 7.7.

The global behavior of the biosensor is respected for all sensors, meaning that the same model does apply to this data. The shift in dose response can be explained by the K_m parameter, which shifts the curve toward less sensitivity by doubling IC_{50}. This is confirmed by the data for 357 and 457 constructs, which are the constructs with the least variability on IC_{50} estimation. We can also observe that the dynamic range is slightly lower, meaning that our modification of the ratio parameter by the same correcting factor is justified (for naringenin concentrations up to 100 µM). This could be explained by an effect that is not taken into account in our model, such as higher load, or some different toxicity between pinocembrin and naringenin. K_m(naringenin) is bigger than one, which means that the dissociation of FdeR dimer with naringenin is higher than the one with pinocembrin. In

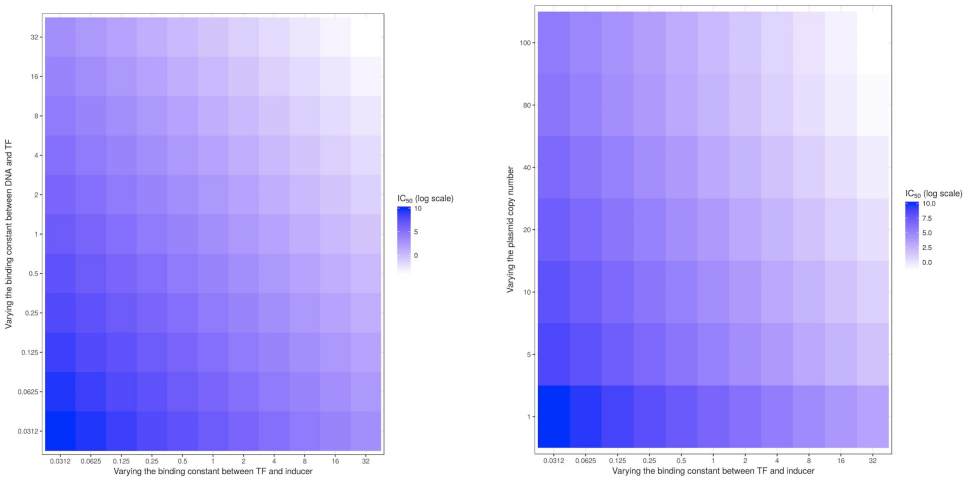
other words, at the same TF and inducer concentration, there is more TF bound with pinocembrin than would be with naringenin. This is surprising given the fact that FdeR was identified in the fde operon from *Herbaspirillum seropedicae*, which is involved and was identified for its implication in naringenin degradation. This means that we expected it to be evolved for naringenin detection, but that it detects pinocembrin at least as well.
All this indicates that our model, although very simple and based on broad knowledge of the sensor rather than precise chemical constant values, manages to successfully capture our system′s behavior.

7.4.8 Time-course model assumptions and derivation

Once we had a satisfying dose-response model, we chose to model the time-dependent response of our biosensor, to determine the delay between the signal and the fluorescence production. We considered a relatively simple time-course model, consisting of a production term and a degradation term for the protein. Results are presented in the Supplementary. This time-course modeling partially allowed us to understand the impact of initial dilution on the biosensor′s behavior and emphasized the need to wait for it to reach steady state for it to be fully functional and decipher between different inducer concentrations. The shortcomings of this time-course modeling confirm that although it is interesting to see the delay in response of the biosensor signal, modeling the dose-response curve is more important to show characteristics of the biosensor, such as changes to the dose-response curve when used for screening pinocembrin-producing strains.

7.4.9 Leveraging our model for biosensor design improvement

Having constructed a satisfying dose-response model, it becomes interesting to use it to make predictions for future improvements of our design. We therefore considered three parameters that synthetic biologists can tune and study their effect on half-maximum induction (IC_{50}), used as a proxy for sensitivity. A higher IC_{50} means shifting the sensitivity of the biosensor toward higher concentrations and therefore can be used to screen higher producing strains. A lower IC_{50} means shifting it toward lower concentrations and more sensitivity to trace amounts of pinocembrin. The three parameters whose effects we decided to study are the following: plasmid copy number, DNA and TF binding strength, and TF and inducer binding strength. Plasmid copy number can easily be tuned by choosing the replication origin of the plasmid, DNA-TF affinity can be modified either by random mutagenesis of the promoter or by protein engineering (and measured through gel retardation assays), and TF–inducer affinity can be tuned by protein engineering. In Figure 7.8, we represent fold change compared with current fitted constants for TF and inducer binding strength. DNA and TF dissociation constant being captured by our Hill equation, it is proportional to our $K_{dsingle}$ constant, so the



(a) **(b)**

Figure 7.8 **Copy number model predictions**: Effect on biosensor sensitivity of varying copy numbers, DNA, and TF binding affinities or transcription factor and inducer binding affinities. Half-maximum induction (IC_{50}), used as a proxy for sensitivity, is represented in colors ranging from white (low IC_{50}, high sensitivity) to dark blue (high IC_{50}, low sensitivity) on a log scale. Binding constants are represented as fold-change compared with current fitted constants. **a** Comparison of the effect of changing TF and DNA binding constants and TF and inducer binding constant. **b** Comparison of the effect of changing plasmid copy number and TF and inducer binding constant

binding strength is proportional to the inverse of $K_{dsingle}$, and we are also representing fold changes around this constant. The copy number, on the other end, is represented as the desired value for copy number, as that can be achieved by choosing a correct replication origin to achieve the desired copy number. We can see in Figure 7.8a that increasing the binding constant between TF and DNA or TF and the inducer has similar consequences: increasing it leads to lower IC_{50} or higher sensitivity, whereas decreasing it leads to higher IC_{50}, allowing one to detect higher titers of pinocembrin. This suggests that random mutagenesis at the promoter might be a better first approach to tune the biosensor′s behavior to an experimentalist′s needs, since it is easier to engineer rather than engineering the binding strength of the TF and its inducer, and both have similar consequences. Figure 7.8b, on the contrary, shows the impact of changing plasmid copy number or binding affinity of the TF for the inducer. As seen in our experimental data, increasing the copy number (which leads to higher expression) also increases sensitivity, allowing for better detection of the inducer but at lower concentrations. Reducing the copy number enables detection at higher titers, but reduces the fold change of the biosensor. On the contrary, augmenting the affinity of the TF to the inducer boosts sensitivity but does not allow differentiating different responses at high concentrations of the inducer. Therefore, our model suggests possibilities to further engineer our system, whether to sense high titers of pinocembrin to increase the biosensor′s sensitivity.

7.5 Discussion

The use of TF-based biosensors is expanding in many fields, ranging from environmental, biomedical to industrial biotechnology applications and more specifically as a fast and reliable screening tool to address the problems of high-throughput limits of the other approaches [394, 395]. Some successful attempts have been reported describing strategies leading to the fine-tuned response dynamics and dynamic ranges by engineering tunable biosensors [396, 397]. TFs have a ligand-binding domain most likely to be promiscuous. In this study, we showcased the potential of chemical structure similarity scoring to select TF starting candidates to develop or engineer biosensors for small molecules. We have constructed a biosensor to detect pinocembrin with a fold change of around 60. FdeR appears unexpectedly to be more sensitive to pinocembrin than to naringenin, its natural effector, and has the required specificity to discriminate against the intermediates in the pinocembrin biosynthetic pathway. Indeed, the first report of this TF in [385] identifies FdeR as the TF responsible for the regulation of a naringenin degradation operon. However, our experiments prove that FdeR senses pinocembrin at least as well, suggesting that this operon could also be involved in pinocembrin degradation. Two possible degradation pathways were identified by [385] based on *in silico* analysis of the enzymes found in the operon. One started by opening the C-ring of naringenin, whereas the other opened the A-ring. In both cases, since pinocembrin differs from naringenin by a group on the B-ring, it could also be degraded by these pathways. A recent study performed by [330] was also successful in generating a new biosensor for specific lactam compounds using a chemoinformatics approach inspired by small-molecule drug discovery. Methodologies based on the structural analysis of compounds could offer an alternative to some heavy strategies based on the design of new TFs for non-natural ligands [398, 399, 400] or by random mutagenesis [401, 402, 403].

To extend our knowledge of the rules governing the sensitivity, specificity, and dose responses of biosensors, we have also built different sensor constructs varying the copy number and the RBS to scan different response patterns that could serve as a template for modeling and to help extract rational understanding of the biosensor behavior.

Although simple, the model developed in this paper allows us to explain the behavior of our biosensor to both naringenin and pinocembrin with a single parameter that accounts for the binding variability between these two compounds and the TF. It also accounts for variations of copy number on the sensitivity of the biosensor starting from a simple idea: if there are more binding sites, there is a need for proportionally more activators to reach half-maximum saturation. This is a simple but useful addition to the synthetic biology modeler′s toolbox when working on poorly characterized systems where more robust modeling approaches, such as mechanistic or statistical modeling, are not possible to use. Our model allows us to not only describe trends but also quantitatively correct values.

An interesting effect we managed to capture is the effect of copy number on IC_{50}. This effect was already observed in a previous work of [158] although the authors did not investigate the link between copy number and IC_{50}. Although they have

an IC_{50} that increases with copy number (although the relationship is not linear), the way they model their binding renders a numerical comparison impossible.
In the present paper, we have two different effects when increasing copy number: we increase the number of binding sites (increasing IC_{50}) but we also increase the number of available TFs, allowing for more binding even with less inducer, thereby reducing the IC_{50}. According to our model, if the TF concentration was not increasing, we would also see a reduced sensitivity, as found in [158], which confirms our biosensor design idea.
As we have seen, the time evolution model is not fully satisfying. A few strategies could help make it closer to the data, but they all present the disadvantage of adding new free parameters: adding a lag time for protein production as the introduced dilution does not seem to be enough and add some toxicity or load effect when copy number, TFs, and inducers are in too great numbers. These were not implemented as our aim was to present a model with a minimal set of parameters that explained the data well enough.
Another interesting feature of our model is to suggest further modifications of our design depending on the desired application: increasing its sensitivity, its dynamic range, or being able to sense higher titers of pinocembrin, by capturing the effects of changing copy number, DNA-TF binding affinity, or TF-inducer binding affinity.
As a conclusion, we have presented a simple model with a minimal number of parameters that allows us to capture the effects of both copy number and inducer variations on our biosensors′ behaviors and most notably on sensitivity, which are effects that have not been addressed as such and especially never with such a simple formalism. This model, based on a simple Hill equation, has the advantage of being very versatile and easy to use on previously uncharacterized systems.
The development of the pinocembrin biosensor, its modeling, and understanding its behavior open doors to generate more transcription-factor-based biosensors to meet the increasing demands of screening and dynamically regulating metabolic pathways in industrial strains.

Name	InChI	Tanimoto
Luteolin	$1S/C15H10O6/c16-8-4-11(19)15-12(20)6-13(21-14(15)5-8)7-1-2-9(17)10(18)3-7/h1-6,16-19H$	0.8125
Apigenin	$1S/C15H10O5/c16-9-3-1-8(2-4-9)13-7-12(19)15-11(18)5-10(17)6-14(15)20-13/h1-7,16-18H$	0.8965
Genkwanin	$1S/C16H12O5/c1-20-11-6-12(18)16-13(19)8-14(21-15(16)7-11)9-2-4-10(17)5-3-9/h2-8,17-18H,1H3$	0.7812
Chrysin	$1S/C15H10O4/c16-10-6-11(17)15-12(18)8-13(19-14(15)7-10)9-4-2-1-3-5-9/h1-8,16-17H$	0.8965
Flavone	$1S/C15H10O2/c16-13-10-15(11-6-2-1-3-7-11)17-14-9-5-4-8-12(13)14/h1-10H$	0.7241
Quercetin	$1S/C15H10O7/c16-7-4-10(19)12-11(5-7)22-15(14(21)13(12)20)6-1-2-8(17)9(18)3-6/h1-5,16-19,21H$	0.8125
Fisetin	$1S/C15H10O6/c16-8-2-3-9-12(6-8)21-15(14(20)13(9)19)7-1-4-10(17)11(18)5-7/h1-6,16-18,20H$	0.7812
Kaempferol	$1S/C15H10O6/c16-8-3-1-7(2-4-8)15-14(20)13(19)12-10(18)5-9(17)6-11(12)21-15/h1-6,16-18,20H$	0.8387
Galengin	$1S/C15H10O5/c16-9-6-10(17)12-11(7-9)20-15(14(19)13(12)18)8-4-2-1-3-5-8/h1-7,16-17,19H$	0.8387
Kaempferid	$1S/C16H12O6/c1-21-10-4-2-8(3-5-10)16-15(20)14(19)13-11(18)6-9(17)7-12(13)22-16/h2-7,17-18,20H,1H3$	0.7647
Eriodictyol	$1S/C15H12O6/c16-8-4-11(19)15-12(20)6-13(21-14(15)5-8)7-1-2-9(17)10(18)3-7/h1-5,13,16-19H,6H2/t13-/m0/s1$	0.9062
Naringenin	$1S/C15H12O5/c16-9-3-1-8(2-4-9)13-7-12(19)15-11(18)5-10(17)6-14(15)20-13/h1-6,13,16-18H,7H2$	*0.9655*
Isosakurametin	$1S/C16H14O5/c1-20-11-4-2-9(3-5-11)14-8-13(19)16-12(18)6-10(17)7-15(16)21-14/h2-7,14,17-18H,8H2,1H3$	0.875
Flavanone	$1S/C15H12O2/c16-13-10-15(11-6-2-1-3-7-11)17-14-9-5-4-8-12(13)14/h1-9,15H,10H2$	0.7586
Pinocembrin	$1S/C15H12O4/c16-10-6-11(17)15-12(18)8-13(19-14(15)7-10)9-4-2-1-3-5-9/h1-7,13,16-17H,8H2/t13/m0/s1$	1

Table 7.3 **Flavonoids similarity to pinocembrin**: Tanimoto scores for flavonoid compounds

7.6 Supplementary Data

7.6.1 Tables for biological data

	Characteristics	**Source or Reference**
Strains		
E. coli BL21 (DE3)	C> B F^- ompT gal dcm lon $hsdS_{B(r_B^- m_B^-)}$ λ(DE3 [lacI lacUV5-T7p07 ind1 sam7 nin5]) [$malB^+$]$K-12$ (λ^S)	[225]
E. coli DH5 α	C> F^- endA1 glnV44 thi-1recA1relA1gyrA96deoRnupGpurB20ϕ80dlacZΔM15 Δ(lacZYA-argF)U169, hsdR17($r_K^- m_K^+$), λ^-	[225]
Mach 1	ΔrecA1398 endA1 tonA Φ 80ΔlacM15 ΔlacX74 hsdR($r_K^- m_K^+$)	Invitrogen Technology.
Plasmids		
pACYC	pACYC Duet / cat + / CmR / P15A rep /LacI	Novagen (EMD Millipore)
pCDF	pCDFDUet / aad+ / SpecR / CDF /T7 Prom / LacO / LacI	Novagen (EMD Millipore)
pCOLA	pCOLA Duet / kan+ / KanR / ColA / lacI	Novagen (EMD Millipore)
pET	pETDuet / bla / ApR / pBR322 / lacI	Novagen (EMD Millipore)
pV20	pSB1A3-FdeR-RFP/ Amp/ FdeR + responsive RFP	This study (Data not shown)

Table 7.5 **Strains and plasmids**

Primer	**5'->3' Sequence**
P1	ACGTCAGGTGGCGCTGACGTCGGTACC
P2	CAACAACGGAGCTCGACCGATGCCCTTGAG
P3	CTCAAGGGCATCGGTCGAGCTCCGTTGTTGTGCTTGTTC
P4	GCTAGCACTGTACCTAGGACTGAGCTAGCCGTCAACTGCAGGAAGACGCAACTAG
P5	CTAGCTCAGTCCTAGGTACAGTGCTAGCCGCCTTTAATATACAAATTTACGTACTCCACTATGCGTTTCAACAAGCTCGAC
P6	CTAGCTCAGTCCTAGGTACAGTGCTAGCTCCCTTTTTAAGCATAGATAAGTAGCCATATTATGCGTTTCAACAAGCTCGAC
P7	CTAGCTCAGTCCTAGGTACAGTGCTAGCACCCTATCTATATATAAGCCTCTAATTCATGCGTTTCAACAAGCTCGAC
P8	CTAGCTCAGTCCTAGGTACAGTGCTAGCACATTTTCACACCTCTCAAGGAGCACACTATGCGTTTCAACAAGCTCGAC
P9	GGTACCGACGTCAGCGCCACCTGACGTCTAAGAAAC

Table 7.6 **Primers list**

Plasmid name	Referenced copy number	Used copy number
PACYC-Duet	10	10
PCDF-Duet	20 – 40	20
PET-Duet	40	40
PRSF-Duet	> 100	100

Table 7.7 **Copy numbers of the used plasmids**

Construct name	Plasmid backbone	Origin of replication	RBS sequence	Resistance Cassette
156	PACYC-Duet	p15A	1 (designed in primer P5)	Chloramphenicol
157	PACYC-Duet	p15A	2 (designed in primer P6)	Chloramphenicol
158	PACYC-Duet	p15A	3 (designed in primer P7)	Chloramphenicol
159	PACYC-Duet	p15A	4 (designed in primer P8)	Chloramphenicol
256	PCDF-Duet	CDF	1 (designed in primer P5)	Spectomycin
257	PCDF-Duet	CDF	2 (designed in primer P6)	Spectomycin
258	PCDF-Duet	CDF	3 (designed in primer P7)	Spectomycin
259	PCDF-Duet	CDF	4 (designed in primer P8)	Spectomycin
356	PET-Duet	pBR322	1 (designed in primer P5)	Ampicillin
357	PET-Duet	pBR322	2 (designed in primer P6)	Ampicillin
358	PET-Duet	pBR322	3 (designed in primer P7)	Ampicillin
359	PET-Duet	pBR322	4 (designed in primer P8)	Ampicillin
456	PRSF-Duet	RSF	1 (designed in primer P5)	Kanamycin
457	PRSF-Duet	RSF	2 (designed in primer P6)	Kanamycin
458	PRSF-Duet	RSF	3 (designed in primer P7)	Kanamycin
459	PRSF-Duet	RSF	4 (designed in primer P8)	Kanamycin

Table 7.8 **Pinocembrin-sensor constructs list**

7.6.2 Time-course model assumptions and derivation

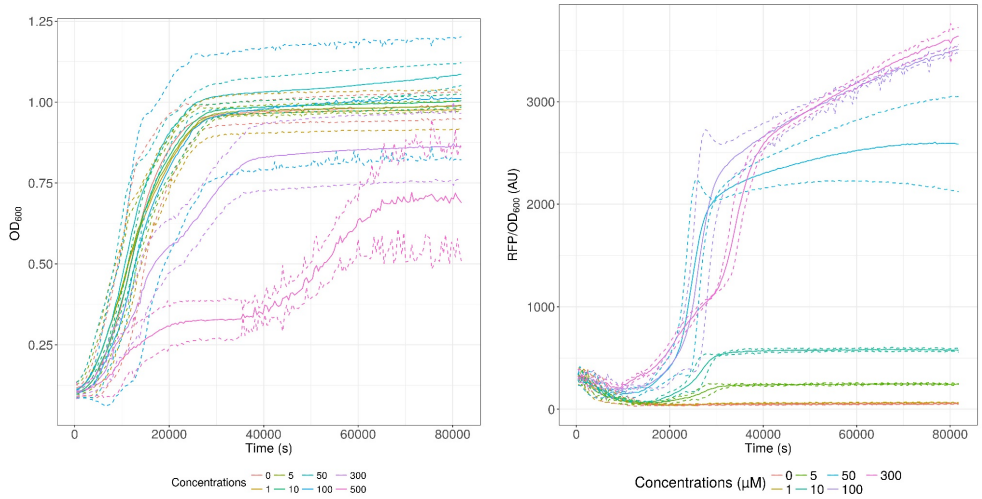
Once we had a satisfying dose-response model, we chose to model the time-dependent response of our biosensor, to see the delay between the signal and the fluorescence

production. We consider a relatively simple time-course model, consisting of a production term and a degradation term for the protein. messenger RNA (mRNA) is not explicitly taken into account here as models explicitly taking it into account do not provide better fits to the data but add unnecessary parameters (results not shown).

$$\frac{dRFP}{dt} = P(inducer, construct) - k(RFP) \tag{7.9}$$

where $P(inducer, construct)$ is a function that describes the production depending on the inducer concentration I and the chosen construct, and k is a term that encompasses both dilution and degradation processes. RFP here represents the RFP produced by an individual cell, therefore it corresponds to the normalized RFP data. *inducer* concentration is considered to be the external inducer concentration.

Modeling dilution and degradation:



(a) **(b)**

Figure 7.9 **Time course modeling of construct 357. a** OD data. **b** RFP divided by OD data. Solid lines represent mean of the experimental data and dashed lines 95% confidence intervals. Different colors represent varying concentrations as indicated on the Figure.

As can be seen in the time evolution of our constructs (Supplementary Figure 7.9) represents time evolution of the construct 357 for OD (a) and RFP/OD (b)), dilution plays an important role during the first hours, which correspond to exponential growth phase. When cells reach stationary phase (after 30 to 40 thousand seconds, or 8 to 11 hours), RFP that has been accumulating also reaches a steady state concentration in the cell when degradation compensates for production. Therefore, we have to take into account both these phenomena when modeling the time response

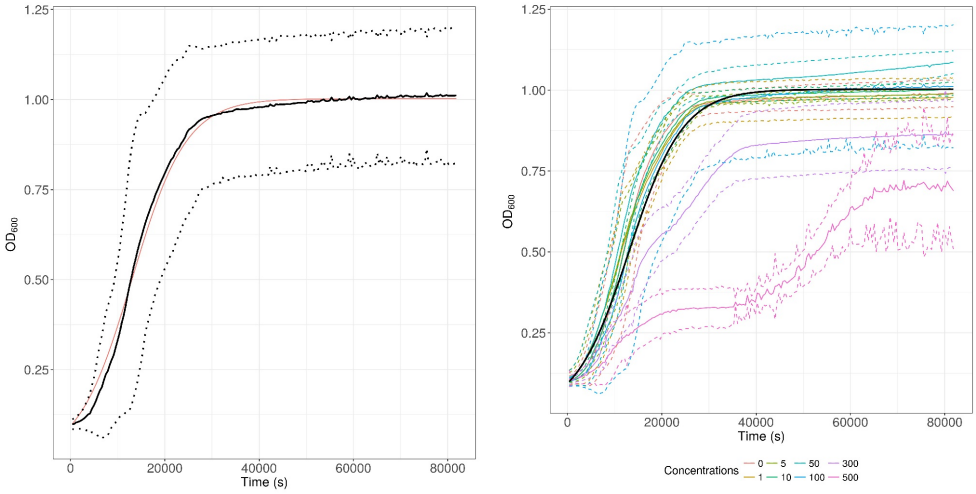
of our biosensors, instead of using a constant term for k as is usually done. We therefore chose to model it that way:

$$k(RFP) = (k_{dil}(t) + k_{deg}) \times RFP \tag{7.10}$$

where k_{deg} is an unknown constant that will be fitted to the data and $k_{dil}(t)$ is a function that accounts for dilution, as we consider first-order degradation and not more complex mechanisms such as Michaelis-Menten degradation. Our aim here was not to fully account for this phenomenon, since we were focusing on characterizing and modeling the effect of copy number. Therefore, we chose a logistic growth as a simple model for modeling bacterial growth:

$$OD(t) = \frac{OD_m \times OD_0 \times exp^{k(t)}}{OD_m + OD_0 \times (exp^{k(t)} - 1)} \tag{7.11}$$

$$k_{dil}(t) = k \times \frac{OD_m - OD_0}{OD_m + OD_0 \times (exp^{k(t)} - 1)} \tag{7.12}$$



(a) (b)

Figure 7.10 **Growth model fitting to construct 357. a** Model (in red) and data (in black) with 95% confidence interval. **b** Model (in black) and varying concentrations ranging from 1 µM to 500 µM using the same parameters.

This model was fitted to the OD on one set of experiments (construct 357, concentration of 100 µM of pinocembrin, (Supplementary Figure 7.10a) and then used to model all other concentrations (Supplementary Figure 7.10b). The growth rates of different constructs are similar and independent of the resistance cassette chosen (Supplementary Figure 7.11). As we can see, this model represents rather well growth for low concentrations but is not built to account for toxicity at higher concentrations. Therefore, the first hours of the time evolution are imperfectly fitted

as this was not fitted for individual constructs and experiments. The aim here was to use only one set of parameters for all constructs and experiments, where another approach would have been to fit the logistic growth model to all individual experiments and then use these parameters. Since the values we are interested in for a biosensor are steady state values, and more precisely does-response curves, using the same set of parameters for growth instead of separately fitting this model to OD data is not an issue. Moreover, these dilution parameters mostly account for what happens at the beginning of the induction, since it is obvious from the dilution equation that $\lim_{t\to\infty} k_{dil}(t) = 0$. This confirms that using the same parameters is not an issue as the values we are interested in for a biosensor are steady state values.

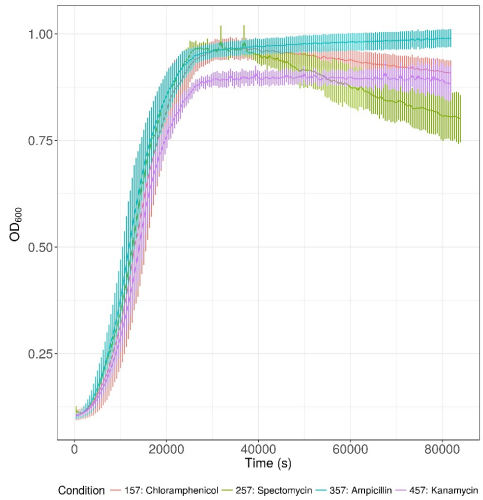


Figure 7.11 **Growth rate of constructs with varying resistance markers** Solid lines represent mean of the experimental data and error bars standard deviation. Different colors represent different constructs as indicated on the Figure.

Steady state- transfer function

Before focusing on the production term that was our main interest in the previous sections, we will show how we derive dose-response equations from the time-course model. As can be seen from the dilution equation, $\lim_{t\to\infty} k_{dil}(t) = 0$. Therefore, at steady state,

$$\lim_{t\to\infty} RFP = \frac{P(inducer, construct)}{k_{deg}}$$

, which justifies why we could study steady-state without considering time evolution.

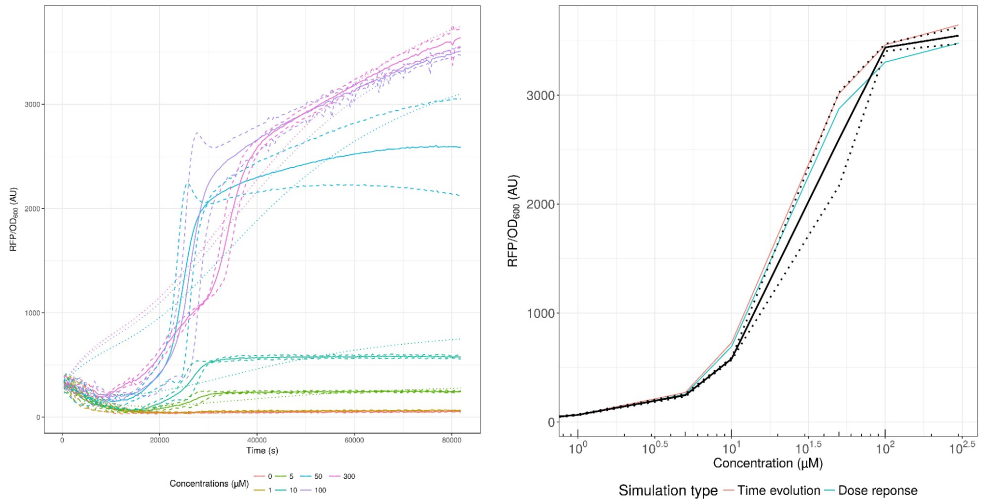
Production term

The production term is classically derived as

$$P = \alpha \times basal \times (1 + P_{fold}(inducer, construct)) \tag{7.13}$$

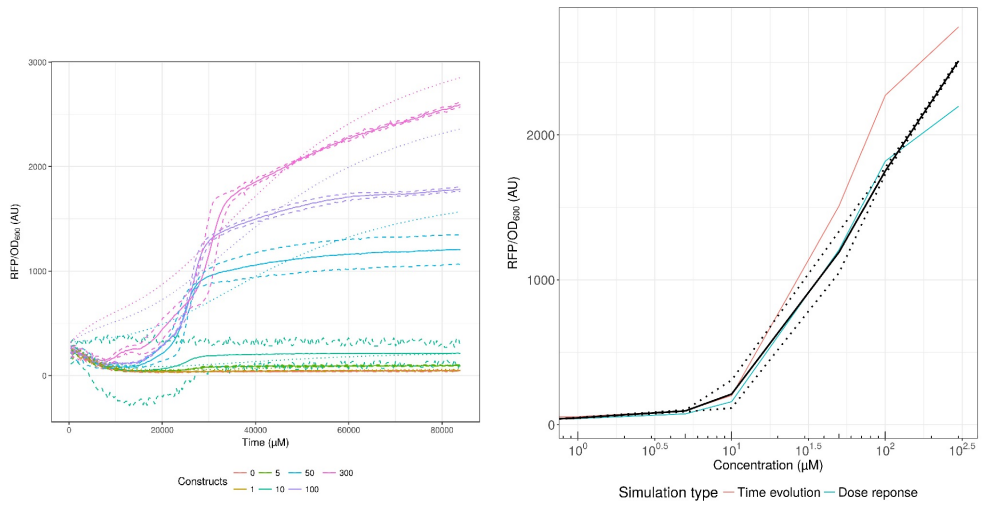
where α takes into account the production, fluorescence intensity and gain of the measurement apparatus. *basal* represents the basal production for the dose-response curve, or the production without induction: it is 1 when we are considering already normalized data (by fold change). The transfer function is the one that accounts for the variations between constructs and the effect of the inducer, which was developed and analyzed in the previous sections.
Therefore, the only 2 parameters that need fitting are k_{deg} and α. This was done on construct 357, with a concentration of 100 µM of pinocembrin according to the procedure presented in materials and methods. We then simulated the expected dose-response from this time simulation and compared it to the steady-state model and the dose-response curve.
We can see in Supplementary Figure 7.12 that the model and the data are in rather good agreement when steady-state behavior is reached, but that the initial dilution is not well taken into account. This is even worse when no time-dependent dilution is included (data not shown) as the curve has to be monotonous and therefore cannot account for the drop in normalized fluorescence. The dose response curves are also in agreement, although the time-evolution is still slightly above the data because of the exponential modeling of the degradation.

The same parameters were then used to simulate the time evolution of the same construct with an induction by naringenin instead of pinocembrin and the results are presented in Supplementary Figure 7.13 7(A, B). The overshooting tendency of the simulated data is even more pronounced. This time-course modeling partially allows us to understand the impact of initial dilution on the biosensor's behavior, and emphasizes the need to wait for it to reach steady-state in order for it to be fully functional and decipher between different inducer concentrations. This initial dilution, although not fully accounted for here, partly explains the delay between the signal (present in the media) and the biosensor's full response. The model also allows us to suggest biological consequences of our constructs that were not accounted for and could be of interest to explain more thoroughly this time-delay: a lag-time in protein production or some toxicity effects of our constructs. The shortcomings of this time-course modeling confirm that although it is interesting to see the delay in response of the biosensor signal, modeling the dose-response curve is more important to show characteristics of the biosensor such as changes to the dose-response curve when used for screening pinocembrin-producing strains.



(a) (b)

Figure 7.12 **Time response and dose-response of time-course model for pinocembrin.** **a** RFP divided by OD data. Solid lines represent mean of the data and dashed lines 95% confidence intervals. Model is represented as dotted lines. Different colors represent varying concentrations as indicated on the Figure. **b** Dose response curve. In black solid line is mean of the data, in black dashed line is the 95% confidence interval. In blue is the model simulating only the dose-response and in red in the model taking into account time evolution.



(a) (b)

Figure 7.13 **Time response and dose-response of time-course model for naringenin.** **a** RFP divided by OD data. Solid lines represent mean of the data and dashed lines 95% confidence intervals. Model is represented as dotted lines. Different colors represent varying concentrations as indicated on the Figure. **b** Dose response curve. In black solid line is mean of the data, in black dashed line is the 95% confidence interval. In blue is the model simulating only the dose-response and in red in the model taking into account time evolution.

8 Models for Cell-free Synthetic Biology: Make Prototyping Easier, Better and Faster

This work was published in Frontiers in Bioengineering and Biotechnology by Mathilde Koch, Jean-Loup Faulon and Olivier Borkowski.
Only minor modifications to the published review have been introduced in the Chapter below.

Detailed contribution to this thesis

Cell-free systems offer various advantages over *in vivo* systems, especially in the case of synthetic metabolic circuits development. First of all, they allow for much faster prototyping. Then, in the context of building complex circuitry, they allow for much finer control of parts such as DNA concentration, which is a feature that was essential in the circuits presented later in this thesis (Chapters 9 and 10). When wanting to develop and analyze cell-free systems, a first step is to asses what the state of the art in modeling is in those systems, which is presented in this review Chapter.

Full reference

Koch M., Faulon J.-L., Borkowski O. (2018) Models for Cell-free Synthetic Biology: Make Prototyping Easier, Better and Faster *Frontiers in Bioengineering and Biotechnology*, 10.3389/fbioe.2018.00182.

Contributions as stated in the article

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

8.1 Abstract

Cell-free TX-TL is an increasingly mature and useful platform for prototyping, testing and engineering biological parts and systems. However, to fully accomplish the promises of synthetic biology, mathematical models are required to facilitate the design and predict the behavior of biological components in cell-free extracts. We review here the latest models accounting for transcription, translation, competition and depletion of resources as well as genome scale models for lysate-based cell-free TX-TL systems, including their current limitations. These models will have to find ways to account for batch-to-batch variability before being quantitatively predictive in cell-free lysate-based platforms.

8.2 Introduction

All the processes required to produce proteins in bacteria can be performed by adding DNA to a cell-free platform. After lysis of living cells, transcription, translation, degradation and protein folding continue to operate as they do *in vivo* [310, 68, 404]. Metabolic pathways like glycolysis or pentose phosphate pathway remain active and are used to regenerate ATP and maximize protein production over time [405, 406]. Protein production outside of the cell simplifies gene expression with well-defined parameters, easy to control inputs, faster time scale and less numerous unknown interactions. As a result, many laboratories use cell-free as a prototyping platform to characterize expression of single proteins or complex metabolic pathways [309, 407, 44]. Mathematical models dedicated to cell-free emerged to predict protein production and understand the limits of this new platform. Cell-free properties are close to living organisms as the same processes take place in both systems, yet significant differences exist. For example, molecular crowding [408] and resources distribution [68] are significantly altered in cell-free and there is no resource competition with the host. Such differences oblige synthetic biologists to adapt the models already developed for living cells. This short review focuses on the recent deterministic models developed to understand lysate-based cell-free platforms and used to predict the behavior of simple or complex pathways 8.1. Those models pave the way for efficient metabolic engineering in the emerging field of cell-free synthetic biology.

Modeling strategy	**Problems tackled**	**Level of detail**	**Description**	**Strength**	**Weakness**	**References**
ODE	Protein production in cell-free	Simple*	Michaelis-Menten for the translation processes, s well as for degradation	Simple, quantitative	Parameters values are experiment dependent	[311]

ODE	Protein production in cell-free	Simple*	Simple description of the transcription and translation processes. One parameter summarizes each	Simple, qualitative	Parameters values are experiment dependent	[409, 410]
ODE	Protein production in cell-free. Resource competition in cell-free	Complex*	Detailed modeling: Particular focus on the translation process and the competition for the translation machinery	General, quantitative	Parameters values are experiment dependent	[411, 44]
ODE	Protein production in cell-free. Resource competition in cell-free	Simple*	Binding, unbinding and elongation bundled in one parameter. Accounts for number of RNA polymerase (RNAP), ribosomes, and promoter/RBS strengths	Easily adaptable to a new situation or phenomena	Necessitates parameter determination for new situation	[412, 413, 173] (and Chapter 9)
ODE	Protein production in cell-free. Resource depletion in cell-free	Complex*	Accounts for all known processes, including nucleoside triphosphate (NTP) consumption and degradation	Models biochemical phenomena precisely so generalisable	Important parameter identification/estimation needed	[414, 415, 416, 417, 308]
Constraint based	Protein production in cell-free. Metabolism in cell-free	Complex*	Modification of *E. coli* metabolic model to account for cell-free constraints	Accounts for the full metabolism, can use constraints based methods such as FBA	Defining the objectives, require deeper knowledge of reactions in cell-free	[418]

Table 8.1 **Deterministic models developed to understand cell-free**: Star Simple stands for one protein produced and limited amount of parameters (less than 10) Complex stands for more than one protein produced or/and large amount of parameters (more than 10)

8.3 Translation and transcription processes in cell-free

Lysate-based cell-free consists of a crude cell extract supplemented with buffer, amino acids, deoxyribonucleotide triphosphate (dNTP), nicotinamide adenine dinucleotide (NAD), polyethylene glycol (PEG), transfer ribonucleic acid (tRNA) and metabolic intermediates [68]. A major advantage of cell-free is the absence of host regulations [310], allowing circuits to function in isolation and an easy quantitative description of gene expression. A constitutively expressed gene in cell-free exhibits specific patterns at the translation and transcription levels. Protein production can be divided in 4 phases: in phase 1, the production rate increases over time, in phase 2, the production rate is constant during around 30 min- 1 h, in phase 3 the production rate decreases slowly and eventually in phase 4 the production rate is null [410] (Figure 8.1A). A similar 4 phases pattern is observed with the mRNA concentration [410] (Figure 8.1B). ordinary differential equation (ODE) models describing transcription, translation, mRNA and protein degradation processes at various scales have been successfully used to predict mRNA and protein dynamics in lysate-based systems [409, 415, 410, 416, 417, 44, 308]. DNA concentrations are usually considered constant: degradation is neglected as plasmid DNA or protected linear template are used, and replication is considered not to happen since no dNTPs are added to the reaction mix.

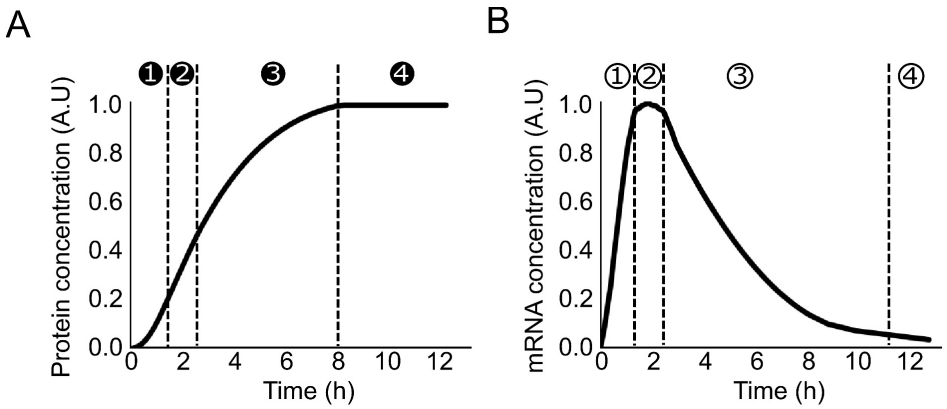


Figure 8.1 **Production of a constitutively expressed gene in cell-free A** Protein concentration over time in a cell-free platform. ❶ the production rate increase over time ❷ the production rate is constant ❸ the production rate decreases ❹ the production rate is null **B** mRNA concentration over time in a cell-free platform ① the concentration increases over time ② the concentration is constant ③ the concentration decreases over time ④ the concentration is close to zero

Numerous models, with varying degrees of complexity, try to reproduce those production phases observed in cell-free reactions.
A simple model based on only 4 reactions and ten parameters is sufficient to fit the full mRNA and protein dynamics during the first hours of reaction [311]. The transcription process is reduced to one step in which the RNA polymerase binds

to the DNA; the rate of mRNA production depends only this binding rate and the DNA length. Similarly, the translation process is described as one binding step of the ribosome on the mRNA with the rate of protein production depending only on the binding rate and the mRNA length. This model is appropriate for the first hours before the consumption of resources and/or the waste accumulation (e.g. ATP degradation, toxic metabolites...) cause the reaction to stop [410]. A simple way to simulate the slow decay in synthesis is the consumption of the NTP over time. The transcription reaction slows down and eventually stops [415, 409]. The decrease in the NTP concentration [405, 419, 308] is an efficient method to obtain a decreasing transcription over time and simulate protein and mRNA production in cell-free but no experimental data either confirms or denies this approach. The accumulation of inactive RNA polymerase/ribosome, [308, 420], accumulation of toxic metabolites [421], or increase of the relative RNases concentration compare to the total amount of mRNA [410] are possible other explanations of the arrest of protein production after 8 hours. [173] (and Chapter 9) added terms corresponding to decreasing resources for protein production and accumulation of toxic by-products as a reduction in production rates parameterized by a Michaelis-Menten like ratio, as an elegant way to account for the slowing production rate. As all cell-free models trying to account for the end of production after 8 hours, the main issue is identifying the exact cause for decreasing production.

Models using Michaelis-Menten kinetics also succeed to capture protein production pattern in cell-free [409]. Those models precisely captured the observable mRNA and proteins dynamics in cell-free while remaining relatively coarse-grained. The model of [415] add extra steps in the transcription (and translation) process with a reversible binding of the RNA polymerase (ribosome) on the DNA (mRNA) followed by a reversible binding of the first NTP (amino acid) and eventually an irreversible elongation step. [308] also developed a model accounting for reversible binding, unbinding and elongation steps, sharing the NTP energy source. Those more detailed descriptions of the transcription and translation processes lead to accurate predictions of the data obtained in cell-free and capture additional properties [409, 415, 44, 308]. For example, the non-additive cost of protein production when several genes are expressed requires higher level of complexity to be predicted [44].

While all models presented in this section described transcription and translation processes, the main challenge they faced is proper parameter identification, as biochemical parameters can vary widely from batch to batch and from *in vivo* to cell-free systems. Currently, models often used components concentration measured *in vivo* and estimated their concentration based on the dilution factor of the *E. coli* cytoplasm after the lysate extraction protocol, which is not entirely satisfactory.

8.4 Resource competition in cell-free

Resource competition is an important phenomenon that impacts circuit behavior in cell-free systems and should be accounted for in modeling approaches.

As a fixed amount of resources is present in the cell-free extract, competition has

been measured between synthetic circuits [410, 44, 308]. Some of the previously described models take into account the limitation of each resource and include a fixed amount of transcription and translation machineries to predict the impact of resource competition in cell-free (Figure 8.2A) [414, 44, 308].

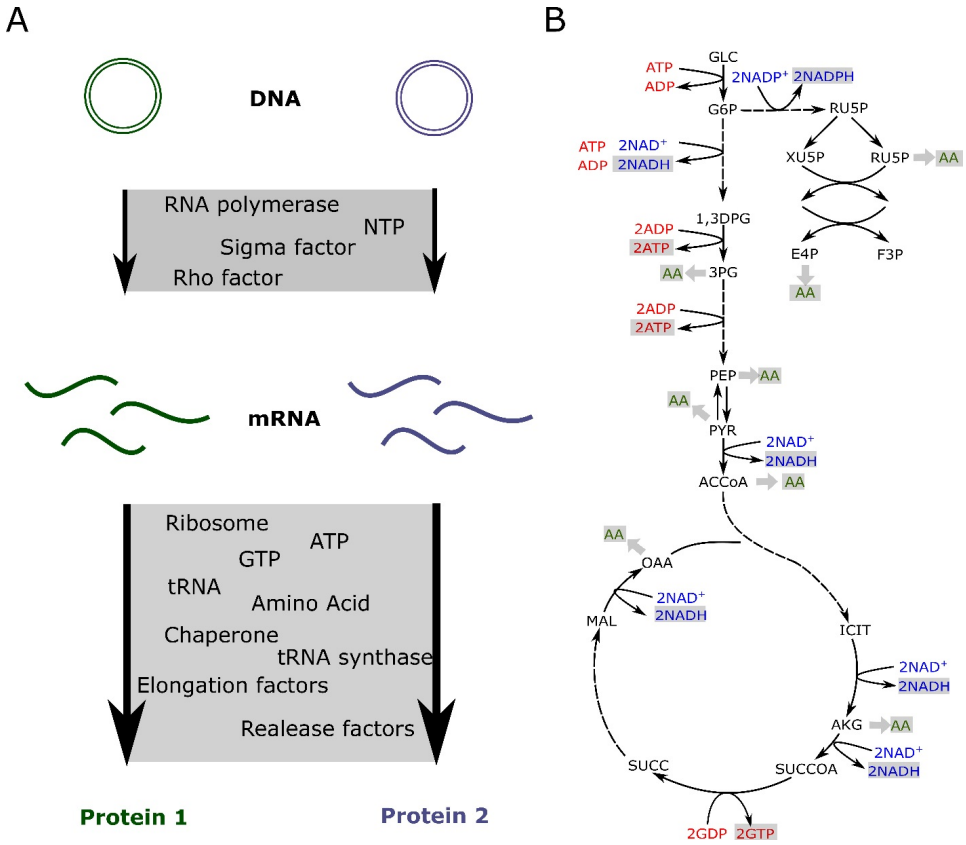


Figure 8.2 **Resource competition in cell-free A** Competition for resources between two genes expressed in cell-free. The transcription and translation machineries amounts are fixed **B** Production of energy and amino acids via the core metabolic network describing glycolysis, pentose phosphate pathway, the tricarboxylic acid cycle (TCA) cycle and the Entner-Doudoroff pathway

A maximal protein production is measured after a few hours before resources depletion and degradation (Figure 8.1). This upper limit on production rate is the result of one or several limited resources (RNA polymerase, NTP, ribosome, elongation factors, amino acids, chaperone, tRNA synthetase or tRNA). DNA, NTP, amino acids and T7 RNA polymerase are directly added to the mix so their impacts on the protein production can be easily measured. Increasing DNA concentration leads to an increase of protein production until a saturation point is reached [410, 44, 173] (and Chapter 9), and toxicity can be observed with high DNA concentration [44]. T7 polymerase [410], amino acids, tRNA and nucleotides [320] are present in excess in the cell-free mix causing no noticeable competition for these resources. Eventually, high NTP concentration negatively affects the translation process [422]. Natural transcription and translation machineries are less controlled as they are added via the crude extract. Indirect measurements using competition

for resources between two plasmids are used to deduce competition for transcription and translation machineries [411, 410, 412, 44, 308]. The main source of competition can be at the transcription and/or translation level depending of the extract and the level of protein produced [411, 410, 423, 412, 44, 308, 173] (and Chapter 9). Parameterization of the models using the appropriate RNA polymerase and ribosomes concentrations and binding/unbinding rates allows an accurate description of the resource competition and to fit properly the production of several proteins expressed concurrently in cell-free [410, 44, 308]. Accounting for RNAPs and ribosomes sharing between parts can also be leveraged to minimize the number of experiments required to fit parameters and obtain a predictive model [413]. While not the main focus of this review, parameter estimation or identification is a major hurdle of detailed models, and techniques from systems biology (e.g. [424]) can be used to tackle this issue.
The models presented in this sections, while being able to account for transcription, translation and resource competition from a lack of generalisability due both to variability in experimental conditions as batches can differ greatly, and to scarcity of biochemical work measuring those parameters in cell-free setting as has been estimated from *in vivo* measurements.

8.5 Metabolism in cell-free

The models presented in this section are constraint based, so as to take the whole metabolism into account and not the circuit in isolation as done in the previous sections.
In cell-free platforms, translation and transcription are not the only active processes. Glycolysis, pentose phosphate pathway, TCA cycle, Entner-Doudoroff pathway and amino acid biosynthesis are still producing ATP, reducing equivalents and amino acids [405, 425, 426]. The previously described models account for the resources competition for a fixed amount of transcription and translation resources and usually do not include any metabolite production or consumption. Such an approach is quite limiting for metabolic pathway prototyping as those circuits also compete for metabolites [407, 44]. Constraint-based models have been used to simulate metabolites production and consumption when various proteins are produced at different levels [418] (Figure 8.2B). This model coupled transcription and translation processes with the availability of metabolic resources. Flux balance analysis was adapted to cell-free conditions with the objective function being the maximization of the protein translation rate. Growth associated reactions were removed and cell-free specific deletions were added from *E. coli* metabolic model, leading to 264 reactions and 146 species [418]. The stoichiometric network was adjusted to cell-free and fluxes were constrained by experimental measurements of glucose, nucleotides, amino and organic acid consumption and production rates. The transcription and translation were bound by Michaelis-Menten formula with a maximum transcription and translation rate depending on RNA polymerase concentration, RNA polymerase elongation rate, gene length, promoter strength and the ribosome concentration, a polysome amplification constant, the ribosome elongation rate, the

protein length, and the RBS strength respectively. The energy efficiency was calculated using the ATP cost by transcription and translation processes. Transcription and translation rates are subject to resource constraints encoded by the metabolic network (Figure 8.2B). This model efficiently predicts proteins production and simulates optimal flux distribution in cell-free metabolic network. It makes predictions possible for metabolic engineering in cell-free as metabolites produced or consumed by a pathway will be accounted for via its energy efficiency.
Constraint based modeling for cell-free systems is an interesting field that would need further developments from the research community, both to include cell-free specific constraints and reactions, as well as to account for dynamic behavior such as metabolite exhaustion in cell-free systems.

8.6 Conclusion

Cell-free appeared as the ideal platform for circuits prototyping. It accelerates characterization and avoids the impact of the host on the circuit behavior. Models can be easily parameterized and predictions are easier and more accurate than *in vivo*, for qualitative behavior. Parameterization for quantitative behavior can be tackled using techniques from systems biology. Simple models succeed to accurately predict the simultaneous production levels of multiple proteins and the competition for the limited amount of resources in cell-free. A certain level of complexity is necessary to capture competition for metabolites but produces a powerful tool for metabolic engineering. The main limit for lysate-based cell-free in metabolic engineering and modeling remains extract preparation: extract efficiency can differ strongly depending on the experimentalist leading to variability of protein production and necessity of robust controls for each new batch, as well as uncertain parameters that vary with each batch for the modeler. Preliminary control of the extract quality and tuning of model parameters on each batch is required to obtain accurate predictions and precludes generalization. A way forward to both increase reproducibility and predictive modeling in cell-free systems would be a higher degree of automation in the extract production providing robust lysate preparation at affordable price.

9

Plug-and-Play Metabolic Transducers Expand the Chemical Detection Space of Cell-Free Biosensors

This work was originally published in Nature Communications by Peter L. Voyvodic, Amir Pandi , Mathilde Koch, Ismael Conejero, Emmanuel Valjent, Philippe Courtet, Eric Renard, Jean-Loup Faulon and Jerome Bonnet.
The main focus of this article is to showcase the development of biosensors in cell-free using metabolic transducers. Mathematical modeling was therefore only a small part of the original article, presently mainly in Supplementary. The text ordering has therefore been modified so as to highlight results concerning this thesis. Supplementary data not concerning mathematical modeling is appended at the end of this Chapter.

Detailed contribution to this thesis

In this article, an implementation of cell-free metabolic transducers, designed used tools presented in Part I, is presented. First, a biosensor is developed and optimized, tuning both transcription factor DNA and reporter DNA. My first contribution to this work was therefore to model this assay using Hill equations. Then, transducers (signal conversion using enzymes) were implemented in those cell-free systems, for hippuric acid and cocaine. A major point of interest in those 2 transducers for modeling was that the signal peaked before decreasing at intermediate enzyme DNA concentrations, showing that resource competition has an important role in our observed effects. Therefore, I proceeded in 2 steps, by first modeling the hippuric acid transducer including resource competition, and applying the same framework to cocaine, accounting for differences in promoter strengths. This modeling was then experimentally verified by analyzing a shift in peak signal of the hippurate transducer when varying transcription factor DNA, thereby increasing resource competition.

Full reference

Voyvodic P., Pandi A., Koch M., Conejero I., Valjent E., Courtet P., Renard E., Faulon J-L. and Bonnet J. (2019) Plug-and-Play Metabolic Transducers Expand the Chemical Detection Space of Cell-Free Biosensors. *Nature Communications*, 10.1038/s41467-019-09722-9.

Contributions as stated in the article

P.L.V., A.P., J.-L.F. and J.B. designed experiments, P.L.V. and A.P. cloned constructs and performed experiments, and M.K. constructed the computer model simulations. I.C., P.C., E.R. and E.V. participated in clinical sample collection and analysis. P.L.V., A.P., M.K., J.-L.F., and J.B. wrote the paper. All authors approved the manuscript.

9.1 Abstract

Cell-free transcription-translation systems have great potential for biosensing, yet the range of detectable chemicals is limited. Here we provide a workflow to expand the range of molecules detectable by cell-free biosensors through combining synthetic metabolic cascades with transcription factor-based networks. These hybrid cell-free biosensors have a fast response time, strong signal response, and a high dynamic range. Additionally, they are capable of functioning in a variety of complex media, including commercial beverages and human urine, in which they can be used to detect clinically relevant concentrations of small molecules. This work provides a foundation to engineer modular cell-free biosensors tailored for many applications.

9.2 Introduction

There is currently an urgent need for low-cost biosensors in a variety of fields from environmental remediation to clinical diagnostics [427, 428, 358]. The ability of living organisms to detect signals in their environment and transduce them into a response can be utilized to create cheap, novel sensors with high sensitivity and specificity. By leveraging the ability of transcription factors to control gene expression, synthetic biologists have genetically engineered microbes to detect a wide range of compounds, from clinical biomarkers to environmental pollutants [429, 288, 430, 431].

Cell-free TX-TL systems have great promise as the next generation of synthetic biology-derived biosensors. They are cheap to produce [68], abiotic, and can be lyophilized such that they are stable at room temperature for up to one year: a

vital necessity for point-of-care applications such as low-resource nation and home diagnostic use [74]. Cell-free TX-TL toolboxes have been designed that support the operation of many of the circuits previously engineered *in vivo* [432, 433]. Encapsulated cell extracts can also be used in combination with living cells to produce new sensing modalities [434]. Cell-free biosensors were engineered to successfully detect Zika virus in rhesus macaques and an acyl homoserine lactone, 3OC12-HSL, from *Pseudomonas aeruginosa* in human clinical samples [435, 23]. However, current cell-free biosensors have been limited to detection of nucleic acid sequences, via toehold displacement, or well-characterized transcription factor ligands.

Here we put forward a generalized, modular workflow utilizing metabolic transducers to rapidly expand the chemical space detectable by cell-free biosensors in a plug-and-play manner. We then illustrate our workflow with a proof-of-concept example: the transcription factor BenR, which is activated by benzoic acid, and two metabolic modules, HipO and CocE, which convert hippuric acid and cocaine, respectively, into benzoic acid. Each component is individually cloned into a cell-free vector, such that the DNA concentrations can be titrated over three orders of magnitude to optimize sensor performance. Finally, we demonstrate that these sensors can function in complex solutions, detecting benzoic acid in commercial beverages and hippuric acid and cocaine in human urine.

9.3 Results

9.3.1 Design workflow for cell-free biosensors

Synthetic metabolic cascades have been used by the synthetic biology community for a wide range of applications, including production of biofuels, pharmaceuticals, and biomaterials [184, 436, 437]. As such, there is a wide variety of well-characterized enzymes catalyzing various reactions transforming one molecule into another. Our framework harnesses this power by using metabolic enzymes as transducers to allow us to 'plug in' a given enzyme into our characterized biosensor modules to detect a ligand with no known transcription factor analog (9.1a). Specifically, the metabolic enzyme converts the undetectable molecule into one for which we have an existing transcription factor-based genetic circuit (9.1b). We used the SensiPath webserver that we previously designed and validated *in vivo* to determine the required metabolic cascade [262, 77].
The workflow to engineer a cell-free biosensor detecting a novel molecule is straightforward (9.1c). First, possible metabolic pathways to convert the molecule of interest into a detectable ligand are identified using SensiPath. Second, the genes coding for the metabolic transducer enzyme, the TF sensor, and the reporter module are synthesized and cloned into cell-free expression vectors. Finally, the DNA concentration of each plasmid is titrated in cell-free reactions to optimize signal strength and dynamic range in response to the molecule of interest (9.1c).

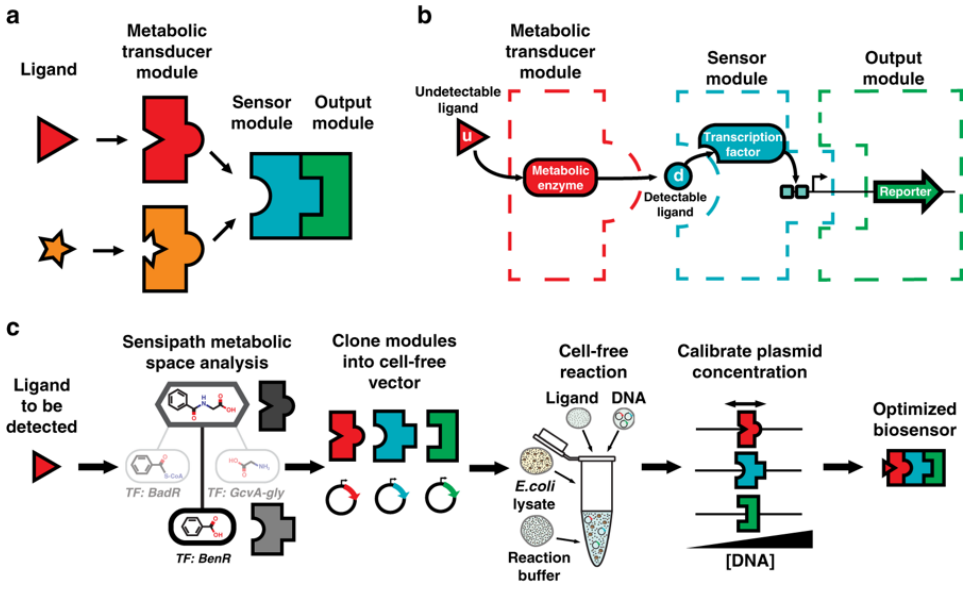


Figure 9.1 **A modular design workflow for engineering scalable cell-free biosensors.** **a** Cell-free biosensors are composed of three modules: a generic sensor module linked to an output module and a metabolic transducer module transforming different molecules into ligands detectable by the sensor module. **b** An undetectable ligand is converted into a detectable ligand by the enzyme from the transducer module. Binding to the transcription factor controls the sensor module and downstream gene expression. **c** The biosensor design workflow starts with retrosynthetic pathway design using the SensiPath server [77]. Once the transducer and sensor modules are determined, the genes encoding enzymes, transcription factors, and target promoters driving a reporter are cloned into cell-free expression vectors. The sensor is calibrated by titrating the concentrations of each plasmid to maximize signal output and dynamic range.

As a proof-of-concept example of this system, we engineered a sensor for benzoic acid using the transcription factor BenR and expanded its detection capabilities with two different metabolic transducers: one for hippuric acid using the HipO hippurate hydrolase and one for cocaine using the CocE cocaine esterase.

9.3.2 Optimization of cell-free benzoic acid sensor

BenR is a member of the AraC/XylS family of transcription factors, originally from Pseudomonas putida . In the presence of benzoate, BenR binds to the P_{Ben} promoter and activates transcription (9.2a). To engineer a benzoate cell-free biosensor, we cloned BenR under the control of the OR2-OR1-Pr promoter, a modified version of the lambda phage repressor promoter Cro, known to express strongly in cell-free systems [320]. The P_{Ben} promoter driving super-folder GFP (sfGFP) was cloned in a separate plasmid. After initial pilot tests demonstrated that BenR was functional in a cell-free environment, we optimized the BenR biosensor by titrating the DNA concentration of the TF and reporter plasmids.

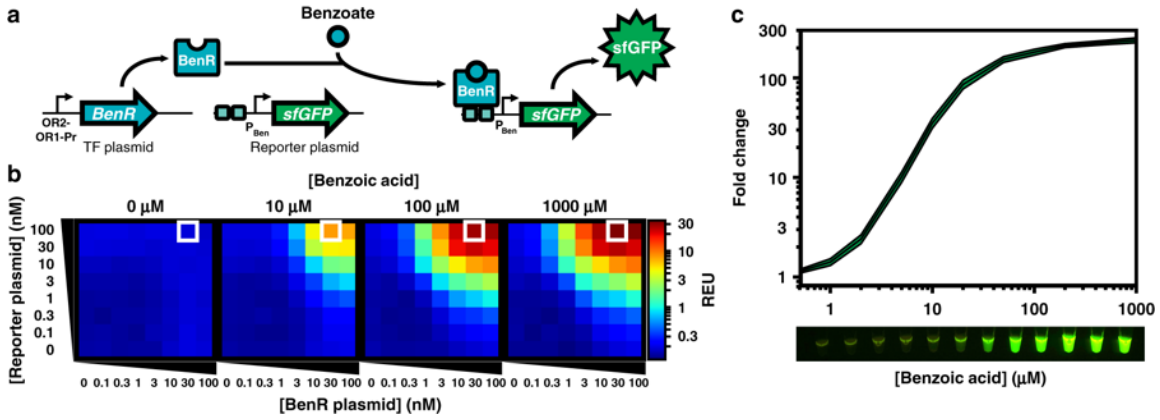


Figure 9.2 **Calibration of sensor and output modules for benzoate detection. a** BenR binds to the P_{Ben} promoter in the presence of benzoate and activates gene expression. Here BenR is cloned in the pBEAST plasmid (a derivative of pBEST 20) and driven by a strong constitutive promoter, OR2-OR1-Pr. The P_{Ben} promoter is cloned into another pBEAST backbone and drives expression of the superfolder green fluorescent protein. Because the system operates without a cellular boundary, multiple plasmids encoding different components of the network can easily be used simultaneously. Plasmid concentrations can then be fine tuned to identify optimal operating conditions. **b** Optimization of the BenR sensor and reporter modules. Cell-free reactions of 20 µL containing different concentrations of the BenR and reporter plasmids were prepared and their response to different concentrations of benzoic acid were monitored. The white square represents the optimal condition (100 nM reporter and 30 nM BenR plasmid) with the highest relative fluorescence. (see 9.9 Supplementary Figure 2 and 9.6 Supplementary Table 1). Reactions were run in sealed 384 well-plates in a plate-reader at 37 °C for at least eight hours. The heat maps represent the signal intensity after four hours. Data are the mean of three experiments performed on three different days and all fluorescence values are expressed in Relative Expression Units (REU) compared to 100 pM of a strong, constitutive sfGFP-producing plasmid. See methods for more details. **c** Upper panel : The BenR sensor can detect benzoic acid over three orders of magnitude and at concentrations as low as 1 µM. Shaded area around curves corresponds to ± SD from the mean of the three experiments. Lower panel : GFP expression in response to the same range of concentrations of benzoic acid as in the upper panel is easily detectable by eye on a UV table.

One advantage of working in a cell-free framework is that the DNA concentration is directly controlled by pipetting. As such, the process of finding an optimal DNA

concentration is relatively straightforward: we created a matrix of DNA concentrations for TF and reporter plasmids between 0 nM and 100 nM and induced these different cell-free reactions using four different concentrations of benzoic acid: 0 µM, 10 µM, 100 µM and 1000 µM (9.2b, 9.6).
Encouragingly, the system had extremely low background signal in the absence of benzoic acid, indicating that the P Ben promoter has very little 'leakiness' in a cell-free environment. When benzoic acid was added to the reaction, the sfGFP output signal was clearly detectable and fluorescence intensity was correlated with increasing reporter plasmid concentration. However, the signal reached a plateau for increasing concentrations of TF plasmid at 30 nM. We hypothesize that this plateau is due to competition for transcriptional and translational resources between transcription factor and reporter plasmid. This plateau is also observed in a mathematical model of cell-free biosensors (See section 9.6 and Fig 9.3). Based on these data, we set the optimal plasmids concentrations to 30 nM for the TF plasmid and 100 nM for the reporter plasmid.

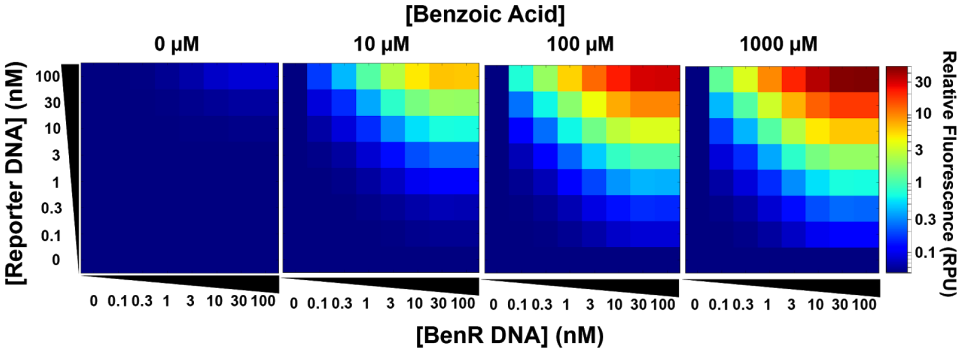


Figure 9.3 **Modeling titration of transcription factor and reporter plasmids.** Conditions for reporter and BenR DNA concentrations used in Figure 9.2 were modeled using ordinary differential equations to capture qualitative trends in the data. Simulations were rescaled to use the same scale as data. The heat-map represents GFP model signal after four hours.
This Figure was originally in Supplementary data

Compared to its *in vivo* counterpart [262], the cell-free benzoic acid biosensor is faster (maximum signal reached in four hours, 9.9), has a much higher sensitivity and dynamic range, and has a maximum fold change of over 200 (vs. ≃10-fold *in vivo*) (9.2c). These results exemplify the advantages of cell-free systems for rapidly engineering biosensors with optimal properties.

9.3.3 Expansion of benzoic acid sensor with hippuric acid and cocaine metabolic modules

With the sensor and output modules optimized, we demonstrated the ability of our system to expand its chemical detection space using different metabolic transducer modules. HipO is an enzyme from *Campylobacter jejuni* and CocE is an esterase from *Rhodococcus sp.* that convert hippuric acid and cocaine into benzoic acid, respectively. We cloned each enzyme into the cell-free expression vector and, using the optimized DNA concentrations of TF and reporter plasmids, titrated different concentrations of metabolic transducer DNA for a range of inducer inputs (Figure 9.4a, Table 9.7). Interestingly, we observed a clear peak in sfGFP signal corresponding to a particular concentration effectiveness: 3 nM for HipO and 10 nM for CocE. We built several mathematical models based on different assumptions that could reproduce the observed bell-shaped response to enzyme DNA concentration as well as its shift between the two enzymes (Figure 9.6). Based on these models, we hypothesized that the observed bell-shaped response is likely due to competition between the different modules, leading to an important and unnecessary enzyme production at high DNA concentrations that divert resources such as RNA polymerase, ribosomes, and energy from sfGFP transcription and translation, as well as generating toxic byproducts. Moreover, we provide evidence that the shifting peak between the two setups is most likely due to lower expression of CocE (Detailed analysis follows, in Supplementary Text and Figure 9.7). Additionally, the model hypothesized that using a higher TF concentration would necessitate a higher level of metabolic enzyme without an increase in overall signal, a shift that we subsequently saw experimentally (Supplementary Text and Supplementary Figure 9.8).

A key observation is that even at very high levels of inducer, there is very little signal in the absence of DNA encoding the metabolic transducer. These data indicate that the metabolic enzyme is essential for sensor selectivity and differentiation between hippuric acid and cocaine from benzoic acid and that they have minimal off-target binding to BenR. Strikingly, both the hippuric acid and cocaine biosensors exhibit fold change and detection range highly similar to that of the benzoic acid sensor, demonstrating the high conversion rate of the metabolic transducer (Figure 9.4b). The conversion also appears to be extremely fast as no significant difference was observed in response kinetics with or without the metabolic transducer, although the lower incubation temperature of the cocaine biosensor showed slightly slower kinetics (Figures 9.9, 9.10 and 9.11).

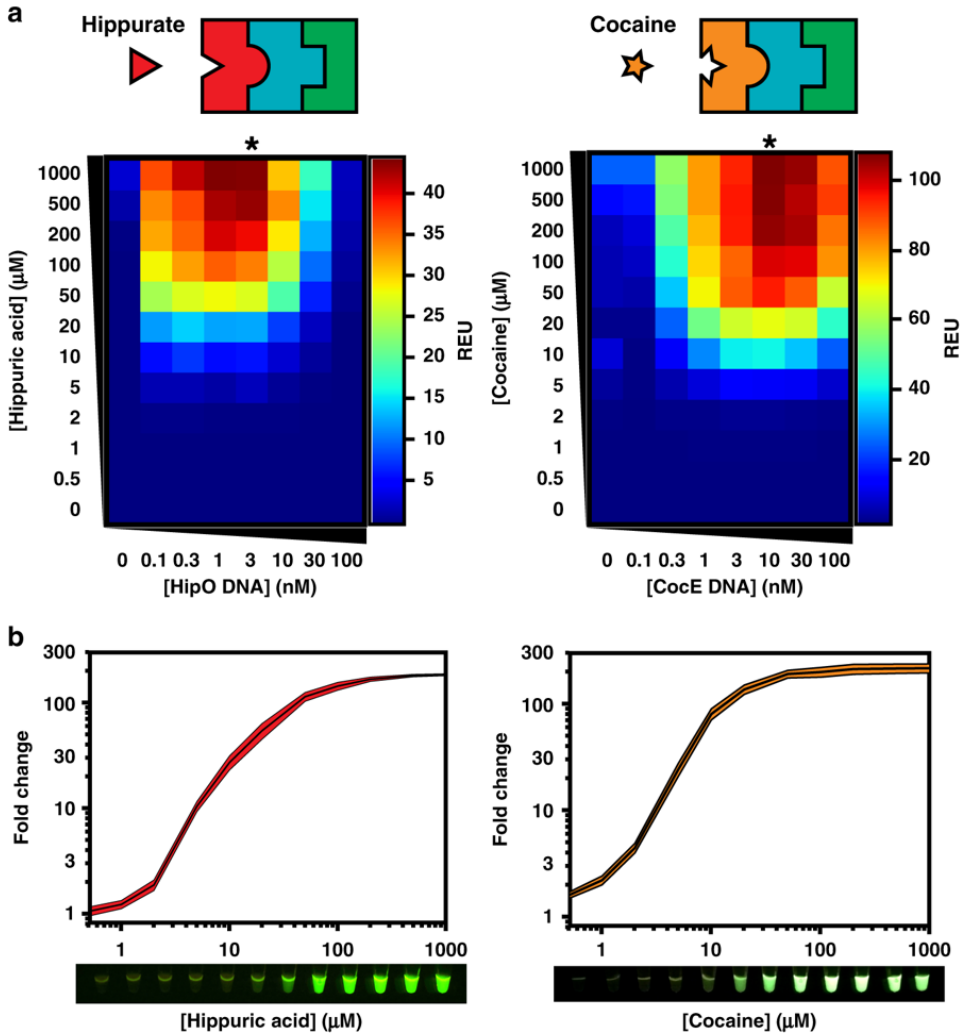


Figure 9.4 **Expanding the chemical detection space of cell-free biosensors by plugging various metabolic transducers into an optimized sensor module. a** Hippurate or cocaine can be detected using different metabolic transducers. Plasmids encoding the HipO or CocE enzymes, which convert hippuric acid or cocaine into benzoic acid, were mixed at different concentrations with optimal BenR and reporter plasmids concentrations as determined in Figure 9.2 (30 nM and 100 nM, respectively). These reactions were then incubated with increasing concentrations of inducer for at least eight hours. The heat maps represent the signal intensity after four hours (Figure 9.10, Figure 9.11 and Table 9.7). Asterisks denote the optimal DNA concentration for the metabolic module. Data are the average of three experiments performed on three different days and all fluorescence values are expressed in Relative Expression Units (REU) compared to 100 pM of a strong, constitutive sfGFP-producing plasmid. **b** Optimized cell-free biosensors incorporating a metabolic transducer module exhibit comparable performance to the BenR sensor module (from Figure 9.2c). All data are the mean of three experiments performed on three different days. Shaded area around curves corresponds to ± SD from the mean of the three experiments. See methods for more details. *Lower panel*: GFP expression in cell-free reactions in response to various concentrations of inducer visualized on a UV table.

9.3.4 Detection of benzoic acid, hippuric acid, and cocaine in complex samples

While the results of our new optimized biosensing were promising, the intended final environment in which they should operate is far more complex. We thus sought to test their capabilities for real-world applications. Benzoic acid and sodium benzoate are widely used food additives for preservation. While classified as Generally Recognized As Safe (GRAS) by the United States Food and Drug Administration, their maximal levels in foodstuffs are limited to 0.1%. Additionally, some people respond poorly to their consumption, particularly patients suffering from chronic inflammation or orofacial granulomatosis, who are frequently placed on benzoate-free diets by their physicians [438, 439]. Lastly, there is evidence that when benzoates are added to beverages in the presence of ascorbic acid, they can be converted into low levels of benzene, a strong carcinogen [440, 441]; this reaction is enhanced by increased temperatures which frequently occur during transportation. In this context, a simple assay for detecting benzoic acid could be useful.

To test if our benzoic acid sensor could function in a monitoring capacity in the food industry, we procured several different carbonated orange and energy drinks from a local supermarket. The nutritional information of each beverage included benzoic acid, sodium benzoate, or no benzoates. Strikingly, after adding 2 µL of the beverages directly to 20 µL reactions of our optimized benzoic acid sensor, we were able to distinguish which beverages contained benzoates with 100% accuracy after only one hour of incubation (Figure 9.12). The beverages were composed of two categories: carbonated orange drinks and Monster®energy drinks. Despite similarities between the non-benzoate ingredients in each class, our cell-free benzoic acid biosensor rapidly produced sfGFP in beverages with listed benzoate ingredients with fold changes up to $\simeq 180$.

While our system has the ability to quickly detect benzoates by directly adding the beverages to the reaction, we noticed that there was up to 75% inhibition to some of the cell-free reactions when comparing expression of a constitutive promoter to a control (Figure 9.13). Therefore, to test our sensor's ability to quantify benzoates, we performed an experiment with a 1:10 dilution, which showed minimal reaction interference (Figure 9.13), and converted the resulting fluorescence intensities to concentrations using a calibration curve from a benzoic acid standard (Figure 9.14). These results were compared against measurements from LC-MS (Figure 9.5b , Table 9.8). Seven of the ten drinks showed very strong agreement between the quantitative results from our sensor and the LC-MS results. Three of the beverages (Monster®Zero, Monster®Ultra, and Monster®Ultra Red) had diminished cell-free values relative to those from LC-MS. Taken together, these results demonstrate that our sensors can remain functional in commercial products and rapidly detect and quantify benzoates.

We then wanted to test if our hippuric acid sensor could detect endogenous levels in a clinical context. Hippuric acid has long been known to be regularly excreted by humans in urine as the end product of several different aromatic compounds, including benzoates, that are converted in the liver [442]. While it has been correlated with higher levels of toluene exposure in some operational conditions [443],

following recent research by [25] it has recently become an interesting biomarker in a Phase 1/2a clinical trial. In the publication, a synthetic strain of modified *E. coli* Nissle, SYNB1618, is used to treat phenylketonuria, a neurotoxic disease characterized by the inability to process the amino acid phenylalanine [25]. Briefly, the bacteria are consumed orally where they can convert phenylalanine into *trans*-cinnamate, which is subsequently converted to hippuric acid by the liver. In the study, hippuric acid in the urine is used as a biomarker for treatment efficacy. We thus wanted to test if our sensor could detect clinical levels of endogenous hippuric acid in human urine. When adding 2 µL of a 1:10 dilution to a 20 µL reaction (1% cell-free reaction concentration) in the presence of an RNase inhibitor, we found little interference from urine to expression of a constitutive GFP plasmid relative to the positive control (Figure 9.15). When testing the urine for hippuric acid, we observed little to no response from our benzoic acid sensor (without the HipO-expressing plasmid) (Table 9.9), but the complete hippuric acid sensor gave levels that fell within our calibration curve (Figure 9.16). Urinary hippuric acid concentrations estimated using our cell-free biosensor closely matched the values determined byLC-MS ($R^2 = 0.98$, Figure 9.17; 9.5c, Table 9.10). These data are a promising step toward developing cell-free biosensors for biomarker detection in clinical samples.

Finally, we aimed to detect cocaine in clinically relevant conditions. Cocaine rapidly enters the bloodstream after ingestion and is subsequently detectable in the urine for up to 10 hours [444]. To determine if our system could detect clinically-relevant cocaine levels, we spiked urine samples with increasing concentrations of cocaine and added 2 µL to 20 µL cell-free reactions with our optimized cocaine biosensor. Our initial experiment showed small, but detectable sfGFP signal at urinary concentration of 1000 µM, but our system was unable to show adequate fold-change at lower, clinically relevant concentrations (Figure 9.18). We found that cell-free reactions produce increasing low levels of noise over time in the GFP fluorescence channel (9.19) and hypothesized that we could increase our signal-to-noise ratio by changing our reporter to luciferase. We cloned the firefly luciferase gene under control of the P_{Ben} promoter and in an initial test we indeed observed an increase in signal-to-noise ratio (Figure 9.20). We then added increasing cocaine concentrations into six different samples containing our cell-free cocaine sensor with the luciferase reporter (Figure 9.5d). Five of the six sample showed strong fold change, with detectable fold changes of 4.3-8.8 at previous clinically detected cocaine concentrations in urine [445] (40.13 µg $\cdot$ mL^{-1} or 118 µM cocaine concentration in urine, corresponding to a 11.8 µM final concentration in the cell-free reaction when using 2 µL urine in a 20 µL reaction). One sample (U3) showed minimal fold change due to high background signal that was also observed using the benzoic acid sensor (Figure 9.21). As the urine samples were supplied by subjects from the endocrinology department, it is possible that the medical condition of this patient results in the presence in their urine of interfering metabolites that can activate the BenR system. This background signal was minimal when we detected for hippuric acid in urine, likely because of the urine samples dilution step (Table 9.9). In conclusion, these data demonstrate that our cell-free biosensors can be used to detect clinically relevant levels of drugs and endogenous metabolites in pure, unprocessed clinical samples.

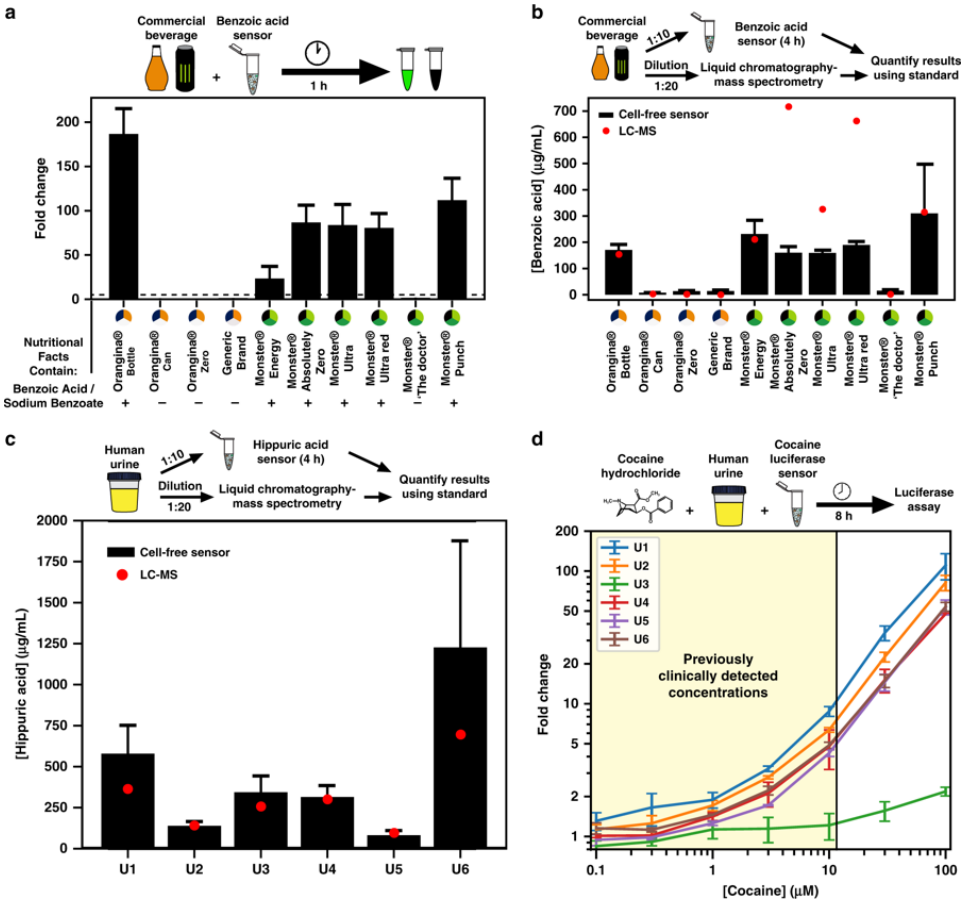


Figure 9.5 **Detecting benzoic acid, hippuric acid, and cocaine in complex samples.** **a** Cell-free benzoic acid sensor can detect benzoates in commercial beverages. Addition of an array of different orange and energy drinks to the optimized benzoic acid biosensor produces up to $\simeq$ 180-fold change response relative to the negative control after one hour incubation at 37 °C. The test showed 100% specificity and sensitivity to detection of benzoates based on their inclusion in the ingredient label using a fold-change of 5 as the cut-off point. **b** Benzoic acid sensor is capable of quantifying the concentration of benzoic acid in different beverages. Beverages were added at 1:10 dilution to cell-free reactions and the benzoic acid concentration was determined using a calibration curve (Figure 9.14) after four hours. Results were compared to those determined by liquid chromatography-mass spectrometry (LC-MS). **c**Endogenous hippuric acid in urine can be quantified with a cell-free biosensor. Clinical urine samples (U1-U6) were diluted 1:10 and added to the optimized hippuric acid sensor for four hours at 37 °C after which endogenous hippuric acid concentration was determined using a calibration curve (Figure 9.16). Results were compared to those determined by LC-MS. **d** Cocaine can be detected in clinical urine samples at previously clinically detected concentrations. Cocaine titrations were added to clinical human urine samples (U1-U6) and cell-free cocaine luciferase-output biosensors and incubated at 30 °C for 8 hours. Subsequently, a luciferase assay was performed to determine the presence of cocaine. The colored region represents the concentration of cocaine previously measured in human clinical samples from hospitalized patients (40.13 µg · mL^{-1} or 118 µM cocaine concentration in urines, corresponding to a 11.8 µM final concentration in the cell-free reaction- 2 µL urine in a 20 µL reaction) 28 . All curves are plotted for the mean of three experiments performed on three different days. Error bars correspond to ± SD from the mean of the three experiments. See methods for more details.

9.4 Discussion

This work demonstrates that we can engineer modular, cell-free biosensors that can be easily calibrated to have high signal strength and dynamic range and can function in complex detection environments. Upon engineering a novel cell-free biosensor for benzoic acid, we show that the system can be scaled by using different metabolic transducer modules to expand the chemical space that each sensor/reporter pair can detect. In addition, we provide a three order-of-magnitude titration for each DNA component to optimize cell-free biosensor performance along with a mathematical model enabling a better understanding of the parameters governing cell-free biosensors response which will help future optimization of such devices . By demonstrating that these sensors can function in samples from the food and beverage industry, as well as complex clinical samples such as human urine, we provide an example for their potential outside the lab in real-world applications. This is the first time, to our knowledge, that cell-free biosensors have been used to detect endogenous molecules in unprocessed samples.

Using our workflow, this process should be applicable to a wide range of other sensor/reporter pairs. One constraint of our system is that the transcription factor must respond only to the product of the enzymatic reaction and not the substrate. Such potential crosstalk can easily be checked by running a control reaction without the metabolic transducer module. We computed that 1205 disease-associated biomarkers from HMDB could be converted into detectable molecules by one enzymatic reaction (Supplementary Note 1 and Supplementary Data 1) . Additionally, 64 HMDB metabolites could be transformed into benzoate and thus theoretically connected via a metabolic transducer to our optimized sensor (Supplementary Note 1 and Supplementary Data 2).

Further improvements to our platform could include exploring sample pre-processing methods that could improve sensor robustness [446, 447] together with adaptation into an off-the-shelf format more amenable to point-of-care applications [74, 448]. Also, while we could detect clinically relevant concentrations of cocaine, this application will likely require achieving higher sensor dynamic range, for example through the use of downstream genetic amplifiers [22].

In summary, by rapidly expanding the number of detectable compounds and remaining functional even in complex samples, cell-free biosensors using plug-and-play metabolic transducers could be used to address many challenges such as environmental detection, drug enforcement, and point-of-care medical diagnostics.

9.5 Methods

9.5.1 Molecular biology

All clones were based on a previously characterized cell-free expression plasmid (pBEST-OR2-OR1-Pr-UTR1-deGFP-T500 was a gift from Vincent Noireaux [Addgene plasmid # 40019] [320]). To better facilitate cloning with a range of tech-

niques and any future component insertion into larger gene circuits, the construct was modified by adding 40 base pair spacers and an upstream terminator and renamed pBEAST. Clones were created via Gibson or Golden Gate assembly in DH5αZ1 chemically competent *E. coli* where the deGFP was replaced by BenR or HipO. For CocE, the promoter was changed to another strong constitutive promoter, J23101, and RBS, B0032. The reporter plasmid for P_{Ben} used native RBS from Pseudomonas putida and superfolder-GFP as the output, which was found to give a stronger, faster signal in cell-free reactions at 37 °C. For experiments testing cocaine levels in urine, the sfGFP output was changed to firefly luciferase via Gibson assembly cloning. DNA for cell-free reactions was prepared from overnight bacterial cultures using Maxiprep kits (Macherey-Nagel). Plasmids used in this paper will be available from Addgene.

9.5.2 Extract preparation

Cell-free *E. coli* extract was produced using a modified version of existing protocols [68, 449]. An overnight culture of BL21 Star (DE3)::RF1-CBD 3 *E. coli* was used to inoculate 660 mL of 2xYT-P media in each of six 2 L flasks at a dilution of 1:100. The cultures were grown at 37 °C with 220 rpm shaking for approximately 3.5 hours until the OD 600 = 2.0. Cultures were spun down at 5000 x g at 4 °C for 12 minutes. Cell pellets were washed twice with 200 mL S30A buffer (14 mM Mg-glutamate, 60 mM K-glutamate, 50 mM Tris, pH 7.7), centrifuging afterwards at 5000 x g at 4 °C for 12 minutes. Cell pellets were then re-suspended in 40 mL S30A buffer and transferred to pre-weighed 50 mL Falcon conical tubes where they were centrifuged twice at 2000 x g at 4 °C for 8 and 2 minutes, respectively, removing the supernatant after each. Finally, the tubes were reweighed and flash frozen in liquid nitrogen before storing at -80 °C.

Cell pellets were thawed on ice and re-suspended in 1 mL S30A buffer per gram cell pellet. Cell suspensions were lysed via a single pass through a French press homogenizer (Avestin; Emulsiflex-C3) at 15000-20000 psi and then centrifuged at 12000 x g at 4 °C for 30 minutes to separate out cellular cytoplasm. After centrifugation, the supernatant was collected and incubated at 37 °C with 220 rpm shaking for 60 minutes to digest remaining mRNA with endogenous nucleases [68]. Subsequently, the extract was re-centrifuged at 12000 x g at 4 °C for 30 minutes, and the supernatant was transferred to 12-14 kDa MWCO dialysis tubing (Spectrum Labs; Spectra/Por4) and dialyzed against 2 L of S30B buffer (14 mM Mg-glutamate, 60 mM K-glutamate, $\simeq$ 5 mM Tris, pH 8.2) overnight at 4 °C. The following day, the extract was re-centrifuged at 12000 x g at 4 °C for 30 minutes. The supernatant was optionally concentrated using a 10,000 MWCO centrifuge column (GE Healthcare; Vivaspin20) based on total protein levels from a Bradford assay (ThermoScientific) to obtain concentrations above 15 mg $\cdot$ mL^{-1}, aliquoted, and flash frozen in liquid nitrogen before storage at -80 °CC.

9.5.3 Cell-free sensor optimization reactions

Cell-free reactions were prepared by mixing 33.3% cell extract, 41.7% buffer, and 25% plasmid DNA, any inducer, and water. Buffer composition was made such that final reaction concentrations were as follows: 1.5 mM each amino acid except leucine, 1.25 mM leucine, 50 mM HEPES, 1.5 mM ATP and GTP, 0.9 mM CTP and UTP, $0.2\,\mathrm{mg}\cdot\mathrm{mL}^{-1}$ tRNA, 0.26 mM CoA, 0.33 mM NAD, 0.75 mM cAMP, 0.068 mM folinic acid, 1 mM spermidine, 30 mM 3-PGA, and 2% PEG-8000. Additionally, the Mg-glutamate (0-6 mM), K-glutamate (20-140 mM), and DTT (0-3 mM) levels were serially calibrated for each batch of cell-extract for maximum signal. Benzoic acid, hippuric acid, and cocaine hydrochloride were purchased from Sigma-Aldrich. Permission to purchase cocaine hydrochloride was given by the French drug regulatory agency (Agence Nationale de Sécurité du Médicament et des Produits de Santé) to allow development of a new biosensor. Inducers were dissolved in ethanol and final reactions contained 0.5% ethanol for all inducer concentrations including the negative control. Reactions were prepared in PCR tubes on ice and 20 µL were transferred to a black, clear-bottom 384 well plate (ThermoScientific), sealed, and the reaction was carried out in a plate reader (Biotek; Cytation3 or Synergy HTX) to measure both endpoints and reaction kinetics. The subsequent data were processed and graphs created using custom Python scripts or Microsoft Excel. Reactions for the representative images in Figures 9.2c and 9.4b were incubated in PCR tubes at 37 °C for four hours and imaged on a UV table with either a Sony α6000 camera (benzoic and hippuric acid sensors) or a cell phone camera (cocaine sensor) and background subtracted with Adobe Photoshop.

9.5.4 Cell-free reactions with commercial beverages or human urine

Cell extract and buffer conditions were maintained from those used in optimization reactions. For the benzoic acid beverage sensor, 10% reaction volume of either 1x or 0.1x (diluted in water) of each beverage was added, in addition to 30 nM pBEAST-BenR and 100 nM pBen-sfGFP plasmids to 20 µL reactions containing extract and buffer. All beverages were purchased at a local supermarket. For the hippuric acid urine sensor, each reaction contained 10% volume of 0.1x urine, pre-diluted in water. Human urine samples were obtained from the Endocrinology Department at the University of Montpellier in accordance with ethics committee approval (#190102). Additionally, each reaction was supplemented with $0.8\,\mathrm{U}\cdot\mu\mathrm{L}^{-1}$ of murine Rnase Inhibitor (New England Biolabs).

9.5.5 Benzoic acid and hippuric acid quantification from cell-free biosensors

In order to quantify fluorescent outputs from our cell-free benzoic and hippuric acid biosensors in complex samples as a measurement of concentration, we created calibration curves by adding a range between 0 µM and 1000 µM of inducer concentrations to 20 µL cell-free reactions. Hippuric acid reactions were supplemented with $0.8\,\mathrm{U}\cdot\mu\mathrm{L}^{-1}$ RNase inhibitor to match reaction conditions. The subsequent calibration curves were fit to a Hill plot in Python using:$y = \frac{y_{max} * x^n}{K_D^n + x^n}$, where y is the fluorescence intensity, x is the inducer concentration, y_{max} is the maximum fluorescence intensity, K_D is the concentration of ligand needed for half-maximum binding occupation at equilibrium, and n is the Hill slope. Commercial beverage benzoic acid and urine hippuric acid concentrations were then calculated by using the fluorescent values from those experiments as y and solved for the inducer concentration x. Undiluted concentrations were increase by a factor of 100 to account for the 1:10 sample dilution and 10% reaction volume contribution (i.e. 2 µL sample in a 20 µL total reaction volume).

9.5.6 Chemical analysis of beverage and urine by LC-MS

The following procedure was developed for detection of benzoic and hippuric acid by UHPLC-MS / MS. The analysis was carried out using an LCMS-8050 mass spectrometer (Shimadzu, Japan) coupled to a NexeraX2 UHPLC chain (Shimadzu, Japan). The column is a Nucleodur pyramid (1.8 µm, 50 × 2.0 mm, Macherey-Nagel) maintained at 40 °C. The eluents used were:H_2O with 0.1% formic acid (A), acetonitrile with 0.1% formic acid (B). The flow rate was set to $0.5\,\mathrm{mL}\cdot\mathrm{min}^{-1}$. The injection volume was 5 µL and all the analytes were eluted over a 5 minute binary gradient with a starting composition percentage of 100/0 (A / B). The LCMS-8050 is a three-quadrupole mass spectrometer with a heated electrospray ionization (ESI) source. The analytes were detected in negative MRM mode. The samples were diluted by 20 in water before injection. Dihydrobenzoic acid was used as an internal standard.

9.5.7 Cell-free reactions detecting cocaine via luciferase output

To test our luciferase-output cocaine biosensor, 20 µL cell-free reactions containing CocE, TF, and reporter plasmid concentrations, $0.8\,\mathrm{U}\cdot\mu\mathrm{L}^{-1}$ RNase inhibitor, cocaine inducer gradient, 2 µL of undiluted human urine samples, extract and buffer were incubated at 30 °C for 8 hours. Samples were then transferred to white 96-well plates and 50 µL of Luciferase Assay Reagent (Promega) was added and mixed by manual orbital agitation. The plates were sealed and luciferase levels were mea-

sured in a plate reader two minutes after addition of the reagent. Fold change was calculated relative to the 0 µM cocaine negative control.

9.5.8 Reaction models

Coarse-grained modeling was performed using ordinary differential equations, simulated using the R software. Briefly, the model combines Michaelis-Menten kinetics for the transducer module and resource competition for RNA polymerases and ribosomes to account for varying DNA concentration effects. Michaelis-Menten equations are used for promoter activation. Production of toxic byproducts as well as energy consumption for mRNA production were also included. Full model derivation can be found in the supplementary materials, or section 9.6.2 of this thesis.

9.5.9 Chemical identifiers

In order to allow easier parsing of our article by bio-informatics tools, we provide here the identifiers of our chemical compounds:

1. Benzoic acid: InChI=1S/C7H6O2/c8-7(9)6-4-2-1-3-5-6/h1-5H,(H,8,9)
2. Hippuric acid: InChI=1S/C9H9NO3/c11-8(12)6-10-9(13)7-4-2-1-3-5-7/h1-5H,6H2,(H,10,13)(H,11,12)
3. Cocaine: InChI=1S/C17H21NO4/c1-18-12-8-9-13(18)15(17(20)21-2)14(10-12)22-16(19)11-6-4-3- 5-7-11/h3-7,12-15H,8-10H2,1-2H3/t12-,13+,14-,15+/m0/s1

9.5.10 Code availability

Simulation scripts are available on GitHub. Custom python scripts used to process data are available upon request to the authors.

9.6 Mathematical Modeling of Cell-Free Biosensors

9.6.1 Main model features description

We built a mathematical model to gain a better understanding of the behavior of our system using the metabolic transducer module. Our aim was to derive a relatively coarse-grained model that could recapitulate key behaviors observed in this dataset. The first step was to model the TF/reporter DNA assay (in Figure 9.3). We then analyzed the behaviors we wanted to reproduce in the hippurate transducer dataset, which included:

- increasing concentrations of hippurate led to increased signal
- at low HipO DNA concentrations, increasing enzyme DNA concentrations led to higher signal
- at high HipO DNA concentrations, the system reaches a peak where increasing enzyme DNA concentration leads to lower signal

Details of the full model derivation are available in Full model derivation section and scripts are available on Github . Summary of the main model features are given here:

$$
\begin{aligned}
\frac{dbenzoate}{dt} &= enz * \frac{k_{cat} * inducer}{inducer + K_M} \\
\frac{dinducer}{dt} &= -enz * \frac{k_{cat} * inducer}{inducer + K_M} \\
TF_{activated} &= TF * \frac{benzoate}{benzoate + K_d^{inducer}} + 0.0005 \\
\epsilon &= \frac{TF_{activated}}{TF_{activated} + K_d^{activated}} \text{ for BenR} \\
\epsilon &= 1 \text{ for constitutive expression} \\
\frac{dmRNA}{dt} &= \gamma * n * \epsilon \frac{x}{x + \chi} * \frac{K_{tox}}{K_{tox} + tox} * \frac{R_{mRNA}}{R_{mRNA} + K_{mRNA}} - \delta * mRNA \\
\frac{dprot}{dt} &= \pi * mRNA * \frac{y}{y + k} * \frac{K_{tox}}{K_{tox} + tox} - \lambda * prot
\end{aligned}
$$

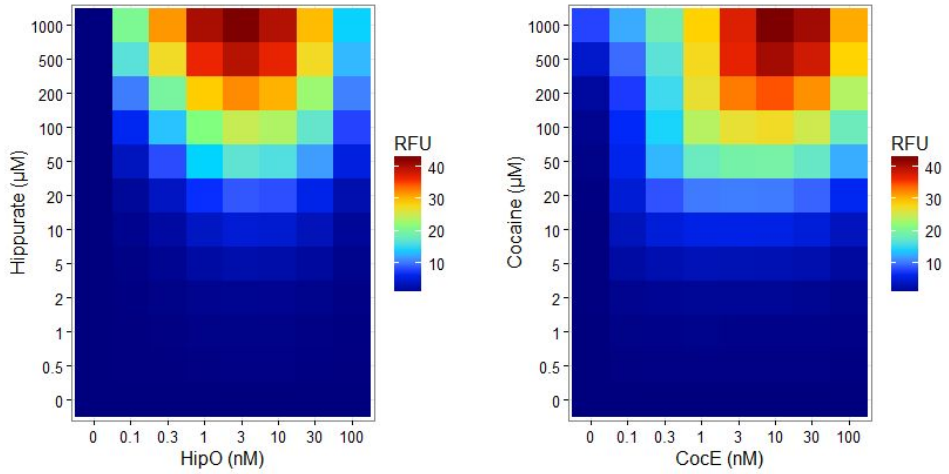
where the variables are defined as follows:

The rest of the notation is standard, with three species for mRNA and protein considered: the enzyme, the transcription factor, and the sfGFP. Spontaneous transformation is also included in the inducer production rate for cocaine.
Increasing benzoic acid leading to increased signal was expected and we modeled this using Michaelis-Menten [141] equations for the activation of the transcription

k_{cat}, K_M, enz	Enzyme Michaelis-Menten constants, enzyme concentration
TF, $TF_{activated}$	Inactivated transcription factor, transcription factor activated by benzoic acid
$K_d^{inducer}$, $K_d^{activated}$	Hill activation constant for the TF activation by benzoic acid/ promoter activation by TF
ϵ	Fraction of activated promoter for induced or constitutive promoters
γ, π	mRNA and protein production rates
χ, k	Affinity of the RNAP/ribosome for the promoter/RBS
x, y	Free RNAP and ribosome
tox, R_{mRNA}	Accumulated toxic by-product, available resources for mRNA production

factor and of the promoter. The fact that signal was low at low TF DNA concentration and increased with increasing TF DNA concentration meant that increasing enzyme concentration led to increased signal, which would not happen if all reactions were catalyzed on very fast time scales (i.e. the enzyme concentration would not matter). We therefore had to include enzyme kinetics in our model. At high DNA concentrations, resource competition effects meant that too many resources were diverted towards enzyme production instead of GFP production, which led to a decrease in signal. We also decided, as we know these effects exist in cell-free systems, to include resource depletion and production of toxic byproducts that would inhibit reactions in our model. For enzyme kinetics, we used the Michaelis-Menten equation [141] with parameters obtained from BRENDA, whereas we used the framework developed by [154] for modeling resource competition, based on competition between DNA and mRNA for RNAP and ribosomes, respectively. More details on the methods employed, as well as a full model derivation, are presented in the full model derivation section.
The results obtained for HipO-hippurate heat-map are presented in Figure 9.6. No parameter fitting was performed, and minimal parameter tuning was involved, as most parameters were taken from or derived from the literature. Constants linked to resource depletion or toxic byproduct production were manually chosen so as to best capture the data, as well as ribosome or RNAP quantity. This, however, only quantitatively changed the data, but did not change the data qualitatively when parameters remained in a realistic range. Therefore, we managed to qualitatively reproduce the three effects we wanted to account for with this model, supporting our hypothesis regarding the main factors underpinning the biological effects in our HipO data.

Next, we decided to apply our model to the CocE data. We changed the enzyme kinetic parameters, as well as transcription and translation rates linked to the length of the gene; however, this failed to reproduce our experimental data, as significant signal was obtained for CocE DNA = 0.1 nM (data was very similar to HipO, despite the above-mentioned parameter changes, results not shown). We

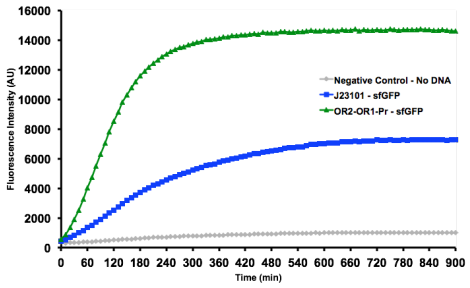


(a) (b)

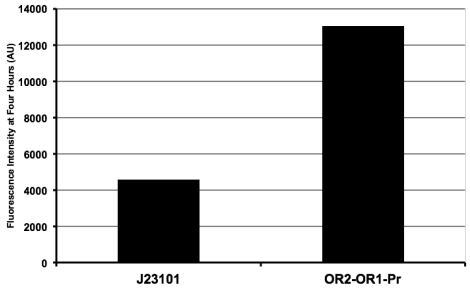
Figure 9.6 **Modeling metabolic transducer behavior for HipO and CocE.** Hippurate or cocaine can be detected using different metabolic tranducers. Conditions for inducer and DNA concentrations used in Figure 9.4 were modeled using ordinary differential equations to capture qualitative trends in the data. Simulations were rescaled to use the same scale as data. The heat-map represents GFP model signal after four hours.

hypothesized that this was because the CocE promoter was weaker (∼3x at four hours, 9.7). This shifted the peak but significant signal was still obtained for CocE DNA = 0.1 nM. However, thanks to the model, we postulated another cause due to a weaker translation initiation rate, as we were using different RBSs for the two enzymes. Using the RBS calculator [450], which takes context into account, we found that CocE translation initiation rate was predicted to be much slower than HipO initiation rate, which we transcribed in our model as a weaker affinity of the RBS for ribosomes. Results obtained through this strategy are presented in 9.6.

Using this RBS affinity change and the changed promoter strength, we managed to capture two of the three differences in the HipO and CocE datasets: signal for low CocE value starts at higher enzyme DNA concentrations (which we attribute to lower enzyme production due to a weaker promoter and putatively weaker RBS); and signal at 100 nM is higher as there are fewer resources diverted into unnecessary enzyme production (or less toxicity and resource exhaustion by unnecessary enzymes). However, we do not capture quantitative values, which could be due to the fact that measurements were performed in a different set-up or that another component our model is lacking. Moreover, the CocE experiment was performed at 30 °C as it is the optimal temperature for this enzyme. Our modeling assumption was that this impacted only kinetic parameters, which is therefore included in our model. However, it might also affect the benzoic acid reporter which the model does not account for.
This shows that with our model, changing only parameters linked to the new enzyme sequence, we accurately captured the differences we aimed to capture in the two setups. Therefore, our model, without any parameter fitting and minimal



(a)



(b)

Figure 9.7 Superfolder-GFP expression with J23101 and pBEST promoter (OR2-OR1-Pr). Expression levels of J23101 and OR2-OR1-Pr promoters were compared in a cell-free reaction to provide comparative strength data for our computer model. Reactions were conducted at $6.5\,\mathrm{ng}\cdot\mu\mathrm{L}^{-1}$ at $37\,^{\circ}\mathrm{C}$ for fifteen hours and data at the four hour time point showed that J23101 is approximately three times weaker than OR2-OR1-Pr in our cell-free system.

parameter tuning within reasonable ranges, achieves satisfying qualitative reproduction of our data. Despite these successes, our model has limitations.

We can see that our model does not adequately capture the resource competition or exhaustion at enzyme concentration of 100 nM (although there is indeed no signal in our model if we increase the concentration of the simulated DNA to 300 nM, results not shown). To correct this limitation, including more resource exhaustion could be the answer. Moreover, although we only tried to qualitatively capture the data, the ease of explanation of CocE data after preliminary work on HipO only led us to suggest improvements that could be made to explain the data quantitatively: including GFP maturation kinetics to become fluorescent, as well as including parameters from the plate reader. However, complete quantitative modeling seems unrealistic on cell-free systems based on extracts rather than individual components, as a number of parameters still vary from batch to batch and will therefore hardly be realistically estimated for predictive modeling of the time course of the data produced on those setups without complementary experiments on each batch to determine batch-dependent relevant parameters. Qualitative predictions seem more relevant in that type of set-up at the moment. Moreover, as long as no definite hypothesis emerges as to why cell-free systems stop functioning (amino acid or nucleotide depletion, energy depletion, toxic byproduct accumulation or any other, as well as any combination of those hypotheses), different models encompassing these hypotheses will be derived mathematically, and capture some effects in the

data, but no definite answer on what modeling strategy is the best can be found before this question is experimentally answered.

9.6.2 Model Prediction Experimental Demonstration

In order to demonstrate that the predictions made by our model were trustworthy, and to test how altering the optimal TF/reporter DNA concentrations determined in the benzoic acid sensor affects the metabolic hybrid sensors, we designed a simple experimental verification. The model predicted that increasing the TF DNA concentration from our optimized concentration (30 nM) to another concentration that also gave good fold change from our initial TF reporter DNA assay (100 nM) would result in a shift of the dose-response curve of fluorescence to high transducer DNA concentration. Indeed, the unnecessary resources consumed to increase TF production would be diverted from the enzyme production that is necessary for efficient conversion of the inducer to benzoic acid. This effect is competing with the increased signal that could come from having higher TF levels, but the model predicts it to be the dominant effect, which was experimentally demonstrated using 1000 µM hippuric acid and varying the HipO concentration in two set ups, with TF concentrations either at 30 nM or 100 nM, while keeping the reporter concentration at 100 nM (**Supplementary Figure 5: add the figure here**). This verification leads us to have greater confidence in model predictions on effects linked to resource competition.

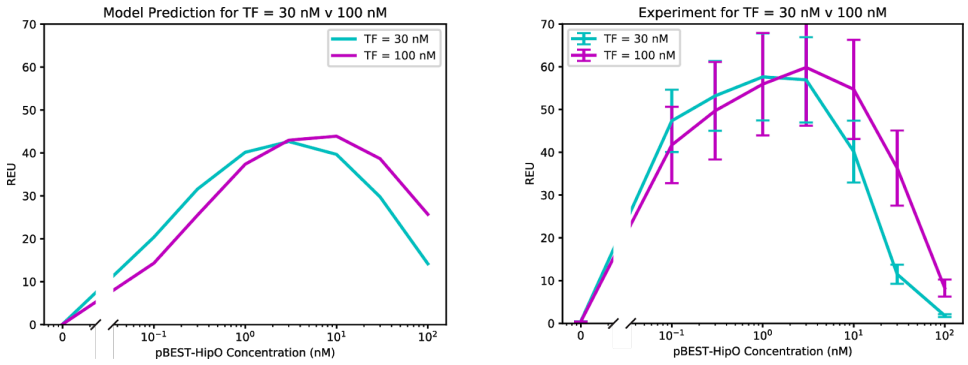


Figure 9.8 **Model-predicted shift in HipO concentration for peak biosensor signal at high concentrations of TF plasmid and inducer.** Increasing TF plasmid concentration results in a right-shift of HipO plasmid concentration for optimal performance. Left panel: Model calculations for sfGFP output for a range of pBEST-HipO concentrations for TF plasmid concentrations at 30 nM and 100 nM. Right panel: Experimental results to examine if the same right-shift could be seen experimentally. Results are the mean from three experiments on three different days and error bars represent the standard deviation. For all experiments and model calculations, reporter plasmid concentration was fixed at 100 nM and a hippurate inducer concentration of 1000 µM was used. All fluorescence values have relative expression unit (REU) compared to the four hour level for 100 pM of a strong, constitutive sfGFP-producing plasmid.

This text was originally in the supplementary information of the paper Plug-and-Play Metabolic Transducers Expand the Chemical Detection Space of Cell-Free Biosensors published in Nature Communications.

9.7 Mathematical model derivation

We will base our time model on classical models of transcription and translation and Michaelis-Menten kinetics [141]. Resource competition is mostly inspired from [154], except used at each time step instead of at steady-state. Resource exhaustion accounts for energy depletion and byproducts secretion. We will first present our assumptions and then expose the model as such.

9.7.1 Hypothesis

We will make the following assumptions:
Equilibrium of fast processes compared to transcriptional and translational elongations:

- Binding and unbinding of RNAP to DNA is on a much faster scale than elongation so considered at equilibrium
- Binding and unbinding of the transcription factor to DNA is on a much faster scale than elongation so considered at equilibrium
- Binding and unbinding of the inducer to the transcription factor is on a much faster scale than elongation so considered at equilibrium
- Binding and unbinding of ribosomes to mRNA is on a much faster scale than elongation so considered at equilibrium

Steady flow of production
We will consider the flow of RNAP and ribosomes to be at steady state. That is, we will neglect the first minutes of elongation at the start of the process before steady-state flow of production, consider that the production rate is constant and use effective production rates as explained in the subsection. Elongation itself is fast but RNAP and ribosomes are shared between processes and therefore modeling elongation and its impact on the available RNAP and ribosomes is key to modeling resource competition.

Using the same framework as [154] for modeling resource competition, we will therefore also adopt their notations. For the sake of the reader's best understanding, we will nonetheless fully derive the model of resource competition before making further simplifications, as well as presenting our accounting of resource exhaustion and enzyme kinetics that are absent from their model.

9.7.2 Derivation of the resource competition model

The circuits described will consist of two types of modules: constitutively expressed ones (enzymes and the BenR transcription factor) and inducible ones (GFP), induced upon the binding by the active transcription factor u (benzoic acid/BenR complex). The promoter complex b_{ind} is formed by u binding to the empty promoter b^*_{ind} of the gene encoding the protein p_{ind} (that appears in the translation derivation). The binding of the available RNAP x can therefore form the active transcriptional complex c_{ind}, producing the mRNA m_{ind}, encoding p_{ind} at a rate γ_{ind} (encompassing all elongation reactions and accounting for the global translation rate). This mRNA decays at a rate δ_{ind}, and all these processes encompassing transcription steps are exemplified below:

$$u+b^*_{ind} \underset{\mu^-}{\overset{\mu^+}{\rightleftharpoons}} b_{ind}, \quad b_{ind}+x \underset{\kappa^-}{\overset{\kappa^+}{\rightleftharpoons}} c_{ind}, \quad c_{ind} \xrightarrow{\gamma_{ind}} b_{ind}+x+m_{ind} \quad and \quad m_{ind} \xrightarrow{\delta_{ind}} \emptyset$$

For the constitutive expression, the model is simpler and is summarized by the following reactions:

$$b_{const} + x \underset{\kappa^-}{\overset{\kappa^+}{\rightleftharpoons}} c_{const}, \quad c_{const} \xrightarrow{\gamma_{const}} b_{const} + x + m_{const} \quad and \quad m_{const} \xrightarrow{\delta_{const}} \emptyset$$

The translation processes are identical for constitutive and inducible promoters, initiated by the binding of the ribosome y to the ribosome binding site (RBS) of the mRNA m, forming the transitionally active complex d. We consider that bound mRNA fragments cannot be degraded by RNases. Protein p is produced at a rate π encompassing elongation and production, and is degraded at a rate λ. The translation reactions are therefore:

$$m + y \underset{k^-}{\overset{k^+}{\rightleftharpoons}} d, \quad d \xrightarrow{\pi} m + y + p \quad and \quad p \xrightarrow{\lambda} \emptyset$$

The corresponding ODE system is given by equations (9.1) for induced proteins:

$$\begin{aligned}
\frac{db_{ind}}{dt} &= (\mu^+ u b^*_{ind} - \mu^- b_{ind}) - (\kappa^+ x b_{ind} - \kappa^- c_{ind}) + \gamma c_{ind} \\
\frac{dc_{ind}}{dt} &= (\kappa^+ x b_{ind} - \kappa^- c_{ind}) - \gamma c_{ind} \\
\frac{dm_{ind}}{dt} &= \gamma c_{ind} - \delta m_{ind} - (k^+ y m_{ind} - k^- d_{ind}) + \pi d_{ind} \\
\frac{dd_{ind}}{dt} &= (k^+ y m_{ind} - k^- d_{ind}) - \pi d_{ind} \\
\frac{dp_{ind}}{dt} &= \pi d_{ind} - \lambda p_{ind}
\end{aligned} \tag{9.1}$$

and by the following equations (9.2) for constitutive ones:

$$\begin{aligned}
\frac{dc_{const}}{dt} &= (\kappa^+ x b_{const} - \kappa^- c_{const}) - \gamma c_{const} \\
\frac{dm_{const}}{dt} &= \gamma c_{const} - \delta m_{const} - (k^+ y m_{const} - k^- d_{const}) + \pi d_{const} \\
\frac{dd_{const}}{dt} &= (k^+ y m_{const} - k^- d_{const}) - \pi d_{const} \\
\frac{dp_{const}}{dt} &= \pi d_{const} - \lambda p_{const}
\end{aligned} \tag{9.2}$$

9.7.3 RNAP and ribosome demands

Notations

We assume DNA concentration n_i is constant for each species. We will introduce some notations that will allow us to simplify our problem given the assumptions presented in 9.7.1

$$\kappa_i = \frac{\kappa_i^- + \gamma_i}{\kappa_i^+}, \quad k_i = \frac{k_i^- + \pi_i}{k_i^+}, \quad and \quad h_i = \frac{\gamma_i n_i}{\delta_i}$$

We also introduce $\mu = \frac{\mu^-}{\mu^+}$, and

$$\epsilon = \frac{\frac{u}{\mu}(1 + \frac{x}{\kappa})}{1 + \frac{u}{\mu}(1 + \frac{x}{\kappa})}$$

As was done in [154], we use ϵ to describe the fraction of induced promoters for our inducible gene.

For our BenR biosensor modeling, we used

$$TF^{activated} = TF * \frac{inducer}{inducer + K_d^{inducer}} + 0.0005$$

$$\epsilon = \frac{TF^{activated}}{TF^{activated} + K_d^{activated} \times 1000}$$

The first equation represents transcription factor activation by the inducer, including some leaky activation, while the second equation represents the activation of the promoter by the activated transcription factor.

Simplification and resolution

Using the assumptions presented in 9.7.1, we can consider that

$$\begin{aligned} \frac{db}{dt} &= 0 \\ \frac{dc}{dt} &= 0 \\ \frac{dd}{dt} &= 0 \end{aligned} \tag{9.3}$$

This allows us to neglect binding events and consider the system to be at equilibrium for binding/unbinding events on time scales inferior to the production and degradation of mRNA and proteins. Therefore, the RNAP and ribosomes are always split between genes and mRNAs and can be solved using the same technique as in [154], considering resource conservation.

Using $\frac{dc}{dt} = 0$, and then $\frac{db}{dt} = 0$ we obtain:

$$\begin{aligned} \frac{dc}{dt} = 0 &\Leftrightarrow (\kappa^+ xb - \kappa^- c) - \gamma c = 0 \\ &\Leftrightarrow c = \frac{xb\kappa^+}{\kappa^+ \gamma} \\ &\Leftrightarrow c = \frac{xb}{\kappa} \end{aligned} \tag{9.4}$$

$$\begin{aligned} \frac{db}{dt} = 0 &\Leftrightarrow (\mu^+ ub^* - \mu^- b) - \frac{dc}{dt} = 0 \\ &\Leftrightarrow (\mu^+ ub^* - \mu^- b) = 0 \\ b^* &= \frac{\mu^- b}{\mu^+ u} \end{aligned} \tag{9.5}$$

Using DNA conservation, i.e.: $n = c + b + b^*$, we have:

$$
\begin{aligned}
n &= c + b + b^* \\
&= \frac{xb}{\kappa} + b + \frac{\mu^- b}{\mu^+ u} \\
&= (\frac{x}{\kappa} + 1 + \frac{\mu^-}{\mu^+ u})b \\
b &= \frac{n}{\frac{x}{\kappa} + 1 + \frac{\mu^-}{\mu^+ u}} \\
c &= \frac{xb}{\kappa} \\
&= \frac{x}{\kappa} \quad \frac{n}{\frac{x}{\kappa} + 1 + \frac{\mu^-}{\mu^+ u}} \\
&= \frac{xn}{\kappa} \quad \frac{1}{\frac{x}{\kappa} + 1 + \frac{\mu}{u}} \\
&= \frac{xn}{\kappa} \quad \frac{u}{\mu} \quad \frac{1}{1 + \frac{u}{\mu}(1 + \frac{x}{\kappa})} \\
&= n\frac{x}{\kappa} \quad \frac{x+\kappa}{x+\kappa} \quad \frac{u}{\mu} \frac{1}{1 + \frac{u}{\mu}(1 + \frac{x}{\kappa})} \\
&= n\frac{x}{x+\kappa} \quad \frac{x+\kappa}{\kappa} \quad \frac{u}{\mu} \frac{1}{1 + \frac{u}{\mu}(1 + \frac{x}{\kappa})} \\
&= n\frac{x}{x+\kappa} \quad \frac{u}{\mu} \quad \frac{\frac{x+\kappa}{\kappa}}{1 + \frac{u}{\mu}(1 + \frac{x}{\kappa})} \\
&= n\frac{x}{x+\kappa}\epsilon
\end{aligned}
\tag{9.6}
$$

Therefore,

$\frac{dm}{dt} = \gamma n \epsilon \frac{x}{x+\kappa} - \delta m$

For constitutive expression, derivation is much simpler, and we easily obtain $c = n\frac{x}{x+\kappa}$, or $c = n\epsilon\frac{x}{x+\kappa}$ with $\epsilon = 1$, considering all promoters are active. Using the same strategy, considering $\frac{dd}{dt} = 0$, we obtain $d = \frac{y m_f}{k}$, where m_f is the free mRNA. Considering that mRNA production and degradation is constant on the time scale of ribosome binding, and that the total amount of mRNA is m (both bound and unbound, the product of transcription from the previous steps), applying the same derivation to $m = m_f + d$ instead of $n = c + b$ leads to $d = m\frac{y}{y+k}$, and

$$\frac{dp}{dt} = \pi m \frac{y}{y+k} - \lambda p$$

Our time evolution model is therefore

$$\frac{dm}{dt} = \gamma n \epsilon \frac{x}{x+\kappa} - \delta m$$

$$\frac{dp}{dt} = \pi m \frac{y}{y+k} - \lambda p$$

.

Repartition between genes and mRNAs

This model allows us to account for resource competition by calculating the repartition of ribosomes and RNAP among different processes at each time step.

The explanation will be done for RNAP (x) and is similar for ribosomes (y). We consider the conservation law for RNAP:

$$X_{tot} = x + c_{GFP} + c_{enz} + c_{BenR}$$

. We look for an integer x minimizing the error so that

$$X_{tot} \simeq x + \epsilon \times n_{GFP} \frac{x}{x+\kappa_{GFP}} + n_{enz} \frac{x}{x+\kappa_{enz}} + n_{BenR} \frac{x}{x+\kappa_{BenR}}$$

which is the optimal RNAP repartition at this time step.

9.7.4 Accounting for resource depletion and toxicity

We decided to account for the exhaustion of the cell-free system in two different ways. First, we consider that there are a limited number of mRNAs that can be produced due to limited nucleotides supply or energy. This is done by multiplying transcription rates by $\frac{resources}{resources+K_d^{resource}}$ We do not consider a limit on amino acids as they are supplemented in the cell-free system, and mRNA production has been shown to stop rapidly in cell-free systems. Each mRNA produced consumes its length in nucleotides. Moreover, we consider that producing proteins also accumulates toxic byproducts, which slow down reactions for both translation and transcription, by multiplying transcription and translation rates by a function of the form $\frac{K_d^{tox}}{K_d^{tox}+ToxicProduct}$. We consider that each produced protein contributes to this effect, rather than amino acids, as we consider toxicity to be due to the fully formed proteins producing by-products or slowing down the extract. Our aim is to reproduce the exhaustion effect qualitatively.

9.7.5 Enzymatic steps

For modeling enzymatic steps, i.e. the conversion of the inducer (either cocaine or hippurate) into benzoic acid, we use Michaelis-Menten kinetics [141]:

$$rate = enzyme * k_{cat} \frac{substrate}{substrate + K_M}$$

9.7.6 Typical range of biochemical parameters

Considerations on cell-free systems

The experimental set-up (cell-free system) allows us to consider nominal DNA concentration values instead of having to consider plasmid copy number as would have to be done *in vivo*. Moreover, the rates will be derived here for *in vivo* systems and will be divided by 10 for simulations, as reactions have been shown to be slower in cell-free compared to *in vivo* [414] and an order of magnitude of difference is suggested in [411] and [416]. Final parameters used for numerical simulations can be found in Table 9.5.

Production rates

We will derive all rates as if it were *in vivo* and divide them by 10 for cell-free modeling.

According to [451], the mRNA chain elongation rate is ≈ 50 nucleotides per sec. The mRNA production rate γ in minutes is therefore $\gamma_{protein} = \frac{50}{length_{protein}} * 60$. Moreover, the peptide chain elongation rate is ≈ 15 amino acids per sec, which means the protein production rate π in minutes is therefore $\pi_{protein} = \frac{15}{length_{protein}} * 60$.

Protein name	**Length in nucleotides**	**Length in amino acids**	γ	π
GFP	720	240	4.2 / min	3.75 / min
BenR	954	318	3.35 / min	2.83 / min
HipO	1200	400	2.5 / min	2.25 / min
CocE	1700	560	1.76 / min	1.61 / min

Table 9.1 *In vivo* transcription and translation rates.

Degradation rates

Since the mRNA half-life is measured to be about 15 minutes in cell-free systems [410], we use $\delta = 0.05$ / min. For in vivo systems, mRNA half-life is shorter, around 4 min, so we use $\delta = 0.2$ / min.

The protein half-life is approximately 1 hour *in vivo* [452]. As our system is purified from extract, we consider that proteases are still present and we use $\lambda = 0.0016$ / min (*in vivo* rate divided by 10). Changing it affects time evolution but not the effect of DNA and inducer concentrations at 240 min that were studied in this article (results not shown).

Transcription and translation rates

According to [453], there can be a transcription initiation every 5 seconds on a DNA strand. Using the fact that the mRNA chain elongation rate is ≈ 50 nucleotides per sec, there are, on the same DNA, at most ω RNAP, with $\omega = round(\frac{length_{protein}}{50*5})+1$. We will rather consider the genes to be present in $\omega * n$ numbers and being able to recruit only 1 RNAP.

In the same manner, we have to account for the fact that multiple ribosomes can be translating the same mRNA strand, but we will assume the average distance between ribosomes to be around 80 nucleotides. We then have at most χ ribosomes on a strand, where $\chi = round(\frac{length_{protein}}{80}) + 1$, and we will consider mRNA to be able to bind a single ribosome, with an effective protein production rate of $\chi * \pi$ for each mRNA.

Protein name	**Length in nucleotides**	ω	χ
GFP	720	4	10
BenR	954	5	13
HipO	1200	6	16
CocE	1700	8	22

Table 9.2 Number of RNAP/ ribosomes per DNA/ mRNA strand

Protein name	χ	π	**Effective *in vivo* π**	**Effective cell-free π**
GFP	10	3.75	37.5	3.75
BenR	13	2.83	36.79	3.679
HipO	16	2.25	36	3.6
CocE	22	1.61	35.42	3.542

Table 9.3 Effective translation rates *in vivo* and in cell-free

Enzymes' catalytic constants

For the two enzymes considered, CocE and HipO, the values used from BRENDA are listed in Table 9.4 [454]. The exact values in our cell-free system may differ from the values in BRENDA as these are often measured *in vitro* and vary according to the organism the enzyme is taken from and the organism or cell-free extract it is expressed in. However, we believe they should still be accurate within an order of magnitude and we expect small changes to have minimal effect on simulation end results due to their fast kinetics related to the other system components. Moreover, despite their possible disadvantages, we prefer using literature values when possible so as to leave a minimum number of parameters free.

Protein name	$k_{cat}, 1/min$	K_M in µM
HipO	5880	764
CocE	3060	5.7

Table 9.4 Enzymes' catalytic constants.

Handling of RBS and DNA binding

Using the same order of magnitude for RNAP binding constants as [154], we used: $\kappa_{GFP} = 100\,\text{nM}$; $\kappa_{HipO} = \kappa_{BenR} = 3000\,\text{nM}$, as these are expressed constitutively under the same promoter. Since CocE was on a promoter that was weaker than that of HipO (See Supplementary Figure 3), we used $\kappa_{CocE} = 4500\,\text{nM}$.

Following the same reasoning, we use $k_{GFP} = 1\,\mu\text{M}$, and $k_{BenR} = k_{HipO} = 10\,\mu\text{M}$. Moreover, using the RBS calculator [455], we found that using gene context and the RBS, initiation of CocE is slower than initiation of HipO. Knowing that the RBS calculator is more trustworthy for trends than qualitative values, we implemented that using $k_{CocE} = 30\,\mu\text{M}$, i.e.: less efficient in binding ribosomes, since initial elongation rate does not appear in our modeling framework. This value was chosen at it recapitulates our data well.

9.7.7 Numerical simulations

Parameters

Parameters used for the final simulations are presented in Table 9.5. A constant value of 0.05 is added to account for background on all data points.

9.7.8 Computational methods

Software tools

All scripts were done in R (version 3.2.3, [388]), using RStudio as an integrated development environment (version 0.99.903, [456]). The ODE solver used is ode from de deSolve package (version 1.14, [389]). For visualization, packages reshape2 [457] and ggplot2 [458] are used.

Availability

Scripts are available on Github at `https://github.com/brsynth`.

Parameter	Value	Unit
κ_{GFP}	100	nM
κ_{BenR}	3000	nM
κ_{HipO}	3000	nM
κ_{CocE}	4500	nM
γ_{GFP}	0.42	min^{-1}
γ_{BenR}	0.335	min^{-1}
γ_{HipO}	0.25	min^{-1}
γ_{CocE}	0.176	min^{-1}
k_{GFP}	1	μM
k_{BenR}	10	μM
k_{HipO}	10	μM
k_{CocE}	30	μM
π_{GFP}	3.75	min^{-1}
π_{BenR}	3.679	min^{-1}
π_{HipO}	3.6	min^{-1}
π_{CocE}	3.542	min^{-1}
$length_{GFP}^{mRNA}$	720	nucleotides
$length_{BenR}^{mRNA}$	954	nucleotides
$length_{HipO}^{mRNA}$	1200	nucleotides
$length_{CocE}^{mRNA}$	1700	nucleotides
k_{cat}^{HipO}	5880	min^{-1}
k_{cat}^{CocE}	3060	min^{-1}
k_{M}^{HipO}	764	mM
k_{M}^{CocE}	5.7	mM
$Spontaneoushydrolisation^{HipO}$	0	μM
$Spontaneoushydrolisation^{CocE}$	0.0001	μM
ω_{GFP}	4	No unit
ω_{BenR}	5	No unit
ω_{HipO}	6	No unit
ω_{CocE}	8	No unit
$K_d^{inducer}$	100	μM
$K_d^{activated}$	50	μM
δ	0.05	min^{-1}
λ	0.0016	min^{-1}
n_{GFP}	100	nM
n_{BenR}	30	nM
X	30	nM
Y	30	nM
K_d^{tox}	100	nM
K_d^{mRNA}	10	nucleotides
$Initial^{resource}$	10000000	nucleotides

Table 9.5 Numerical parameters used during simulations

9.7.9 Supplementary Figures: time courses and real world application

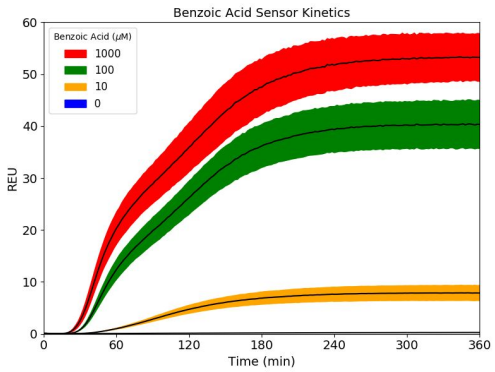


Figure 9.9 **Time course of the benzoic acid biosensor response to varying concentrations of inducer.** Kinetics of optimized benzoic acid sensor at 37 °C, where the TF plasmid concentration was 30 nM and the reporter plasmid concentration was 100 nM. Data are the average, with standard deviation, of three technical repeats from three experiments performed on three different days and all fluorescence values have REU compared to the four hour level for 100 pM of a strong, constitutive sfGFP-producing plasmid. Fold change measurements were taken from the four hour time point.

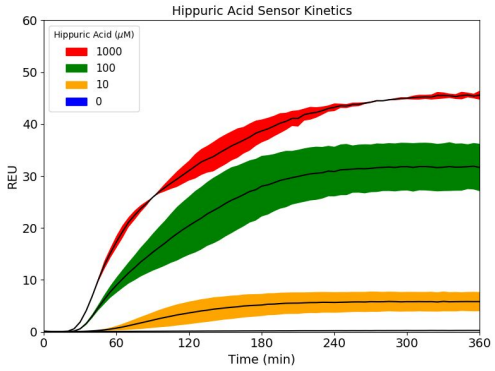


Figure 9.10 **Time course of the hippuric acid biosensor response to varying concentrations of inducer.** Kinetics of optimized hippuric acid sensor at 37 °C, where the HipO plasmid concentration was 3 nM and the TF and reporter plasmids were maintained at the same concentrations as the optimized benzoic acid sensor (30 nM and 100 nM, respectively). Data are the average, with standard deviation, of three experiments performed on three different days and all fluorescence values have REU compared to the four hour level for 100 pM of a strong, constitutive sfGFP-producing plasmid. Fold change measurements were taken from the four hour time point.

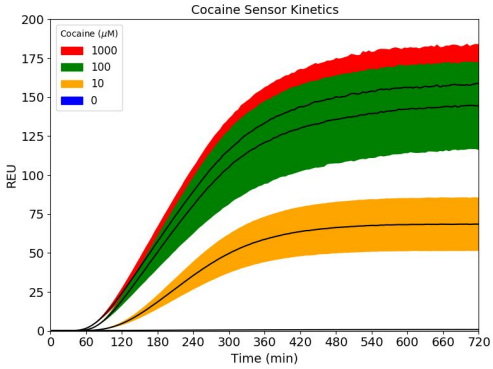
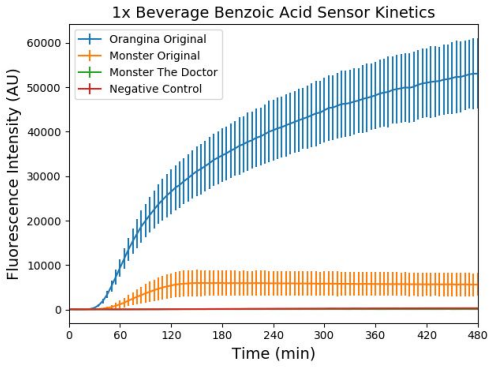
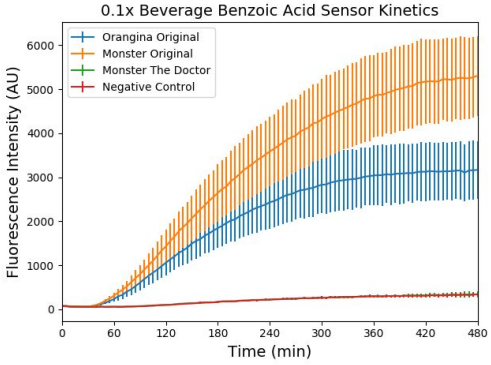


Figure 9.11 **Time course of the cocaine biosensor response to varying concentrations of inducer.** Kinetics of optimized cocaine biosensor at 30 °C, in which the CocE plasmid concentration was 10 nM and the TF and reporter plasmids were maintained at the same concentrations as the optimized benzoic acid sensor (30 nM and 100 nM, respectively). Data are the average, with standard deviation, of three experiments performed on three different days and all fluorescence values have REU compared to the four hour level for 100 pM of a strong, constitutive sfGFP-producing plasmid. Fold change measurements were taken from the four hour time point.

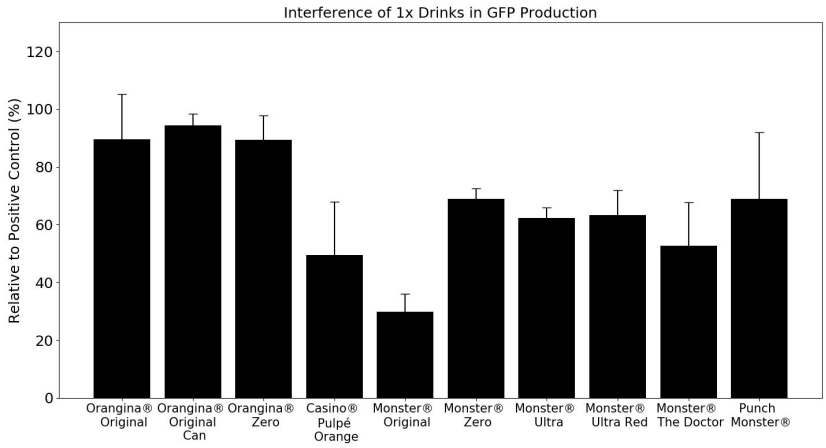


(a)

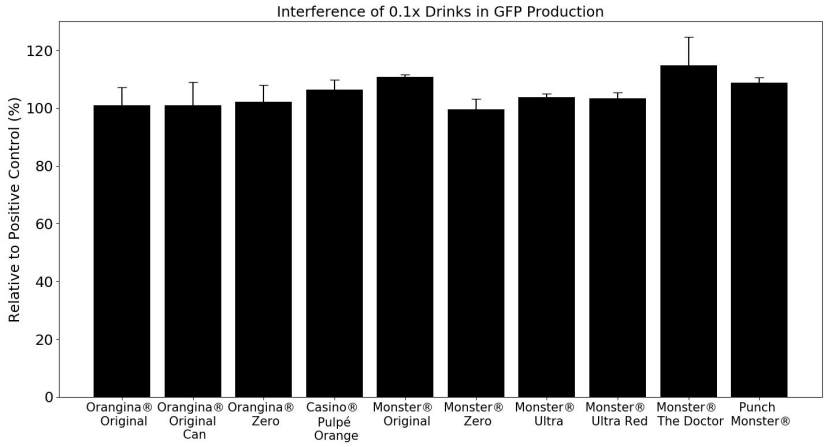


(b)

Figure 9.12 **Time course of the benzoic biosensor response to 1x and 0.1x beverages.** Kinetics of sfGFP expression at 37 °C using our optimized benzoic acid biosensor to detect benzoates in commercial beverages. The top panel depicts kinetics in response to addition of 2 µL of unaltered beverage to a 20 µL cell-free reaction. The bottom panel depicts kinetics after the samples were first diluted 1:10 in water before being added to the reaction. 'Orangina Original' and 'Monster Original' include sodium benzoate and benzoic acid, respectively, in their list of ingredients. 'Monster The Doctor' lists no benzoates in the ingredients. Water was used in place of the beverage for the negative control. Data depict the mean of three experiments conducted on three different days and error bars represent the standard deviation. Fluorescence intensity y-axis scale was adjusted for the weaker signal dilution experiment to enable adequate visualization of the kinetics.



(a)



(b)

Figure 9.13 Interference of 0.1x and 1x beverages on cell-free reaction with constitutive sfGFP plasmid. Ten-fold dilution of inducing beverage in water greatly reduces their interference in cell-free reactions. 2 µL of either 1x (top panel) or 0.1x (bottom panel) beverages were added to 20 µL cell-free reactions containing 10 nM of the strong constitutive GFP plasmid pBEAST-sfGFP. Fluorescence intensities at four hours were normalized to a negative control containing water instead of the commercial beverage. Data are mean values from three experiments on three different days and error bars represent the standard deviation.

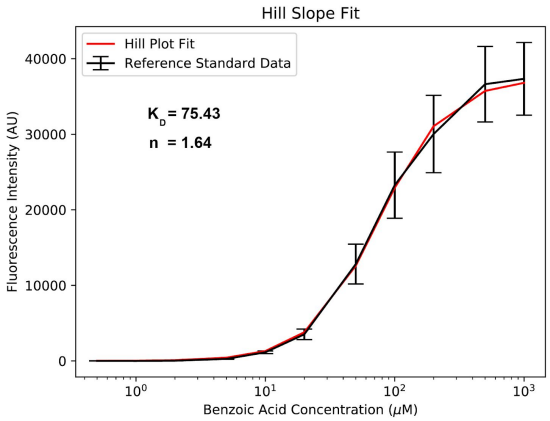


Figure 9.14 **Hill plot fit of a standard gradient of benzoic acid to calibrate sensor.** A standard gradient of benzoic acid concentration was added to our optimized benzoic acid sensor at 37 °C for four hours. The fluorescence intensity values were fit to a Hill plot function in order to convert fluorescence measurements of benzoates in beverages into sample concentration. The data are the mean of three experiments on three different days and error bars represent the standard deviation.

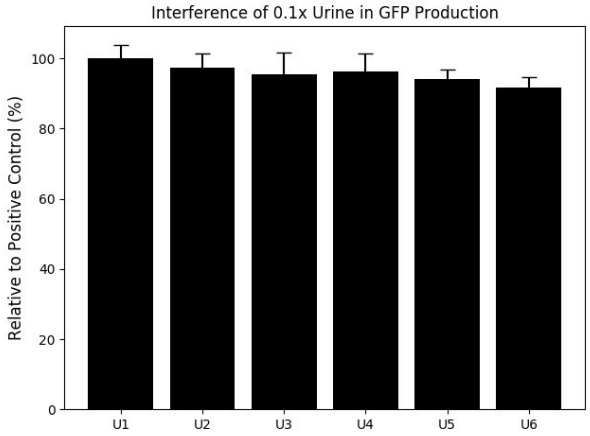


Figure 9.15 **Interference of human urine on cell-free reaction with constitutive sfGFP plasmid.** Ten-fold dilution in urine in the presence of an RNase inhibitor minimizes interference of human urine on cell-free production. Urine samples from six patients (U1-U6) were diluted 1:10 in water and 2 µL were added to20 µL cell-free reactions (1 % final concentration) containing 10 nM of the strong constitutive GFP plasmid pBEAST-sfGFP and 0.8 U · µL^{-1} of a murine RNase inhibitor. Fluorescence intensities at four hours were normalized to a negative control containing water instead of urine. Data are mean values from three experiments on three different days and error bars represent the standard deviation.

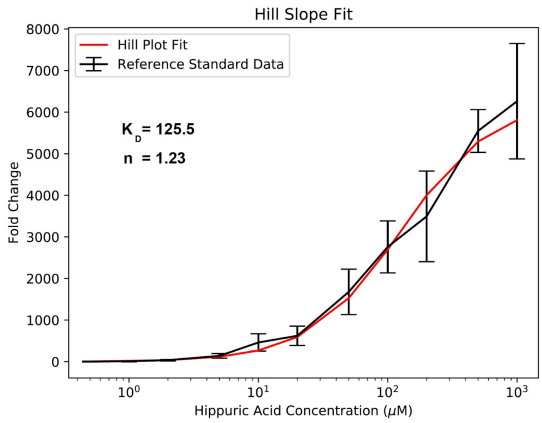


Figure 9.16 **Hill plot fit of a standard gradient of hippuric acid to calibrate sensor.** A standard gradient of hippuric acid concentration was added to our optimized hippuric acid sensor with $0.8\,\mathrm{U}\cdot\mu\mathrm{L}^{-1}$ of a murine RNase inhibitor at 37 °C for four hours. The fluorescence intensity values were fit to a Hill plot function in order to convert fluorescence measurements of hippuric acid in urine samples into sample concentration. The data are the mean of three experiments on three different days and error bars represent the standard deviation.

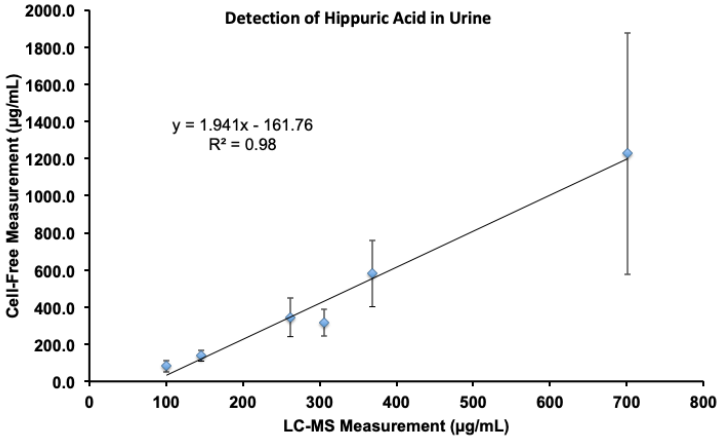


Figure 9.17 **Correlation between cell-free biosensor and LC-MS measurements of endogenous hippuric acid levels in human urine.** Quantified cell-free biosensor values of hippuric acid measurement were determined using a Hill plot fit to our standard curve (9.16) and cell-free data are the mean of three experiments on three different days (error bars represent standard deviation). LC-MS measurements are from a single measurement. R^2 value was calculated by a linear regression fit.

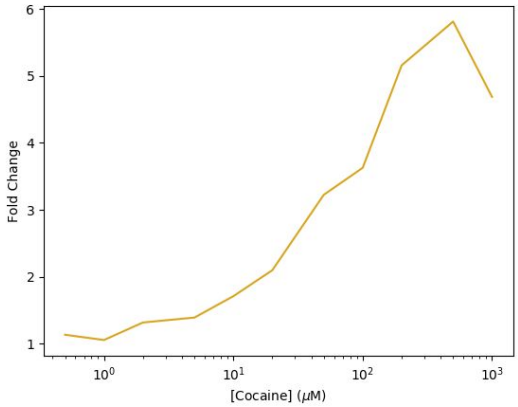


Figure 9.18 Detection of cocaine spiked into clinical urine samples with sfGFP output module. A standard gradient of cocaine hydrochloride was added with 2 µL of a human urine sample to20 µL cell-free reactions containing our optimized cocaine biosensor with $0.8\,\mathrm{U}\cdot\mu\mathrm{L}^{-1}$ of a murine RNase inhibitor and incubated at 30 °C for eight hours. Fold change was calculated relative to the 0 µM cocaine inducer. Data are from a single pilot experiment.

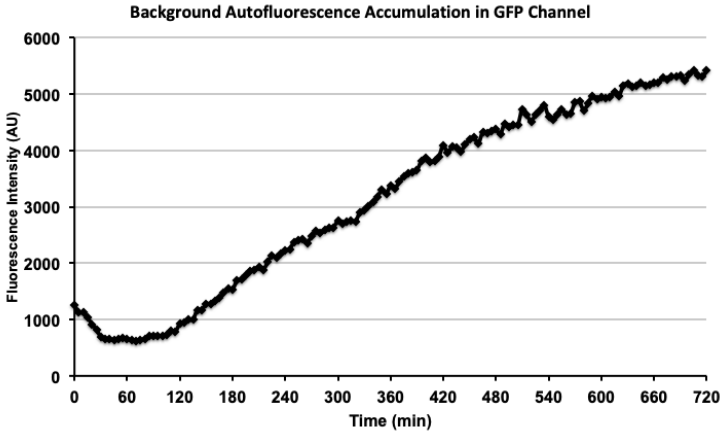


Figure 9.19 Cell-free reactions accumulate autofluorescent products in the GFP channel even in the absence of DNA. Data are from one 20 µL cell-free reaction containing only buffer, extract, and water incubated at 37 °C for 12 hours.

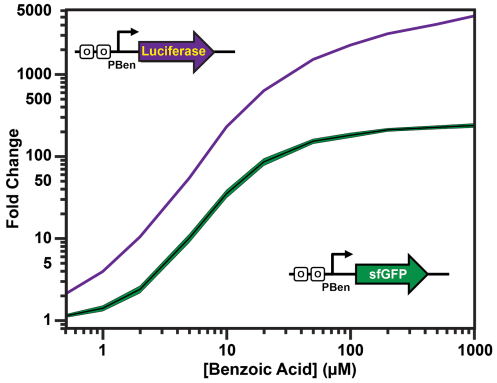
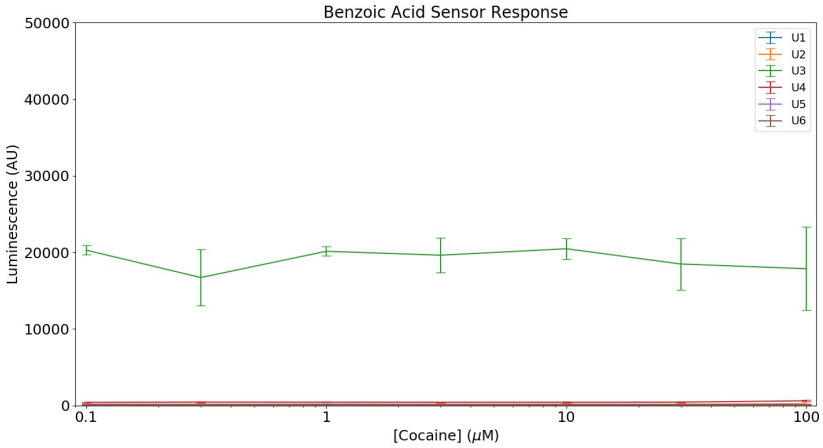
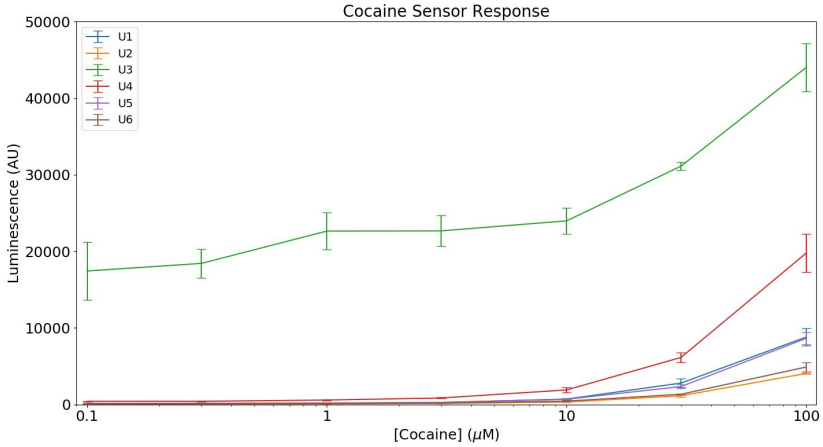


Figure 9.20 **Use of firefly luciferase as an output module enhances benzoic acid sensor fold change.** The firefly luciferase gene was cloned under the PBen promoter and added to 20 µL cell-free reactions at the same plasmid concentrations previously used with sfGFP (TF = 30 nM; Reporter = 100 nM). Reactions were incubated at 37 °C for eight hours and subsequently luciferase activity was measured on a plate reader after addition of 50 µL luciferase assay reagent. Data (purple line) was normalized to the 0 µM benzoic acid concentration and are from a single pilot experiment. Superfolder GFP curve (green line) is from FIGURE 2c and used as visual comparison.



(a)



(b)

Figure 9.21 **Comparison of benzoic acid and cocaine biosensor expression in response to urinary cocaine gradient.** A standard gradient of cocaine hydrochloride was added with 2 µL of human urine sample to 20 µL cell-free reactions containing either our optimized benzoic acid sensor or cocaine sensor with0.8 $\mathrm{U} \cdot \mu\mathrm{L}^{-1}$ RNase inhibitor as in FIGURE 4d . After incubatation at 30 °C for eight hours, the samples were transferred to white 96-well plates and 50 µL of luciferase assay reagent was added. The plates were subsequently read on a plate reader two minutes after adding the reagent and luciferase measurements in arbitrary units (AU) are shown above for both the benzoic acid sensor (top panel) and cocaine sensor (bottom panel). Data are mean values from three experiments on three different days and error bars represent the standard deviation.

9.7.10 Supplementary Tables

[Reporter Plasmid] (nM)	0 μM Benzoic Acid								
100	0.197 ± 0.023	0.200 ± 0.013	0.201 ± 0.018	0.195 ± 0.027	0.202 ± 0.027	0.200 ± 0.023	0.195 ± 0.034	0.177 ± 0.022	
30	0.197 ± 0.004	0.192 ± 0.010	0.187 ± 0.027	0.190 ± 0.018	0.181 ± 0.013	0.188 ± 0.023	0.193 ± 0.029	0.186 ± 0.018	
10	0.169 ± 0.011	0.170 ± 0.012	0.159 ± 0.015	0.163 ± 0.010	0.169 ± 0.014	0.187 ± 0.026	0.183 ± 0.026	0.189 ± 0.029	
3	0.152 ± 0.007	0.153 ± 0.012	0.151 ± 0.016	0.146 ± 0.007	0.155 ± 0.008	0.179 ± 0.024	0.194 ± 0.034	0.188 ± 0.024	
1	0.142 ± 0.005	0.144 ± 0.009	0.141 ± 0.014	0.140 ± 0.012	0.150 ± 0.010	0.167 ± 0.014	0.188 ± 0.029	0.177 ± 0.021	
0.3	0.143 ± 0.015	0.134 ± 0.008	0.145 ± 0.015	0.145 ± 0.014	0.151 ± 0.014	0.171 ± 0.021	0.187 ± 0.019	0.186 ± 0.027	
0.1	0.146 ± 0.005	0.148 ± 0.010	0.141 ± 0.010	0.137 ± 0.018	0.157 ± 0.022	0.165 ± 0.011	0.196 ± 0.030	0.179 ± 0.021	
0	0.150 ± 0.010	0.150 ± 0.015	0.143 ± 0.015	0.146 ± 0.014	0.147 ± 0.012	0.177 ± 0.023	0.197 ± 0.011	0.189 ± 0.018	
	0	0.1	0.3	1	3	10	30	100	[TF Plasmid] (nM)

[Reporter Plasmid] (nM)	10 μM Benzoic Acid								
100	0.196 ± 0.018	0.199 ± 0.014	0.198 ± 0.015	0.283 ± 0.036	0.714 ± 0.145	4.583 ± 0.839	8.034 ± 0.361	7.445 ± 0.734	
30	0.187 ± 0.007	0.185 ± 0.016	0.188 ± 0.010	0.241 ± 0.016	0.530 ± 0.124	3.114 ± 0.960	4.749 ± 0.609	4.894 ± 1.405	
10	0.174 ± 0.014	0.168 ± 0.009	0.161 ± 0.012	0.185 ± 0.022	0.322 ± 0.067	0.825 ± 0.204	1.892 ± 0.046	2.006 ± 0.310	
3	0.147 ± 0.004	0.143 ± 0.008	0.145 ± 0.007	0.153 ± 0.013	0.207 ± 0.022	0.352 ± 0.014	0.661 ± 0.047	0.826 ± 0.063	
1	0.145 ± 0.010	0.142 ± 0.014	0.135 ± 0.008	0.137 ± 0.010	0.166 ± 0.022	0.253 ± 0.036	0.335 ± 0.023	0.386 ± 0.030	
0.3	0.146 ± 0.013	0.142 ± 0.009	0.147 ± 0.015	0.138 ± 0.005	0.149 ± 0.018	0.180 ± 0.018	0.243 ± 0.011	0.247 ± 0.013	
0.1	0.144 ± 0.013	0.139 ± 0.011	0.134 ± 0.014	0.132 ± 0.014	0.144 ± 0.020	0.176 ± 0.006	0.216 ± 0.009	0.215 ± 0.019	
0	0.148 ± 0.006	0.141 ± 0.012	0.143 ± 0.019	0.143 ± 0.009	0.148 ± 0.017	0.186 ± 0.012	0.198 ± 0.018	0.205 ± 0.014	
	0	0.1	0.3	1	3	10	30	100	[TF Plasmid] (nM)

[Reporter Plasmid] (nM)	100 μM Benzoic Acid								
100	0.196 ± 0.017	0.230 ± 0.010	0.402 ± 0.029	2.128 ± 0.171	8.453 ± 1.804	23.268 ± 1.200	28.299 ± 4.737	28.584 ± 5.207	
30	0.188 ± 0.017	0.205 ± 0.010	0.373 ± 0.009	1.454 ± 0.190	6.325 ± 1.350	19.134 ± 1.013	23.251 ± 3.040	19.890 ± 2.750	
10	0.166 ± 0.011	0.186 ± 0.013	0.284 ± 0.004	0.913 ± 0.037	2.508 ± 0.297	4.844 ± 0.303	7.614 ± 0.214	8.724 ± 1.168	
3	0.156 ± 0.014	0.145 ± 0.012	0.174 ± 0.012	0.307 ± 0.025	0.873 ± 0.088	1.545 ± 0.087	2.110 ± 0.131	2.819 ± 0.440	
1	0.144 ± 0.003	0.143 ± 0.003	0.134 ± 0.019	0.166 ± 0.007	0.332 ± 0.035	0.588 ± 0.042	0.769 ± 0.086	0.957 ± 0.106	
0.3	0.148 ± 0.011	0.133 ± 0.006	0.136 ± 0.007	0.143 ± 0.012	0.189 ± 0.016	0.297 ± 0.032	0.329 ± 0.016	0.390 ± 0.006	
0.1	0.145 ± 0.007	0.140 ± 0.016	0.132 ± 0.011	0.137 ± 0.003	0.162 ± 0.024	0.200 ± 0.012	0.225 ± 0.030	0.258 ± 0.029	
0	0.154 ± 0.021	0.144 ± 0.014	0.141 ± 0.019	0.146 ± 0.018	0.145 ± 0.016	0.168 ± 0.017	0.184 ± 0.026	0.196 ± 0.026	
	0	0.1	0.3	1	3	10	30	100	[TF Plasmid] (nM)

[Reporter Plasmid] (nM)	1000 μM Benzoic Acid								
100	0.205 ± 0.008	0.257 ± 0.002	0.624 ± 0.085	3.329 ± 0.575	12.805 ± 0.931	27.240 ± 3.315	32.983 ± 6.468	33.464 ± 4.077	
30	0.195 ± 0.017	0.251 ± 0.012	0.553 ± 0.047	2.407 ± 0.219	9.353 ± 1.242	21.718 ± 2.330	25.349 ± 2.320	21.771 ± 4.279	
10	0.178 ± 0.005	0.192 ± 0.019	0.390 ± 0.008	1.257 ± 0.186	3.054 ± 0.262	5.401 ± 0.233	8.547 ± 0.270	10.253 ± 1.928	
3	0.163 ± 0.024	0.152 ± 0.008	0.184 ± 0.014	0.370 ± 0.023	1.103 ± 0.072	1.683 ± 0.084	2.282 ± 0.253	3.285 ± 0.778	
1	0.139 ± 0.010	0.139 ± 0.011	0.141 ± 0.010	0.171 ± 0.010	0.386 ± 0.038	0.666 ± 0.057	0.799 ± 0.086	1.087 ± 0.322	
0.3	0.141 ± 0.008	0.137 ± 0.012	0.128 ± 0.007	0.146 ± 0.007	0.194 ± 0.021	0.298 ± 0.026	0.351 ± 0.016	0.424 ± 0.034	
0.1	0.146 ± 0.020	0.128 ± 0.011	0.141 ± 0.015	0.135 ± 0.013	0.151 ± 0.004	0.205 ± 0.013	0.238 ± 0.015	0.273 ± 0.021	
0	0.137 ± 0.017	0.134 ± 0.013	0.136 ± 0.011	0.123 ± 0.012	0.137 ± 0.017	0.164 ± 0.018	0.192 ± 0.032	0.208 ± 0.024	
	0	0.1	0.3	1	3	10	30	100	[TF Plasmid] (nM)

Fluorescence results from calibration of TF and reporter plasmids. Values represent those in Fig 9.2b and are the mean ± standard deviation for three experiments on three different days.

	Cell-Free Biosensor Concentration ($\mu g \cdot mL^{-1}$)				LC-MS Concentrations
Beverage Name	Replicate 1	Replicate 2	Replicate 3	Mean ± St. Dev.	($\mu g \cdot mL^{-1}$)
Orangina® Bottle	170.5	143.3	197.8	170.6 ± 22.3	154.23
Orangina® Can	10.3	3.4	9.6	7.7 ± 3.1	2.86
Orangina® Zero	16.6	11.8	12.3	13.6 ± 2.2	1.65
Generic Brand	18.1	13.8	10.3	14.1 ± 3.2	Not detectable
Monster® Original	304.4	172.5	217.4	231.4 ± 54.8	211.52
Monster® Absolutely Zero	147.8	139.0	193.9	160.2 ± 24.1	718.97
Monster® Ultra	172.3	150.9	154.6	159.3 ± 9.3	326.88
Monster® Ultra Red	191.1	169.0	208.4	189.5 ± 16.1	664.35
Monster® 'The Doctor'	19.0	15.6	11.0	15.2 ± 3.3	1.61
Monster® Punch	575.9	157.4	196.3	309.9 ± 188.8	315.60

Table 9.8 Benzoate concentration in commercial beverages determined from three replicates of our cell-free biosensor and LC-MS. Values represent those in Fig 9.5b . Cell-free biosensor replicates are from three experiments performed on three separate days.

Balnk	Benzoic Acid Sensor Fluorescence (AU)			
Urinary Sample	Replicate 1	Replicate 2	Replicate 3	Mean ± St. Dev.
U1	148	148	144	147 ± 2.31
U2	155	157	165	159 ± 5.29
U3	167	193	210	190 ± 21.7
U4	137	136	129	134 ± 4.36
U5	150	116	131	132 ± 17.04
U6	132	118	136	129 ± 9.45
Negative Control	152	121	134	13 ± 15.6

Table 9.9 Benzoic acid sensor shows minimal activation in response to human urine without HipO metabolic transducer. Replicates are from three experiments performed on three separate days.

[Hippuric Acid] (μM)								
1000	3.418 ± 0.937	37.338 ± 4.207	42.286 ± 2.880	44.845 ± 1.976	44.592 ± 1.666	31.485 ± 7.517	18.732 ± 3.113	2.611 ± 0.698
500	1.823 ± 1.184	34.331 ± 3.957	37.399 ± 2.495	43.171 ± 0.853	43.814 ± 1.988	34.240 ± 2.989	15.917 ± 2.386	2.314 ± 0.686
200	0.420 ± 0.010	32.865 ± 4.769	36.282 ± 1.553	41.395 ± 1.847	40.453 ± 3.345	29.809 ± 5.084	13.566 ± 3.224	1.808 ± 0.578
100	0.299 ± 0.022	29.140 ± 5.284	33.189 ± 3.416	36.347 ± 3.867	34.785 ± 5.206	25.818 ± 4.628	10.590 ± 3.103	1.256 ± 0.487
50	0.282 ± 0.022	24.886 ± 5.175	27.684 ± 4.226	28.876 ± 4.443	27.634 ± 4.623	19.913 ± 6.594	7.083 ± 2.380	0.809 ± 0.265
20	0.267 ± 0.019	12.607 ± 3.131	14.963 ± 4.850	13.064 ± 3.845	13.148 ± 3.870	8.451 ± 3.902	2.755 ± 1.121	0.345 ± 0.100
10	0.247 ± 0.026	6.187 ± 2.189	8.191 ± 3.457	6.260 ± 1.939	6.573 ± 1.744	3.319 ± 1.127	1.330 ± 0.549	0.251 ± 0.053
5	0.235 ± 0.032	2.157 ± 0.793	2.129 ± 0.697	1.600 ± 0.339	2.528 ± 0.482	1.198 ± 0.365	0.456 ± 0.150	0.206 ± 0.042
2	0.236 ± 0.031	0.534 ± 0.100	0.588 ± 0.132	0.508 ± 0.111	0.453 ± 0.090	0.363 ± 0.077	0.225 ± 0.047	0.184 ± 0.032
1	0.244 ± 0.031	0.323 ± 0.027	0.325 ± 0.032	0.322 ± 0.045	0.296 ± 0.046	0.239 ± 0.044	0.192 ± 0.039	0.177 ± 0.036
0.5	0.256 ± 0.028	0.283 ± 0.008	0.269 ± 0.022	0.262 ± 0.034	0.255 ± 0.044	0.213 ± 0.050	0.196 ± 0.039	0.185 ± 0.040
0	0.264 ± 0.021	0.268 ± 0.021	0.266 ± 0.020	0.246 ± 0.039	0.244 ± 0.042	0.210 ± 0.037	0.195 ± 0.045	0.179 ± 0.040
	0	0.1	0.3	1	3	10	30	100

[HipO Plasmid] (nM)

(a)

[Cocaine] (μM)								
1000	24.083 ± 12.948	24.127 ± 1.216	56.984 ± 6.055	80.445 ± 11.017	94.253 ± 13.664	107.991 ± 18.193	105.540 ± 15.864	89.035 ± 12.908
500	14.616 ± 10.917	15.654 ± 1.740	54.427 ± 7.990	80.311 ± 11.604	95.633 ± 16.476	107.334 ± 18.623	102.870 ± 15.567	91.222 ± 14.606
200	6.260 ± 4.509	8.904 ± 0.716	48.761 ± 5.815	76.948 ± 11.223	95.171 ± 16.311	106.021 ± 19.621	103.877 ± 17.949	85.047 ± 14.092
100	5.493 ± 5.826	6.856 ± 2.948	43.217 ± 5.932	73.683 ± 11.207	86.946 ± 15.755	99.351 ± 17.942	98.562 ± 17.365	81.615 ± 14.440
50	5.497 ± 6.513	4.600 ± 1.452	35.109 ± 6.509	69.859 ± 11.171	88.033 ± 13.621	95.244 ± 13.512	88.080 ± 11.762	64.205 ± 12.038
20	1.956 ± 1.337	1.803 ± 0.205	24.087 ± 6.205	52.674 ± 10.508	64.834 ± 8.938	68.181 ± 11.806	65.340 ± 10.151	43.962 ± 7.367
10	8.709 ± 11.304	1.036 ± 0.118	12.201 ± 2.314	28.312 ± 2.812	39.310 ± 7.824	40.145 ± 8.324	34.731 ± 3.889	23.756 ± 2.999
5	3.152 ± 3.522	1.119 ± 0.026	4.878 ± 0.537	8.934 ± 1.345	13.402 ± 1.792	12.191 ± 2.943	11.799 ± 0.967	7.943 ± 0.694
2	1.113 ± 0.760	0.733 ± 0.166	1.338 ± 0.079	1.603 ± 0.435	2.292 ± 0.348	2.230 ± 0.402	1.865 ± 0.309	1.638 ± 0.012
1	0.502 ± 0.078	0.703 ± 0.163	0.812 ± 0.119	0.891 ± 0.150	1.017 ± 0.177	1.108 ± 0.182	0.937 ± 0.217	0.806 ± 0.097
0.5	0.548 ± 0.132	0.591 ± 0.067	0.633 ± 0.125	0.648 ± 0.062	0.671 ± 0.077	0.803 ± 0.094	0.695 ± 0.143	0.614 ± 0.049
0	0.495 ± 0.083	0.498 ± 0.018	0.513 ± 0.128	0.469 ± 0.056	0.475 ± 0.019	0.503 ± 0.071	0.486 ± 0.012	0.529 ± 0.025
	0	0.1	0.3	1	3	10	30	100

[CocE Plasmid] (nM)

(b)

Fluorescence results from calibration of HipO and CocE metabolic transducer plasmids. Values represent those in Figure 9.4a and are the mean *pm* standard deviation for three experiments on three different days.

	Cell-Free Biosensor Hippuric Acid Concentration ($\mu g \cdot mL^{-1}$)				LC-MS Concentrations
Urine Sample	Replicate 1	Replicate 2	Replicate 3	Mean ± St. Dev.	($\mu g \cdot mL^{-1}$)
Urine 1	367.1	570.1	800.9	579.4 ± 177.2	368.90
Urine 2	97.6	167.8	152.2	139.2 ± 30.1	145.98
Urine 3	218.5	342.7	471.3	344.2 ± 103.2	261.91
Urine 4	218.5	331.3	394.3	314.7 ± 72.7	305.49
Urine 5	47.3	72.6	125.1	81.6 ± 32.4	100.47
Urine 6	697.3	840.1	2142.5	1226.6 ± 650.2	700.91

Table 9.10 **Endogenous hippuric acid concentration in human urine samples determined from three replicates of our cell-free biosensor and LC-MS.** Values represent those in Fig 9.5c . Cell-free biosensor replicates are from three experiments performed on three separate days.

9.7.11 SensiPath Metabolic Space Analysis

In order to probe how many biosensors could be engineered using our workflow, we downloaded the HMDB database [459] as of 25/05/2018. A set of 1445 biomarkers, with a molecular weight <500 atomic mass units (amu), was compiled for which at least one disease was identified (Supplementary Data 1).
Next, we used the RetroPath algorithm [138] embedded in the SensiPath web server [77]. RetroPath finds metabolic pathways linking analytes (source set) to effectors (sink set), i.e. small molecules activating or inhibiting transcription factors. Taking as a sink set of 727 effectors taken from a database we recently released [76], RetroPath was run using 20845 metabolic reaction rules extracted from MetaNetX [202]. We found that 192 out of 1445 biomarkers were effectors and could thus directly be detected by transcription factors. We also found that 1205 out of 1445 biomarkers could be transformed into 392 effectors through ∼80000 one-step pathways. We observed that several biomarkers could be transformed into the same effector while other biomarkers could be transformed into different effectors (see Supplementary Data 1). Finally, we found that ∼25 % of biomarkers were shared by at least two diseases. Therefore, while one can develop biosensors and re-purpose them for several diseases, biosensors can also be designed for a panel of biomarkers specific to a given disease. Altogether these results show a great potential for our workflow to engineer many biosensors detecting several pathological biomarkers.
We also probed to which extend our benzoate sensor could be used to detect various biomarkers. To that end, we computed how many HMDB metabolites could be connected to benzoate via RetroPath applying reverse reaction rules (computed from MetaNetX) to benzoate. We found that 64 HMDB metabolites could be transformed into benzoate via a one-step enzymatic transformation (Supplementary Data 2).

10 Metabolic Perceptrons for Neural Computing in Biological Systems

This work was published in Nature Communications by Amir Pandi *, Mathilde Koch *, Peter Voyvodic, Paul Soudier, Jérôme Bonnet, Manish Kushwaha and Jean-Loup Faulon.
Only minor modifications to the published paper have been introduced in the Chapter below.
* stands for equal contributions.

Detailed contribution to this thesis

In this article, complex synthetic biology circuits were developed using design tools presented in Part I and modeling and analysis tools to learn from previous designs and improve the circuits. More precisely, in a first step, the biosensors and metabolic transducers were modeled using custom Hill functions. Then, in a second step, behavior when combining those transducers was predicted *in silico* and tested *in vivo* and in cell-free systems. Our aim was first build weighted adders, which can then be modified to make a perceptron, the most basic machine learning algorithm, which is essentially a digitalized weighted sum. The control my colleagues and I require on part quantity (notably enzyme) was only possible to achieve in cell-free systems. My model was used to predict the enzyme quantities necessary to achieve the logic gates we wished to implement in our cell-free perceptrons. Therefore, my contribution to this project involved data analysis, modeling and prediction for further experimental designs.

Full reference

Pandi A.*, Koch M.*, Voyvodic P., Soudier P., Bonnet J., Kushwaha M. and Faulon J.-L. (2019) Metabolic perceptrons for neural computing in biological systems *Nature Communications*, 10.1038/s41467-019-11889-0.
* stands for equal contributions.

Contributions as stated in the article

A.P., M.Ko., M.Ku. and J.-L.F. designed the project. A.P. designed and cloned the constructs, and performed the whole-cell experiments. A.P., P.L.V., and J.B. designed cell-free experiment platform. A.P., P.L.V., and P.S. performed cell-free experiments. MKo. performed computational model simulations. All authors contributed to the manuscript write-up and approved the final manuscript.

10.1 Abstract

Synthetic biological circuits are promising tools for developing sophisticated systems for medical, industrial, and environmental applications. So far, circuit implementations commonly rely on gene expression regulation for information processing using digital logic. Here, we present a new approach for biological computation through metabolic circuits designed by computer-aided tools, implemented in both whole-cell and cell-free systems. We first combine metabolic transducers to build an analog adder, a device that sums up the concentrations of multiple input metabolites. Next, we build a weighted adder where the contributions of the different metabolites to the sum can be adjusted. Using a computational model fitted on experimental data, we finally implement two four-input 'perceptrons' for desired binary classification of metabolite combinations by applying model-predicted weights to the metabolic perceptron. The perceptron-mediated neural computing introduced here lays the groundwork for more advanced metabolic circuits for rapid and scalable multiplex sensing.

10.2 Introduction

Living organisms are information-processing systems that integrate multiple input signals, perform computations on them, and trigger relevant outputs. The multidisciplinary field of synthetic biology has combined their information-processing capabilities with modular and standardized engineering approaches to design sophisticated sense-and-respond behaviors [33, 32, 31]. Due to similarities in information flow in living systems and electronic devices [460], circuit design for these behaviors has often been inspired by electronic circuitry, with substantial efforts invested in implementing logic circuits in living cells [460, 19, 18]. Furthermore, synthetic biological circuits have been used for a range of applications including biosensors for detection of pollutants [20, 21] and medically-relevant biomarkers [22, 23], smart therapeutics [24, 25] and dynamic regulation and screening in metabolic engineering [26, 75] (and Chapter 6).

Synthetic circuits can be implemented at different layers of biological information processing, such as: (i) the genetic layer comprising transcription [29] and translation [74], (ii) the metabolic layer comprising enzymes [47, 461] and (iii) the signal

transduction layer comprising small molecules and their receptors [462, 5]. Most designs implemented so far have focused on the genetic layer, developing circuits that perform computations using elements such as feedback control [54], memory systems [463, 464], amplifiers [465, 466], toehold switches [467], or CRISPR machinery [468, 469]. However, gene expression regulation is not the only way through which cells naturally perform computation. In nature, cells carry out parts of their computation through metabolism, receiving multiple signals and distributing information fluxes to metabolic, signaling, and regulatory pathways [47, 48, 46]. Integrating metabolism into synthetic circuit design can expand the range of input signals and communication wires used in biological circuits, while bypassing some limitations of temporal coordination of gene expression cascades [41, 42].

The number of inputs processed by synthetic biological circuits has steadily increased over the years, including physical inputs like heat, light, and small molecules such as oxygen, IPTG, anhydrotetracycline (aTc), arabinose and others. However, most of these circuits process input signals using digital logic, which despite its ease of implementation lacks the power that analog logic can offer [33, 52, 53]. The power of combining digital and analog processing is exemplified by the 'perceptron', the basic block of artificial neural networks inspired by human neurons [470] that can, for instance, be trained on labeled input datasets to perform binary classification. After the training, the perceptron computes the weighted sum of input signals (analog computation) and makes the classification decision (digital computation) after processing it through an activation function.

Here we describe the development of complex metabolic circuitry implemented using analog logic in whole-cell and cell-free systems by means of enzymatic reactions. For circuit design, we first employ computational design tools, Retropath [138] and Sensipath [77], that use biochemical retrosynthesis to predict metabolic pathways and biosensors. We then build and model three whole-cell metabolic transducers and an analog adder to combine their outputs. Next, we transfer our metabolic circuits to a cell-free system [471, 308] in order to take advantage of the higher tunability and the rapid characterization it offers [72, 70, 472], expanding our system to include multiple weighted transducers and adders. Finally, using our integrated model fitted on the cell-free metabolic circuits we build a more sophisticated device called the 'metabolic perceptron', which allows desired binary classification of multi-input metabolite combinations by applying model-predicted weights on the input metabolites before analog addition, and demonstrate its utility through two examples of four-input binary classifiers. Altogether, in this work we demonstrate the potential of synthetic metabolic circuits, along with model-assisted design, to perform complex computations in biological systems.

10.3 Results

10.3.1 Whole-cell processing of hippurate, cocaine and benzaldehyde inputs

To identify the metabolic circuits to build, we use our metabolic pathway design tools, Retropath [138] and Sensipath [77]. These tools function using a set of sink compounds at the end of a metabolic pathway, here metabolites from a dataset of detectable compounds [76], and a set of source compounds that can be used as desired inputs for the circuit. The tools then propose pathways and the enzymes that can catalyze the necessary reactions, allowing for promiscuity. Our metabolic circuit layers are organized according to the main processing functions: transduction and actuation (Figure 10.1a). Transducers are the simplest metabolic circuits that function as SEMP [262], consisting of one or more enzymes that transform an input metabolite into a transduced metabolite. The transduced molecule, in turn, is detected through an actuation function that is implemented using a transcriptional regulator.

We used benzoate as our transduced metabolite, its associated transcriptional activator BenR, and the responsive promoter pBen to construct the actuator layer of our whole-cell metabolic circuits [473]. To compare the shape of the response curve, we constructed the actuator layer in two formats: (i) an open-loop circuit (Figure 10.1b) and (ii) a feedback-loop circuit (Figure 10.7). When compared to the open-loop format, the feedback-loop circuit has previously been shown to exhibit a linear dose-response to input [54, 474]. We found that while the feedback-loop format does linearize the actuator response curve, it also reduces its dynamic range (Supplementary Figure 10.7). Furthermore, the growth inhibition observed at high concentrations makes it difficult to recover the lost dynamic range by further addition of benzoate (Supplementary Figure 10.12). Therefore, we selected the open-loop format due to its higher dynamic range of activation in the tested range of benzoate concentration (Figure 10.1c), setting the maximum concentration of benzoate used in this work to the saturation point of this open-loop circuit.

We have previously implemented sensing-enabling metabolic pathways in whole-cells for detection of molecules like cocaine, hippurate, parathion and nitroglycerin [331]. Building on that work, here we implemented three upstream transducers that convert different input metabolites into benzoate for detection by the actuator layer already tested. The transducer layers were composed of enzymes HipO for hippurate (Figure 10.1d), CocE for cocaine (Figure 10.1e), and an amidase coded by vdh gene for benzaldehyde (Figure 10.1f). Compared to the benzoate output signal, we found that the transduction capacities of the three transducers were 99.6%, 49.2%, and 77.8%, respectively (10.8), indicating a partial dissipation in signal.

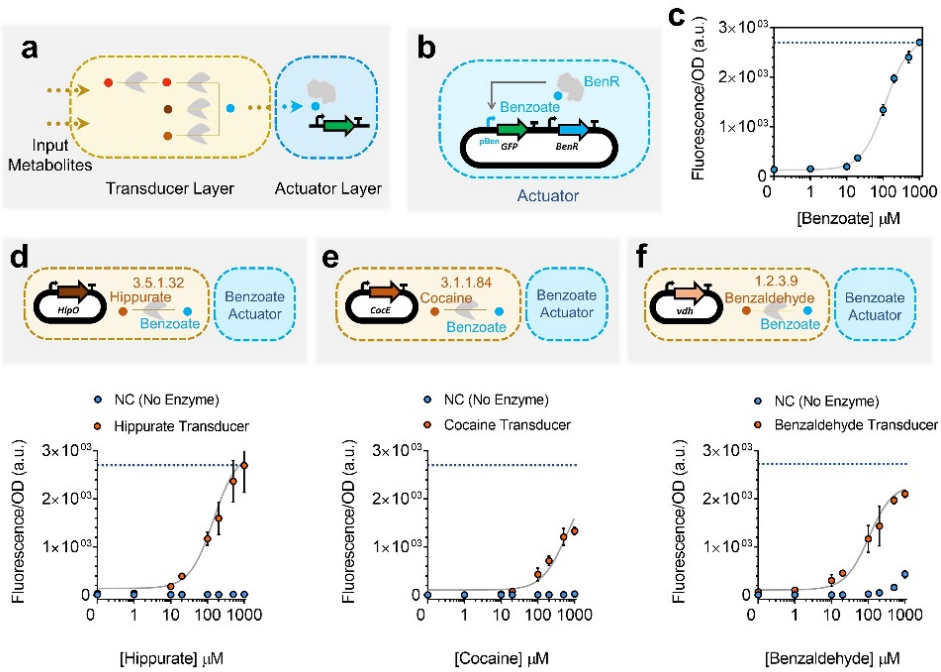


Figure 10.1 **Whole-cell actuator and metabolic transducers.** **a** Designed synthetic metabolic circuits using Retropath [138] or Sensipath [77] consist of a transducer layer and an actuator layer. **b** Open-loop circuit construction of the benzoate actuator, which is used downstream of transducer metabolic circuits in this work. For the open-loop circuit, the gene encoding the TF is expressed constitutively under control of the promoter J23101 and RBS B0032. **c** Dose-response plot of the open-loop circuit for the benzoate actuator. The gray curve is a model-fitted curve (see Methods section) for the open-loop circuit. **d**, **e**, **f** Whole-cell metabolic transducers for hippurate **d**, cocaine **e** and benzaldehyde **f** represented in dose-response plots (orange circles) and their associated dose-response when there is no enzyme present (blue circles). The blue dotted lines refer to the maximum signal from the actuator **c**. The transducer output benzoate is reported through the open-loop circuit actuator. The genes encoding the enzymes are expressed under constitutive promoter J23101 and RBS B0032. All data points and the error bars are the mean and standard deviation of normalized values from measurements taken from three different colonies on the same day.

10.3.2 A Whole-cell metabolic concentration adder

A metabolic concentration adder is an analog device composed of more than one transducer that converts their respective input metabolites into a common transduced output metabolite. For our whole-cell concentration adder, we combined two transducers to build a hippurate-benzaldehyde adder actuated by the benzoate circuit (Figure 10.2a). Unlike digital bit-adders that exhibit an ON-OFF digital behavior, our metabolic adders exhibit a continuous analog behavior that is natural for metabolic signal conversion [475] (Figure 10.2b and Supplementary Figure 10.9). Increasing the concentration of one of the inputs at any fixed concentration of the other shows an increase in the output benzoate, and thus in the resulting fluorescence (Figure 10.2b and Supplementary Figure 10.9).

The maximum output signal for our analog adder, when hippurate and benzaldehyde were both at the maximum concentration of 1000 µM, was lower than the maximum signal produced by hippurate and benzaldehyde transducers alone (Supplementary Figure 10.8). However, as seen above, the difference between the maximum signal of their transducers and the actuator was smaller. The dissipation in signal could either be because of resource competition (as a result of adding more genes) or because of enzyme efficiency (as a result of poorly balanced enzyme stoichiometries). To test these two hypotheses, we investigated the effect of the enzymes on cellular resource allocation. For this purpose, the cocaine transducer and the hippurate-benzaldehyde adder were characterized by adding benzoate to these circuits (Supplementary Figures 10.10 and 10.11). Comparing the results of these characterizations with the benzoate actuator reveals that dissipation in signal from the transducers to the actuators is due to enzyme efficiency (Supplementary Figure 10.10), whereas that from the adders to the actuators is due to resource competition (Supplementary Figure 10.11). The effect of the metabolic circuits on cell physiology are presented as the specific growth rate (μ) of the cells harboring the circuits at different concentrations of inputs (Supplementary Figures 10.12 and 10.13). Compared to the specific growth rate of cells containing empty plasmids ($\mu = 1.05 \pm 0.32$ h^{-1}), adding the metabolic circuits alone results only in a mild growth reduction. However, adding the metabolic circuits with their input metabolite(s) has a much more pronounced effect on growth reduction, particularly at high concentrations.

In order to gain a quantitative understanding of the circuits' behavior, we empirically modeled their individual components to see if we were able to successfully capture their behavior. We first modeled the actuator (gray curve in Figure 10.1c) using Hill formalism [155] as it is the component that is common to all of our outputs and therefore constrains the rest of our system. We then modeled our transducers, considering enzymes to be modules that convert their respective input metabolites into benzoate, which is then converted to the fluorescence output already modeled above. This simple empirical modeling strategy would be able to explain our transducer data, including the effects of enzyme efficiency, but not to account for observations made in Supplementary Figure 10.11, which is why we also included resource competition is our models to explain circuits with one or more transducers. To this end, we extended the Hill model to account for resource competition following previous works [153, 158], with a fixed pool of available resources

for enzyme and reporter protein production that is depleted by the transducers. This extension is further presented in the Methods section. We fitted our model on all transducers, with and without resource competition (i.e. individual transducers, or transducers where another enzyme competes for the resources). This model (presented in gray lines in Figure 10.1d,e,f and Figure 10.2c), which was not trained on adder data but only on actuator, transducer, and transducers with resource competition data, recapitulates it well. This indicates that the model accounts for all important effects underlying the data. The full training process is presented in the Methods section, and a table summarizing scores of estimated goodness of fit of our model is presented in Supplementary Table 10.3.

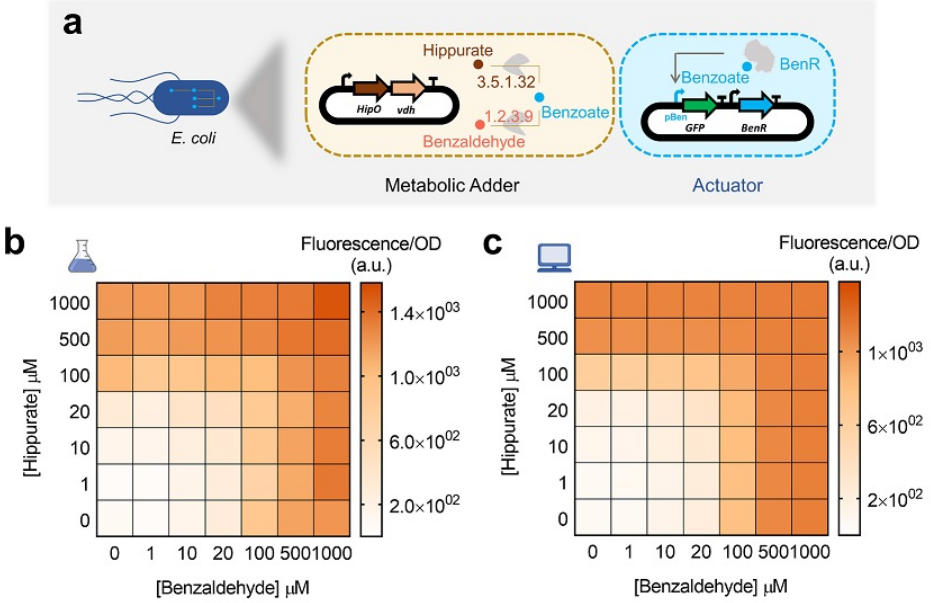


Figure 10.2 **Whole-cell metabolic adder of hippurate and benzaldehyde. a** Hippurate and benzaldehyde transducers are combined to build a metabolic adder producing a common output, benzoate, which is reported through the benzoate actuator. The genes encoding the enzymes are expressed in one operon under control of constitutive promoter J23101 and RBSs B0032 for HipO and B0034 for vdh. **b** Heat-map representing the output of the adder while increasing the concentration of both inputs, hippurate and benzaldehyde. All data points are the mean of normalized values from measurements taken from three different colonies on the same day. **c** Model simulations for experimental conditions presented in **b**. The model was fitted on transducer data and resource competition data.

10.3.3 Cell-free processing of multiple metabolic inputs

Cell-free systems have recently emerged as a promising platform [471] that provide rapid prototyping of large libraries by serving as an abiotic chassis with low susceptibility to toxicity. We took advantage of an *E. coli* cell-free system with the aim of increasing the computational potential of metabolic circuits in several ways (Figure 10.3a). Firstly, a higher number of genes can be simultaneously and combinatorially used to increase the complexity and the number of inputs for our circuits. Secondly, the lower noise provided by the absence of cell growth and maintenance of cellular pathways [476] improves the predictability and accuracy of the computation. Thirdly, having genes cloned in separate plasmids enables independent tunability of circuit behavior by varying the concentration of each part individually. Finally, cell-free systems are highly adjustable for different performance parameters and components. In all, these advantages of cell-free systems enable us to develop more complex computations than the whole-cell analog adder.

Following from our recent work [173] (and Chapter 9), we first characterized a cell-free benzoate actuator to be used downstream of other metabolic transducers. Figure 10.3a shows a schematic of the cell-free benzoate actuator composed of a plasmid encoding the BenR transcriptional activator and a second plasmid expressing sfGFP reporter gene under the control of a pBen promoter. This actuator showed a higher operational range than the whole-cell counterpart (Figure 10.1c). The optimal concentration of the TF plasmid (30 nM) and the reporter plasmid (100 nM) were taken from our recent study [173]. Following successful implementation of the actuator, we proceeded to build five upstream cell-free transducers for hippurate, cocaine, benzaldehyde, benzamide, and biphenyl-2,3-diol (Figure 10.3c,d,e,f,g) that convert these compounds to benzoate. Each of the five transducers used 10 nM of enzyme DNA per reaction, except the biphenyl-2,3-diol transducer that used two metabolic enzymes with 10 nM DNA each.

Compared to its whole-cell counterpart (Figure 10.1f), in the cell-free transducer reaction (Figure 10.3e) benzaldehyde appears to spontaneously oxidize to benzoate without the need of the transducer enzyme vdh. This behavioral difference between the whole-cell and cell-free setups could be due to the difference in redox states inside an intact cell and the cell-free reaction mix [477, 478]. Furthermore, benzamide and biphenyl-2,3-diol transducers exhibit reduction in fluorescence outputs at very high (1000 µM) input concentrations.

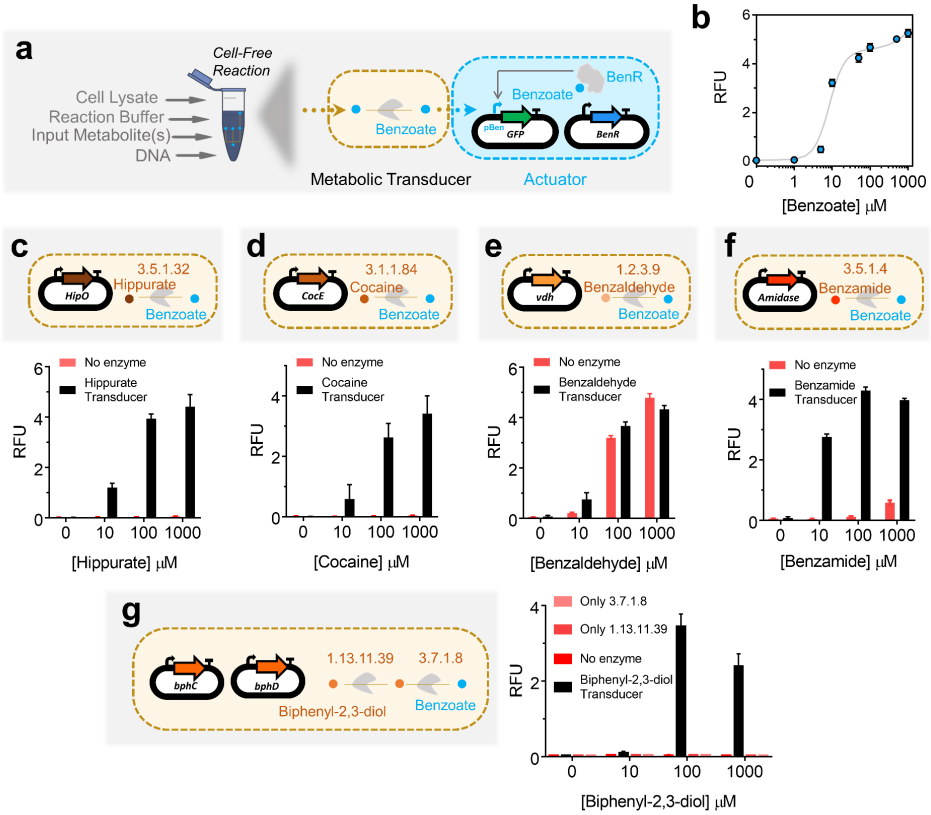


Figure 10.3 **Cell-free actuator and metabolic transducers.** **a** Implementing benzoate actuator and transducers in *E. coli* TX-TL cell-free system. Cell-free reactions are composed of cell lysate, reaction buffer (energy source, tRNAs, amino acids, etc.) and DNA plasmids. **b** Dose-response plot of the benzoate actuator in the cell-free system with 30 nM of TF-plasmid (constitutively expressed BenR) and 100 nM of reporter plasmid (pBen-sfGFP) per reaction. The data points represent the dose-response of the actuator to different concentrations of benzoate and the gray curve is a model-fitted curve on actuator data **c**, **d**, **e**, **f**, **g** Cell-free transducers coupled with the benzoate actuator for hippurate **c**, cocaine **d**, benzaldehyde **e**, benzamide **f**, and biphenyl-2,3-diol **g**, which is composed of two enzymes. All enzymes are cloned in a separate plasmid under the control of a constitutive promoter J23101 and RBS B0032. 10 nM of each plasmid was added per reaction. The bars are the response of the circuits to different concentrations of input with (transducers, black bars) and without enzyme (red bars). All data are the mean and the error bars are the standard deviation of normalized values from measurements taken from three independent cell-free reactions on the same day (RFU: Relative Fluorescence Unit).

10.3.4 Cell-free weighted transducers and adders

After characterizing different transducers in the cell-free system that enable building a multiple-input metabolic circuit, we sought to rationally tune the transducers. Cell-free systems allow independent tuning of each plasmid by pipetting different amounts of DNA. We applied this advantage to weight the flux of enzymatic reactions in cell-free transducers (Figure 10.4a). The concentration range we used was taken from our recent study [173], in order to have an optimal expression with minimum resource competition. We built four weighted transducers for hippurate (Figure 10.4b), cocaine (Figure 10.4c), benzamide (Figure 10.4d) and biphenyl-2,3-diol (Figure 10.4e). Increasing the concentration of the enzymes produces a higher amount of benzoate from the input metabolites, and hence higher GFP fluorescence. Compared to the others, the hippurate transducer reached higher GFP production at a given concentration of the enzyme and the input, and biphenyl-2,3-diol reached the weakest signal. For the biphenyl-2,3-diol transducer built with two enzymes (Figure 10.4e), both enzymes are added at the same concentration (e.g., 1 nM of 'enzyme DNA' indicates 1 nM each of plasmids encoding enzymes bphC and bphD). For a given concentration of the input there is a range within which the concentration of the enzyme DNA(s) can be varied to tune the weight of the input (Supplementary Figure 10.14).

Data in Figure 10.4 shows that similar output levels can be achieved for different input concentrations, provided the appropriate transducer concentrations are used. In the next step, we applied this finding to build hippurate-cocaine weighted adders by altering either the concentration of the enzymes or the concentration of the inputs (Figure 10.5a). The fixed-input adder is an analog adder in which the concentration of inputs, hippurate and cocaine, are fixed to 100 µM and the concentration of the enzymes is altered (top panel in Figure 10.5b). In this device, the weight of the reaction fluxes is continuously tunable. We then characterized a fixed-enzyme adder by fixing the concentration of the enzymes' DNA (1 nM for HipO, 3 nM for CocE; the cocaine signal is weaker, which is why a higher concentration of its enzyme is used) and varying the inputs, hippurate and cocaine (top panel in Figure 10.5c). However, it is important to note that the observed GFP is not a direct output from the weighted adders. Instead, the adder output is transformed by the actuator to produce the GFP signal. Since the benzoate actuator has a sigmoidal response curve (Figure 10.3b), the transformation by the actuator layer makes the visible output appear more switch-like (ON / OFF).
In order to have the ability to build any weighted adder with predictable results, we developed a model that accounts for the previous data. We first empirically modeled the actuator (gray curve in Figure 10.3b) since all other functions are constrained by how the actuator converts metabolite data (benzoate) into a detectable signal (GFP). We then fitted our model with individual weighted transducers (Supplementary Figure 10.15) and predicted the behaviors of the weighted adders (bottom panel in Figure 10.5b,c). The results shown in Figure 10.5b,c indicate that our model describes the adders well, despite being fitted only on transducer data. Supplementary Table 10.4 summarizes the different scores to estimate the goodness of fit of our model. Briefly, the model quantitatively captures the data but tends

to overestimate values at intermediate enzyme concentration ranges and does not capture the inhibitory effect observed at the high concentration of benzamide or biphenyl-2,3-diol, as this was not accounted for in the model.
Using the above strategy, we can build any weighted adder for which we have pre-calculated the weights using the model on weighted transducers. We use this ability in the following section to perform more sophisticated computation for a number of classification problems.

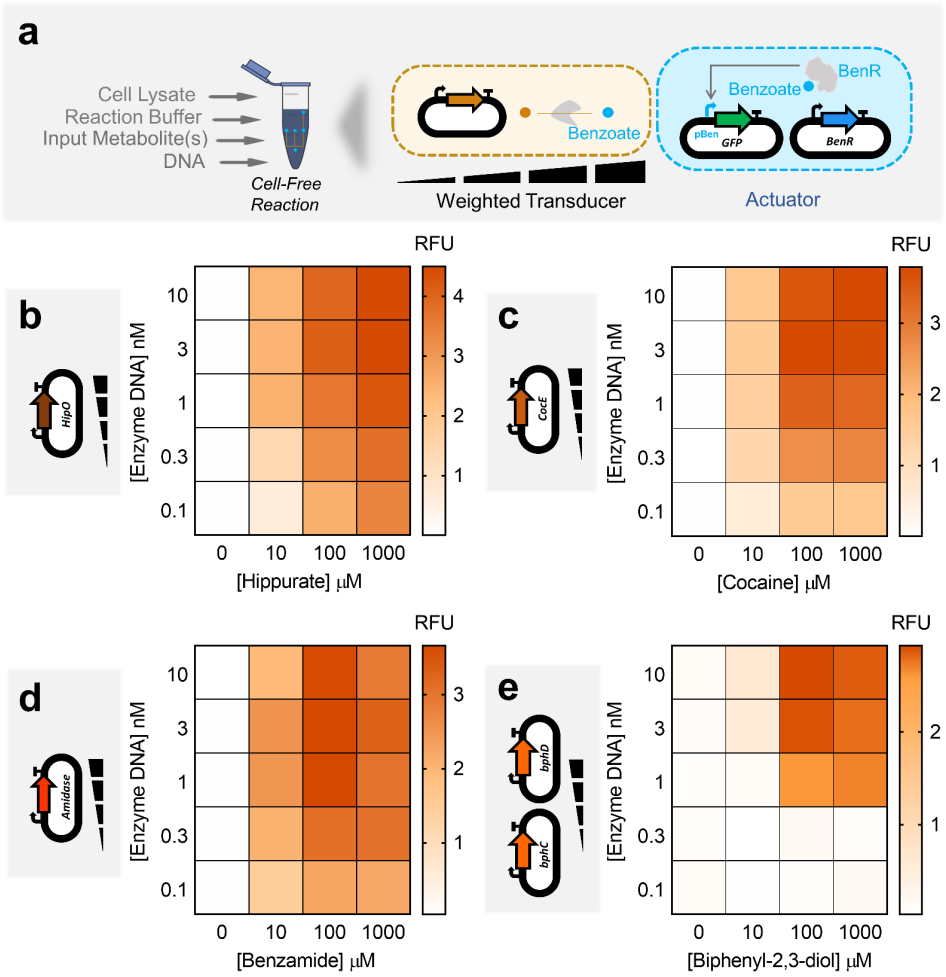


Figure 10.4 **Cell-free weighted transducers characterized by varying the concentration of the enzyme DNA. a** In the cell-free system, the circuits can be tuned by varying the amount of each enzyme pipetted per reaction. Weighted transducers are characterized by varying the concentration of the enzymes in transducers which then are reported through the benzoate actuator. The range of the concentrations was varied to get optimal expression and minimum resource competition. **b**, **c**, **d**, **e** Heat-maps representing weighted transducers at different concentrations of input molecules and enzymes DNA for hippurate **b**, cocaine **c**, benzamide **d** and biphenyl-2,3-diol **e**. For the biphenyl-2,3-diol weighted transducer**e**, concentrations represent those of each metabolic plasmid (e.g., 1 nM of 'enzyme DNA' refers to 1 nM of bphC plus 1 nM of bphD). See Supplementary Figure 10.15 for model results of each weighted transducer. All data are the mean of normalized values from three measurements. (RFU: Relative Fluorescence Unit).

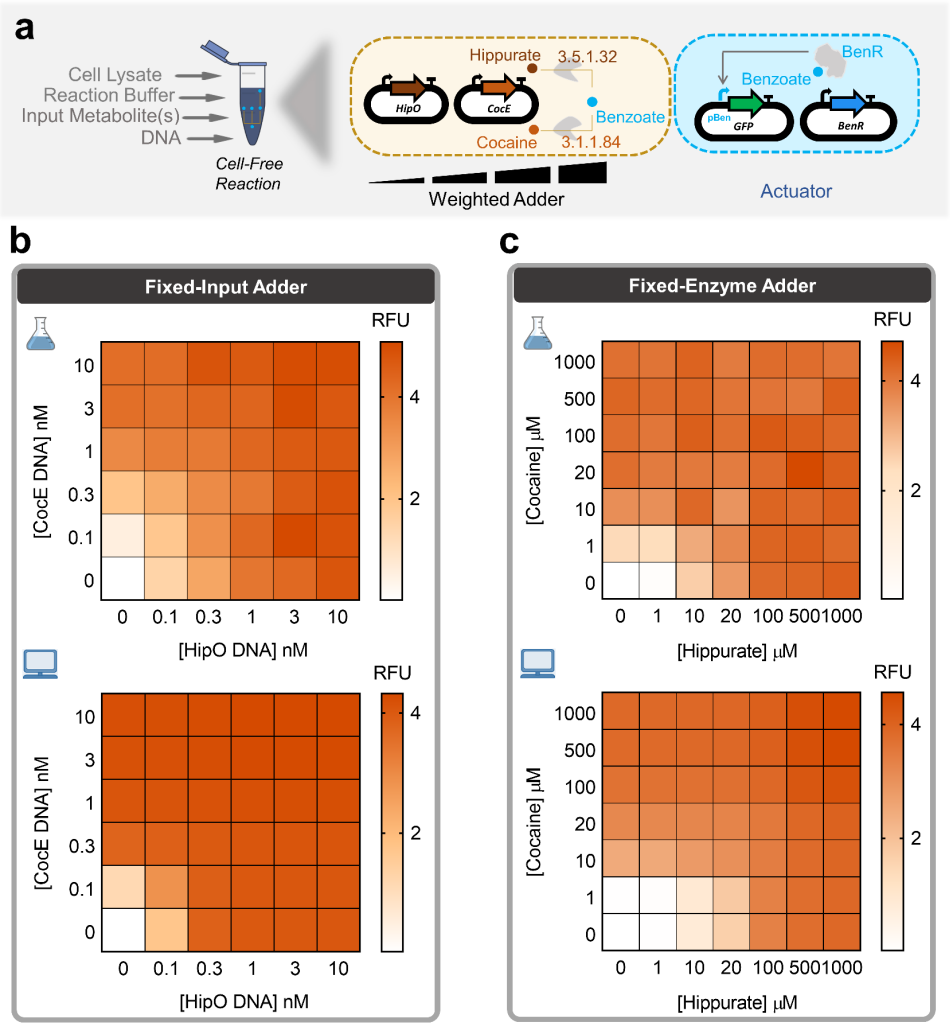


Figure 10.5 **Multiple transducers are combined to shape an adder while weighting inputs or enzymes. a** Cell-free adder characterization by varying the concentration of either inputs or enzymes producing different levels of fluorescence through the actuator. **b** Heat-map showing fixed-input adder in which the inputs, hippurate and cocaine, are fixed to 100 µM and concentrations of associated enzyme are altered by altering the concentration of plasmid DNA encoding them. *Top*: Cell-free experiment of hippurate-cocaine fixed-input (weighted) adder. *Bottom*: Model simulation (prediction) of hippurate-cocaine fixed-input (weighted) adder. **c** Fixed-enzyme adder with fixed concentrations of the enzyme DNA, 1 nM for HipO and 3 nM for CocE, and various concentrations of the inputs, hippurate and cocaine. *Top*: Cell-free experiment of hippurate-cocaine fixed-enzyme adder. *Bottom*: Model simulations (prediction) of hippurate-cocaine fixed-enzyme adder. All data are the mean of normalized values from three measurements. (RFU: Relative Fluorescence Unit).

10.3.5 Cell-free perceptron for binary classifications

The perceptron algorithm was first developed to computationally mimic the neuron's ability to process information, learn, and make decisions [479]. Perceptrons are the basic blocks of artificial neural networks enabling the learning of deep patterns in datasets by training the model's input weights [480]. Like a neuron, the perceptron receives multiple input signals (x_i) and triggers an output depending on the weighted (w_i) sum of the inputs [470]. A perceptron can be used to classify a set of input combinations after it is trained on labeled data. In binary classification, the weighted sum is first calculated ($\sum w_i.x_i$) and an activation function (f), coupled with a decision threshold d, finally makes the decision: ON if $f(\sum w_i.x_i) > d$, OFF otherwise (Figure 10.6a). The activation function can be linear or non-linear (Sigmoid, tanh, ReLU, etc.) depending on the problem [481], although a sigmoid is generally used for classification.

Since our weighted transducer models have already been fitted on the cell-free experimental data, we checked if we could use them to calculate the weights needed to classify different combinations of two inputs: hippurate and cocaine. We tested our model on five different 2-input binary classification problems (Supplementary Figure 10.16). For each problem, the two types of data were represented as a cluster of dots on the scatter plot, with the axes representing the two inputs. The fitted model was then used to identify weights needed to be applied to the weighted transducers such that a decision threshold d exists to classify the two clusters into red (ON, $> d$) or blue (OFF, $<= d$). In each binary classification, three iso-fluorescence lines threshold the data into the binary categories: ON and OFF (Supplementary Figure 10.16). These theoretical classification problems demonstrate the ability of our perceptron model to successfully carry out binary classification. It is worth noting that a binary classifier whose input(s) and output are binary values can also be represented as a logic gate. Therefore, the theoretical classification functions implemented here can also be interpreted as logic gate functions. For example, the third classifier in the figure can also be represented as the equivalent logic function (H OR C) (Supplementary Figure 10.16c).

Using the integrated model from our weighted transducers and adders, we next sought to design four-input binary classifiers using a metabolic perceptron, and test them experimentally. Our metabolic perceptron is a device enabling signal integration of multiple inputs with associated weights, represented by enzyme DNA concentrations (Figure 10.6b). The 4-input adder performs the weighted sum and the benzoate actuator acts as the activation function of the metabolic perceptron. Similar to the 2-input binary classifications above (Supplementary Figure 10.16), the weights of the four inputs can be adjusted to implement different classification functions. To illustrate the potential of building perceptrons with metabolic weighted adders, we computed adder weights using our model for two different classifiers: a simple classifier equivalent to a 'full OR' gate (Figure 10.6c), and a more complex classifier. To define the second classifier, we used our fitted model to simulate with different weights various 4-input functions that combined AND and OR behaviors. Our simulation outcomes were most reliable for hippurate and cocaine inputs since we had previously verified our model predictions on the fixed enzyme and fixed input adders (Figures 10.4 and 10.5). Consequently, we decided

to test the classification function equivalent to a '[cocaine AND hippurate] OR benzamide OR biphenyl-2,3-diol' gate (Figure 10.6d). Weight calculation methods are reported in the Methods section.

Finally, we used the cell-free system to implement the classifiers using the calculated weights and to execute the computations. While our perceptrons are trained *in silico*, they are executed in the cell-free system to predict the outcome of a given set of input signals. This is comparable to how computational perceptrons also proceed in the two phases of training and prediction. For the classifiers, the input metabolites are fixed to 100 µM, as it allows the best ON-OFF behavior for all inputs and weight-tuning according to model simulations. The model accurately predicted weights to obtain the simple 'full OR' classifier behavior (Figure 10.6d), as well as cocaine, benzamide, and biphenyl-2,3-diol weights for the second complex classifier. The initial weights computed by the model are presented in Supplementary Figure 10.17. The optimal weight of HipO (hippurate transducing enzyme) was calculated to be 0.1 nM of its DNA plasmid, which leads to higher signals than predicted, particularly for the 'ON' behavior with only hippurate. To further characterize the HipO weights at still lower concentrations of the enzyme, we performed an additional complementary characterization (Supplementary Figure 10.18). Our aim here was to find a weight for HipO through which a classifier outputs a low signal ('OFF') with only hippurate and high signal ('ON') when coupled with other inputs. We arrived at 0.03 nM DNA for HipO enzyme which exhibited this shifting behavior between 'OFF' and 'ON' (Figure 10.6d and Supplementary Figure 10.18). Using our model-guided design and rapid cell-free prototyping on the HipO weight, we were able to design two 4-input binary classifiers. In Figure 10.6c,d red circles are the weights predicted with 0.03 nM for HipO and the bars are experimental results. As noted earlier, the sigmoidal nature of the benzoate actuator's response curve (Figure 10.3b) is key to achieving the 'OFF' and 'ON' behavior exhibited by our binary classifiers. All actual values of the model and the experiments are provided in Supplementary Table S7 provided online.

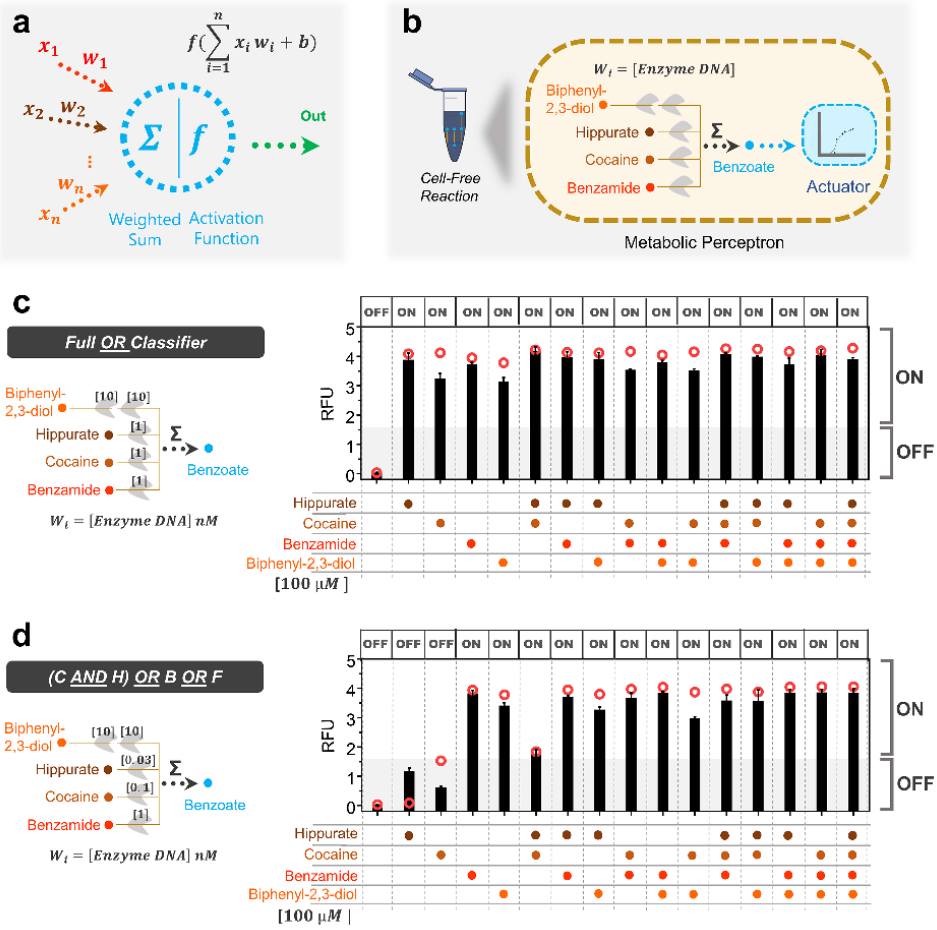


Figure 10.6 **Cell-free perceptron enabling development of classifiers.** **a** A perceptron scheme showing the inputs and their associated weights, the computation core, and the output. The perceptron computes the weights and actuates the weighted sum through an activation function. **b** Metabolic perceptron integrating multiple inputs and actuating an output. The benzoate actuator acts as the activation function of the perceptron reporting the sum of benzoate produced by the metabolic perceptron. Hippurate, cocaine, benzamide, and biphenyl-2,3-diol are the inputs of the metabolic perceptron fixed to 100 µM. The weights of the perceptron are the concentration of the enzymes calculated using the model made on weighted metabolic circuits (red circles). These weights are calculated to develop two classifiers using the metabolic perceptron and benzoate actuator.
'Full OR' classifier **c**, '[cocaine (C) AND hippurate (H)] OR benzamide (B) OR biphenyl-2,3-diol (F)' classifier **d** are the two classifiers built using this metabolic perceptron. The 'Full OR' classifier (c) classifies to 'OFF' when none of the inputs is present and it passes an arbitrary threshold to 'ON' when any of the inputs or their combinations are present. The second classifier **d** performs a more complex computation. The shading represents the arbitrary threshold that allows for perceptron decision making and the panel of 'OFF' and 'ON' at the top of the bars are the expected output of the classifiers. All data are the mean and the error bars are the standard deviation of normalized values from three measurements and red circles are the model predictions. (RFU: Relative Fluorescence Unit).

10.4 Discussion

Computing in synthetic biological circuits has largely relied on digital logic-gate circuitry for almost two decades [19, 482], treating inputs as either absent (0) or present (1). While such digital abstraction of input signals provides conceptual modularity for circuit design, it is less compatible with the physical-world input signals that vary between low and high values on a continuum [52]. As a result, digital biological circuits must carefully match input-output dynamic ranges at each layer of signal transmission to ensure successful signal processing [32, 46]. More recently, the higher efficiency of analog computation on continuous input has been recognized [483], and some analog biological circuits have started emerging [54]. In this regard, using metabolic pathways for cellular computing seems like a natural progression for analog computation in biological systems [54, 46].

In this study, we investigated the potential of metabolism to perform analog computations using synthetic metabolic circuits. To that end, we first established a benzoate actuator to report the output from our metabolic circuits in both whole-cell and cell-free systems (Figures 10.1c and 10.3b). Upstream of the actuator, we constructed hippurate, cocaine, and benzaldehyde transducers in the whole-cell system (Figures 10.1d,e,f) and a metabolic analog adder by combining the benzaldehyde and hippurate transducers (Figure 10.2). Similarly, we constructed hippurate, cocaine, benzaldehyde, benzamide, and biphenyl-2,3-diol transducers in the cell-free system (Figures 10.3c,d,e,f,g) and weighted adders by combining them (Figure 10.5). Compared to the numerous digital biological devices, which compute through multi-layered genetic logic circuits, the metabolic adder is a simple one-layered device with fast execution times.

Our computational models fitted only on the actuator and transducer data predicted adder behaviors with high accuracy (Supplementary Tables 10.4 and 10.3). This further enabled us to calculate the required weights for more complex 'metabolic perceptrons' that compute weighted sums from multiple inputs and use them to classify the multi-input combinations in a binary manner (Figures 10.6 and Supplementary Figure 10.17). Although we used fixed concentrations of inputs to demonstrate the ability of our perceptrons to classify, models fitted on characterization data from weighted transducers should enable one to build classifiers for other concentrations in the operational range of the transducers (Supplementary Figure 10.19). Indeed, as shown in Figures 10.4 and 10.5, for different input concentrations in the operational range the weight of the input can be tuned through the concentration of the enzyme DNA. To the best of our knowledge, the metabolic adders and perceptrons presented in this work are the first engineered biological circuits that use metabolism for analog computation.

Unlike genetic circuits that experience expression delays [32], metabolic circuits have the advantage of faster response times since the genes have already been expressed in the system. Yet, metabolic circuits can be connected with the other layers of cellular information processing (like genetic or signal transduction layers) when needed, to build more complex sense-and-respond behaviors. The actuator layer of our perceptrons is a good example of this, where the calculated weighted sum is converted to fluorescence output via the genetic layer. In addition, we took

advantage of the properties of cell-free systems, such as higher tunability and lack of toxicity [173, 484], to rapidly build and characterize multiple combinations of transducer-actuator circuits. Cell-free systems can be lyophilized on paper and stored at ambient temperature for < 1 year for diagnostic applications [74]. This expands the potential scope of cell-free metabolic perceptrons for use in multiplex detection of metabolic profiles in medical or environmental samples [74, 173].
Here, we have built a single-layer perceptron, with positive weights, that can classify different profiles of input metabolites by applying different weights to each transducer. In the future, by adding competing or attenuating reactions that reduce the concentration of the transduced metabolite in response to an input, it may be possible to expand the training space by applying negative weights to certain inputs [485]. Furthermore, a single-layer perceptron can only classify data that is linearly separable [486], which means that it should be possible to draw a line between the two classes of data points in order for the perceptron to classify them (Supplementary Figure 10.16). In contrast, multi-layer perceptrons can approximate any function [487] and can be used for more complex pattern recognition tasks [488]. With the use of bio-retrosynthesis-based computational tools for metabolic pathway design, like Retropath [138] and Sensipath [77], it is theoretically possible to build multi-layer metabolic perceptrons that can classify complex patterns of metabolic states *in vivo*, or identify different metabolite concentrations in analytical samples (Supplementary Figure 10.20). Finally, it may also be possible to apply *in situ* learning (within the whole-cell or cell-free environment) by applying winner selection strategies on successful classifiers [489].
However, the use of the metabolic layer for biological computing is currently underexplored. To expand the computing potential of metabolic circuits, many more metabolic parts and devices (transducers, adders, and actuators) will need to be exhaustively characterized and databases built with descriptions of activities, dynamic ranges, cross-talk, chassis dependence, cell-free composition dependence, and other functional parameters. Here, we provide a detailed method for the identification of novel parts and the step-wise building of new devices, and make our scripts available. These can form the stepping-stone for building a larger framework for fully automated design of metabolic circuits, similar to the Cello tool for automated genetic circuit design [29].

10.5 Methods

10.5.1 Designing synthetic metabolic circuits

Retropath [138] and Sensipath [77] were used to design the metabolic circuits between potential input metabolites and detectable metabolites as outputs [76]. These tools function using a set of sink compounds, a set of source compounds, and a set of chemical rules [116] implementing enzyme-mediated chemical transformations. They then use retrosynthesis to propose pathways and the enzymes that can catalyze the necessary reactions, allowing promiscuity, between compounds from the

sink and compounds from the source. To design the adder, the Retropath software was used with a set of detectable compounds as the sink and the molecules we wish to use as circuit inputs as the source. The results were potential pathways and the associated enzymes, which were then analyzed for feasibility. The sequences of the enzymes were codon-optimized, synthesized and implemented in *E. coli* or taken from a previous study.

10.5.2 Molecular biology

All plasmids were made using Golden Gate assembly in *E. coli* Mach1 chemically competent cells (strain W, genotype: $F^{-}\phi 80(lacZ)\Delta M15\ \Delta lacX74 hsdR(r_{K^-}m_{K^+})$ $\Delta recA1398endA1tonA$). Whole-cell constructs were cloned in BioBrick standard vectors pSB1K3 (kanamycin resistance, pMB1 replication origin, high-copy plasmid, ∼32 plasmids per genome [490]) and pSB4C5 (chloramphenicol resistance, pSC101 replication origin, low-copy plasmid, ∼3.4 plasmids per genome [490]) and the genes encoding TF and all the enzymes were expressed under constitutive promoter J23101 and RBS B0032. All cell-free plasmids were cloned in pBEAST52 (a derived vector from pBEST [320], ampicillin resistance, pMB1 replication origin, high-copy plasmid, ∼32 plasmids per genome [490]). BenR cell-free plasmid and its cognate responsive prompter, pBen, expressing super-folder GFP were taken from our recent work [173]. All other cell-free enzymes were cloned under constitutive promoter J23101 and RBS B0032. Sequence and source of all the genes and parts are available in Supplementary Table S5 online and the plasmids used in this study (Addgene deposit) are listed in Supplementary Table S6 online. Synthetic sequences were provided by Twist Bioscience. Enzymes for cloning including Q5 DNA polymerase, BsaI, and T4 DNA ligase were purchased from New England Biolabs. DNA plasmids for cell-free reactions were prepared using the Macherey-Nagel maxiprep kit.

10.5.3 Characterization of whole-cell circuits

For each circuit separate colonies of *E. coli* TOP10 (strain K-12) strains harboring the circuit plasmids were cultured overnight at 37 °C in LB with appropriate antibiotic. The next day each culture was diluted 100x in LB with antibiotics. 95 µL of fresh cultures were distributed in 96-well plate (Corning 3603) and the plate was incubated to reach the $OD_{600} \sim 0.1$ in a plate reader (Biotek Synergy HTX). Then 5 µL of the input metabolites (100x ethanol solutions 5x diluted in LB) were added and the plate was incubated for 18 hours at 37 °C. During the incubation, the OD_{600} and GFP fluorescence (gain: 35, ex: 458 nm, em: 528 nm) were measured. Benzoate, hippurate, cocaine hydrochloride, benzaldehyde, benzamide and biphenyl-2,3-diol (2,3-dihydroxy-biphenyl) were purchased from Sigma-Aldrich. Permission to purchase cocaine hydrochloride was given by the French drug regulatory agency (Agence Nationale de Sécurité du Médicament et des Produits de Santé). For all chemicals, serial dilutions of 100x concentrations were prepared in ethanol. The formula presenting the results of the circuits' characterization is shown

in data normalization section. The mean and standard deviation of all normalized data are provided in Supplementary Table S7 available online.

10.5.4 Cell-free extract and buffer preparation

Cell-free *E. coli* extract was produced as previously described [173, 68, 449]. Briefly, an overnight culture of BL21 Star (DE3)::RF1-CBD3 *E. coli*was used to inoculate 4L of 2xYT-P media in six 2 L flasks at a dilution of 1:100. The cultures were grown at 37 °C with 220 rpm shaking for approximately 3.5-4 hours until the OD_{600} = 2-3. Cultures were centrifuged at 5000 x g at 4 °C for 12 minutes. Cell pellets were washed twice with 200 mL S30A buffer (14 mM Mg-glutamate, 60 mM K-glutamate, 50 mM Tris, pH 7.7), centrifuging after each wash at 5000 x g at 4 °C for 12 minutes. Cell pellets were then re-suspended in 40 mL S30A buffer and transferred to pre-weighed 50 mL Falcon conical tubes where they were centrifuged twice at 2000 x g at 4 °C for 8 and 2 minutes, respectively, removing the supernatant after each. Finally, the tubes were reweighed and flash frozen in liquid nitrogen before storing at −80 °C.
Cell pellets were thawed on ice and re-suspended in 1 mL S30A buffer per gram of cell pellet. Cell suspensions were lysed via a single pass through a French press homogenizer (Avestin; Emulsiflex-C3) at 15000-20000 psi and then centrifuged at 12000 x g at 4 °C for 30 minutes to separate out cellular cytoplasm. After centrifugation, the supernatant was collected and incubated at 37 °C with 220 rpm shaking for 60 minutes. The extract was re-centrifuged at 12000 x g at 4 °C for 30 minutes, and the supernatant was transferred to 12-14 kDa MWCO dialysis tubing (Spectrum Labs; Spectra/Por4) and dialyzed against 2 L of S30B buffer (14 mM Mg-glutamate, 60 mM K-glutamate, ∼ 5 mM Tris, pH 8.2) overnight at 4 °C. The following day, the extract was re-centrifuged one final time at 12000 x g at 4 °C for 30 minutes, aliquoted, and flash frozen in liquid nitrogen before storage at −80 °C. The buffer for cell-free reactions is composed such that final reaction concentrations were as follows: 1.5 mM each amino acid except leucine, 1.25 mM leucine, 50 mM HEPES, 1.5 mM ATP and GTP, 0.9 mM CTP and UTP, 0.2 $mg \cdot mL^{-1}$ tRNA, 0.26 mM CoA, 0.33 mM NAD, 0.75 mM cAMP, 0.068 mM folinic acid, 1 mM spermidine, 30 mM 3-PGA, and 2% PEG-8000. Additionally, the Mg-glutamate (0-6 mM), K-glutamate (20-140 mM), and DTT (0-3 mM) levels were serially calibrated for each batch of cell-extract for maximum signal. One batch of buffer was made for each batch of extract, aliquoted, and flash frozen in liquid nitrogen before storage at −80 °C.

10.5.5 Characterization of cell-free circuits

Cell-free reactions were performed in 15.75 µL of the mixture of 33.3% cell extract, 41.7% buffer, and 25% plasmid DNA, input metabolites, and water. The reactions were prepared in PCR tubes on ice and 15 µL of each was pipetted into 384-well plates (Thermo Scientific 242764). GFP fluorescence out of each circuit was recorded in the plate reader at 30 °C (gain: 50, ex: 458 nm, em: 528 nm). The

background (cell-free reaction without any plasmid) corrected fluorescence data were normalized by $20\,\mathrm{ng}\cdot\mu\mathrm{L}^{-1}$ of a plasmid expressing strong constitutive sfGFP (under OR2-OR1-Pr promoter [173]) and were plotted after 8 hours incubation. The mean and standard deviation of all normalized data are provided in Supplementary Table S7 available online.

10.5.6 Data normalization

For whole-cell data, we use the following normalization:

$$Fluorescence(input) = \frac{GFP(input) - GFP(LB)}{OD(input) - OD(LB)} - \frac{GFP(empty - plasmid) - GFP(LB)}{OD(empty - plasmid) - OD(LB)}$$

Reference: cells harboring empty plasmids

For cell-free data, we consider Relative Fluorescence Unit (RFU):

$$RFU(input) = \frac{GFP(input) - GFP(extract)}{GFP(reference) - GFP(extract)}$$

Reference: $20\,\mathrm{ng}\cdot\mu\mathrm{L}^{-1}$ of a plasmid expressing the constitutive sfGFP under OR2-OR1-Pr promoter [173].

10.5.7 Simulation tools and parameter fitting

All data analysis and simulations were run on R (version 3.2.3). Dose-response curves were fitted using ordinary least squares errors and the R optim function (from Package stats version 3.2.3, using the L-BFGS-B method implementing the Limited-memory Broyden Fletcher Goldfarb Shanno algorithm, which is a quasi-Newton method). For the random parameter sampling around the mean fit, values were sampled from within$\pm$1.96 standard error of the mean of the parameter estimation. The seed was set so as to ensure reproducibility. All simulations were run in the Rstudio development environment. All parameters are presented in Supplementary Tables 10.6 and 10.5.

10.5.8 Whole-cell model

The whole-cell model is composed of three parts: the actuator, the transducers (which all obey the same law) and the resource competition.

$$Actuator(total) = (\frac{total^{hill_a}}{K_M^{hill_a} + total^{hill_a}} * f_c + 1) * basal$$

where *total* is the concentration of the considered input (in µM), K_M is the concentration that allows for half-maximum induction (in µM), also termed IC_{50}, $hill_a$ is the Hill coefficient that characterizes the cooperativity of the induction system, f_c is the dynamic range (in AU) and *basal* is the basal GFP fluorescence without input (benzoate).

$$Transducer(input) = input * range_{enz}$$

Where *input* is the input concentration in µM and $range_{enz}$ is a dimensionless number characterizing the capacity of the enzyme to transduce the signal. When combining transducers with the actuator, transducer results are added before being fed into the actuator equation, just as benzoate concentrations are added before being converted to a fluorescent signal in the cell.

To account for resource competition, given our experimental results where there is little competition with one enzyme and significant competition with two, we used an equation including cooperativity of resource competition. This reduces the fold change of the actuator as there are less resources available for producing transcription factors and GFP.

$$Result(out) = range_{res} * out * (\frac{E^{nr}}{E^{nr} + (coce + benz + ratio * hipo)^{ns}})$$

where out is the result of the actuator transfer function before accounting for resource competition, $range_{res}$, E, nr characterize the Hill function that accounts for competition, *coce*, *benz* and *hipo* are the enzyme plasmid concentrations. *ratio* accounts for the differences in burden from different enzymes, its value around 0.8 is close to the ratio between enzyme lengths (1500 for benzaldehyde transducing enzyme and 1200 for HipO).

10.5.9 Cell-free model

The model is composed of two parts: the actuator and the transducers.

$$Actuator(total) = (\frac{total^{hill_a}}{K_M^{hill_a} + total^{hill_a}} * f_c + 1) * basal + lin * 0.0001 * total$$

where *total* is the concentration of the considered input metabolite (in µM), K_M is the concentration that allows for half-maximum induction (in µM), also termed IC_{50}, $hill_a$ is the Hill coefficient that characterizes the cooperativity of the induction system, f_c is the dynamic range (in AU) and *basal* is the basal GFP fluorescence without input (benzoate). *lin* accounts for the linearity observed in the actuator behavior at concentrations saturating the Hill transfer function.

$$Transducer(input) = range_{enzyme} * \frac{E^{n_E}}{K_E^{n_E} + E^{n_E}} * \frac{input^{n_{input}}}{K_I^{n_{input}} + input^{n_{input}}}$$

Where $range_{enzyme}$ is a dimensionless number characterizing the capacity of the enzyme to transduce the signal. The activity of the enzyme is characterized by a Hill function as increasing concentrations do not lead to a linear increase but enzymes saturate (E is the enzyme quantity in nM, K_E and n_E are its Hill constants), and similarly, input is the input metabolite concentration in µM with K_I and n_{input} as its Hill constants.

When combining transducers, transducer results are added before being fed into the actuator equation, just as benzoate concentrations are added before being converted to the fluorescent signal in the cell.

10.5.10 Model parameters fitting process

Our fitting process is detailed in the Readme files supporting our modeling scripts provided in GitHub and is summarized here. It is done in the two steps presented here: first fitting of the actuator then fitting of the transducers.
As the first step, the actuator transfer function model (benzoate transformed into fluorescence) is fitted 100 times on the actuator data (Figures 10.1c and 10.3b), with all actuator parameters allowed to vary. The mean, standard deviation, standard error of the mean and confidence interval were saved at 95% of the estimation of those parameters. For transducer fitting (all transducers in cell-free and all except cocaine in whole-cell, data from Figures 10.1d and 10.1f, resource competition from Figures 10.2b and 10.2c , 10.4b, 10.4c, 10.4d, 10.4e), we constrained the actuator characteristics in the following way: upper and lower allowed values are within the 95% confidence interval (or plus or minus one standard deviation from the mean for fold change and baseline in cell-free as it allowed a wider range, accounting for the decrease in actuator signal in transducer experiments without affecting the shape of the sigmoid). The initial values for the fitting process were sampled from a Gaussian distribution centered on the mean parameter estimation and spread with a standard deviation equal to the standard error of this parameter estimation. We then allowed fitting of all transducer parameters freely and of the actuator parameters within their 95% confidence interval.
Once this is done, all common parameters (actuator transfer function and resource competition) were sampled using the same procedure and fitting on the cocaine

transducer was performed. To show that parameters are well constrained (proving they minimally explain the data from Figure 10.1e), Supplementary Figures 10.21 and 10.22 show results of sampling parameters from the final parameters distribution (without fitting at that stage) and how they compare to the data.

10.5.11 Objective functions and model scoring

In order to evaluate and compare our models, we used the following functions.

$$RMSD = \sqrt{(\frac{\sum_1^n (y_i^{true} - y_i^{pred})^2}{n}}$$

It measures how close the model is to the experiments. It allows for comparison of different models on the same data, the one with the smaller $RMSD$ being better, but does not allow comparison between experiments.

$$R^2 = 1 - \frac{\sum_1^n ((y_i^{true} - y_i^{pred})^2)}{\sum 1n((y_i^{true} - y_{mean}^{true})^2)}$$

R^2 allows measuring the goodness of fit. When the prediction is only around the sample mean, $R^2 = 0$. When the predictions are close to the real experimental value, R^2 gets closer to 1, whereas it can have important negative values when the model is really far off.

$$WeightedR^2 = 1 - \frac{\sum_1^n (\frac{(y_i^{true} - y_i^{pred})^2}{std_i^2})}{\sum 1n(\frac{(y_i^{true} - y_{mean}^{true})^2}{std_i^2})}$$

It is a variant of R^2 that weights samples according to their experimental error, giving more weight or more certain samples. It otherwise has the same properties as R^2.

$$Error - percentage = abs(\frac{y_i^{true} - y_i^{pred}}{y_i^{true}}) * 100$$

This measures the percentage of error for each point. We present the average on all experiments in Supplementary Tables 10.4 and 10.3.

10.5.12 Perceptron weights calculation

In order to calculate the weights for the classifiers presented in Figure 10.6, we followed the following procedure. First, we defined the expected results (expressed in 'OFF's and 'ON's). We also defined a list of weights to test for each enzyme (here, between 0.1 nM and 10 nM, as tested in our weighted transducers). Then, for each combination of enzyme weights, we simulated the outcome of the classifiers for all possible input combinations using our previously fitted model. We then tested various possible thresholds and kept the enzyme combinations for which a threshold exists that allows for the expected behavior. As the last step, we manually analyzed the classifier to keep the ones both a high difference between ON and OFF, and a minimal enzyme weight to prevent resource competitions issues that could arise as we are adding more genes than previous experiments. In order to perform clustering presented in Supplementary Figure 10.17, we sampled values uniformly within the stated ranges ([0, 2 µM] for low values and [80, 100 µM] for high values). We then simulated the results to assess the robustness of our designs. The best set of weights from this procedure to achieve the desired classification function (the 'trained' weights) are then used for the cell-free implementation.
The difference between our metabolic perceptron and an *in silico* perceptron is that the latter exhibits a perfect activation behavior: digital (0 / 1), sigmoidal, ReLU, or another activation function; its weights can be tuned exactly as desired. In our implementation of the cell-free metabolic circuits, many biological details complicate the relationship between the inputs and the activator output. We therefore used more detailed step-wise empirical modeling to account for the biology in our system rather than an off-the-shelf perceptron code that would be unable to capture all the subtleties in our data.

10.5.13 Binary clustering experiments

In order to perform the binary/2D clustering experiments, we sampled values uniformly within the stated ranges ([0, 2 µM] for low values and [80, 100 µM] for high values). For different weight (HipO and CocE) values, we simulated the fluorescence output of each of those cocaine-hippurate combinations. Moreover, for different threshold values (3, 3.5 and 4, as presented in Supplementary Figure 10.16), we numerically solved for the benzoate concentration such that $transfer(benzoate) = fluorescence - threshold$ and then for values of cocaine and hippurate such that $transducer(cocaine) + transducer(hippurate) = benzoate$ This equation with two unknowns gives us a curve of cocaine and hippurate values that would lie on our decided threshold for this set of weights. All combinations on the top right of that curve will be classified to 'ON' and all combinations below will be classified as 'OFF'.

10.5.14 Data availability

Source data for main and supplementary figures are provided in the supplementary materials. Other raw data are available from the corresponding authors upon reasonable request.

10.5.15 Code availability

All scripts and data for generating results presented in this paper are available on GitHub.

10.5.16 Biological and chemical identifiers

In order to allow easier parsing of our article by bio-informatics tools, we provide here the identifiers of our biological sequences and chemical compounds.

Compound name	**InChI**
Benzoate (Benzoic acid)	InChI=1S/C7H6O2/c8-7(9)6-4-2-1-3-5-6/h1-5H,(H,8,9)
Hippurate (Hippuric acid)	InChI=1S/C9H9NO3/c11-8(12)6-10-9(13)7-4-2-1-3-5-7/h1-5H,6H2,(H,10,13)(H,11,12)
Cocaine	InChI=1S/C17H21NO4/c1-18-12-8-9-13(18)15(17(20)21-2)14(10-12)22-16(19)11-6-4-3-5-7-11/h3-7,12-15H,8-10H2,1-2H3/t12-,13+,14-,15+/m0/s1
Benzaldehyde	InChI=1S/C7H6O/c8-6-7-4-2-1-3-5-7/h1-6H
Biphenyl-2,3-diol	InChI=1S/C12H10O2/c13-11-8-4-7-10(12(11)14)9-5-2-1-3-6-9/h1-8,13-14H
Benzamide	InChI=1S/C7H7NO/c8-7(9)6-4-2-1-3-5-6/h1-5H,(H2,8,9)

Table 10.1 Chemical structures for compounds used in this work.

Gene	Description	Specy	Identifier (UniProtKB)
BenR	Benzoate sensitive transcription factor	*Pseudomonas putida*	Q9L7Y6
HipO	Hippurate hydrolase (EC: 3.5.1.32)	*Campylobacter jejuni*	P45493
CocE	Cocaine esterase (EC: 3.1.1.84	*Rhodococcus sp.*	Q9L9D7
vdh	Aryl-aldehyde oxidase (EC: 1.2.3.9)	*Acinetobacter johnsonii SH046*	D0RZT4
bphC	Biphenyl-2,3-diol 1,2-dioxygenase (EC: 1.13.11.39)	*Pseudomonas sp.*	P17297
bphD	2-Hydroxy-6-oxo-6-phenylhexa-2,4-dienoate hydrolase (EC: 3.7.1.8)	*Pseudomonas putida*	Q52036
Benzamide transforming enzyme	Amidase (EC: 3.5.1.4)	*Rhodococcus erythropolis*	B4XEY3

Table 10.2 **Sequences identifiers for parts used in this work.**

Sequence and source of all the genes and parts are available in Supplementary Table S5 available online and the plasmids used in this study (Addgene deposit) are listed in Supplementary Table S6 available online and at Addgene 1 and Addgene 2.

10.6 Supplementary data

10.6.1 Circuit design

10.6.2 Detailed data from Figures 10.1 and 10.2

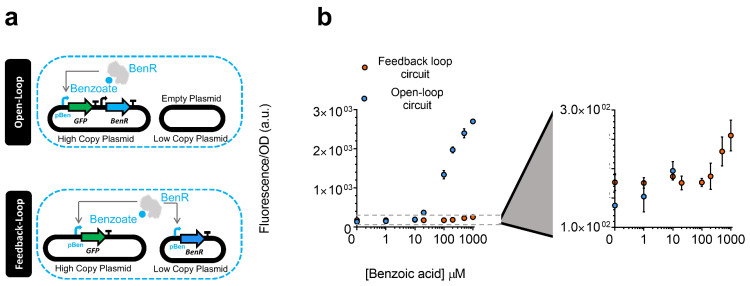


Figure 10.7 **Feedback-loop circuit design of the benzoate actuator.** **a** The open-loop circuit (Figure 10.1b) versus a feedback-loop circuit for the benzoate actuator. In the feedback-loop actuator the gene encoding TF is expressed under its responsive promoter, pBen, in a low copy plasmid and sfGFP reporting the signal in a high copy plasmid [54]. **b** The dose-response of the feedback-loop versus the open-loop circuit (Figure 10.1c) to different concentrations of benzoate. All data points and the error bars are the mean and standard deviation of normalized values from measurements taken from three different colonies on the same day.

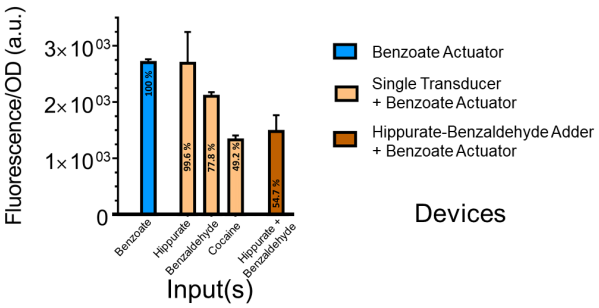


Figure 10.8 **Comparison of the maximum signals of whole-cell circuits.** Comparison of the maximal signal of hippurate, benzaldehyde, and cocaine transducers (beige) as well as hippurate-benzaldehyde adder (orange) with benzoate actuator (blue). The maximum signal of all the circuits are at the maximum concentration of their inputs (1000 µM). The percentage in each bar represents its value with regard to the maximum signal of benzoate in benzoate actuator. The actuator (blue) and transducer (beige) data and error bars are from the results presented in Figure 10.1. The adder (orange) data and error bars are from the results presented in Figure 10.2.

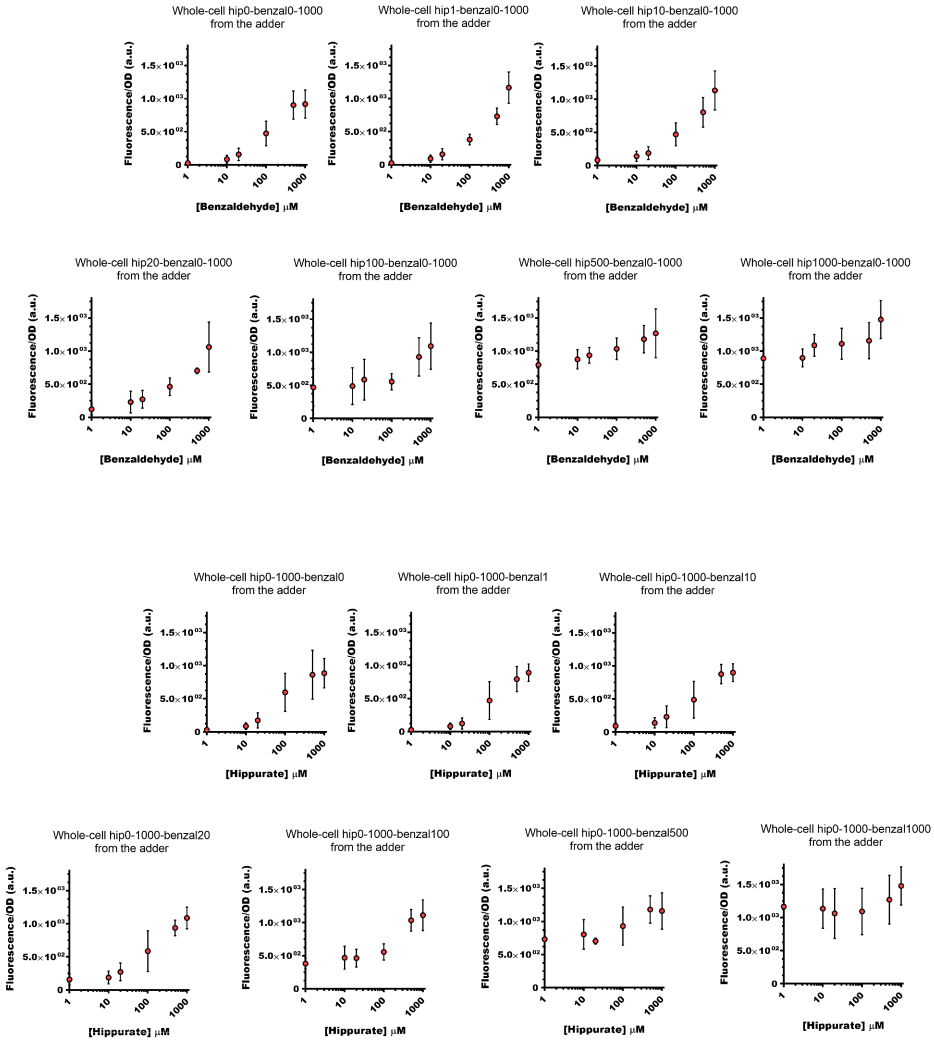


Figure 10.9 **2D plots for the data presented in heat-map in Figure 10.2b**. These 14 plots help visualize the linearity of metabolic addition. At the top of each plot the columns/rows corresponding to the heat-map in Figure 10.2b have been labeled.

10.6.3 Analyzing resource competition

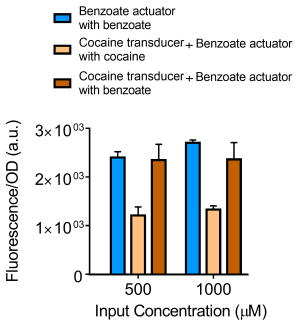


Figure 10.10 **Examining the effect of resource competition versus enzyme efficiency on the whole-cell cocaine transducer.** To study these effects on the single-enzyme metabolic circuit, the following experiment was performed: cocaine transducer (with the highest signal dissipation among the three tested in Figure 10.1) was supplied with benzoate input, to test the effect of enzymes on only cellular resource allocation but not the conversion of inputs to benzoate. The cocaine transducer (+ benzoate actuator) with benzoate input shows a behavior similar or close to the benzoate actuator alone. All data points and the error bars are the mean and standard deviation of normalized values from measurements taken from three different colonies on the same day.

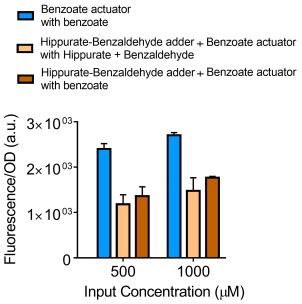


Figure 10.11 **Examining the effect of resource competition versus enzyme efficiency on the whole-cell cocaine transducer.** To study these effects on the two-enzyme metabolic circuit (adder) the following experiment was performed: hippurate-benzaldehyde adder was supplied with benzoate input, to test the effect of enzymes on only cellular resource allocation but not the conversion of inputs to benzoate. The adder (+ benzoate actuator) with benzoate input shows a behavior similar to the adder (+ benzoate actuator) with hippurate and benzaldehyde inputs. All data points and the error bars are the mean and standard deviation of normalized values from measurements taken from three different colonies on the same day.

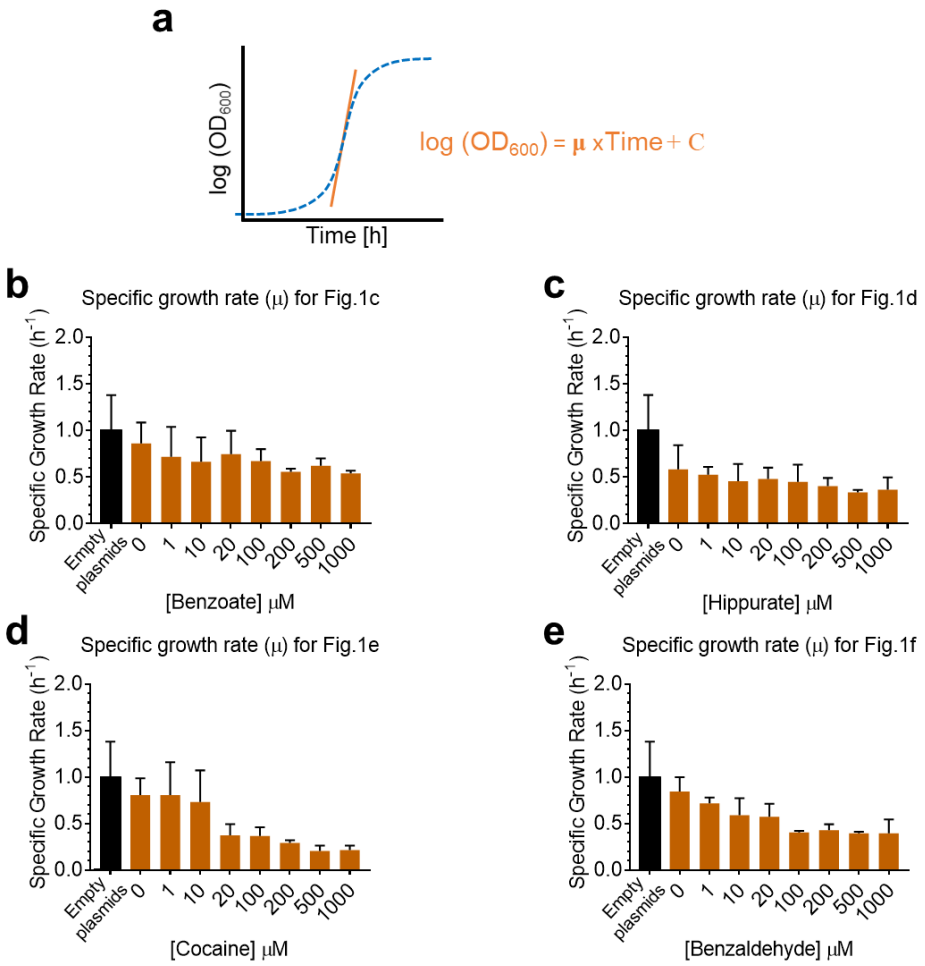


Figure 10.12 **The specific growth rate (μ) values of the whole-cell circuits presented in Figure 10.1.** **a** The schematic of the calculation of the specific growth rate (μ) values from OD_{600} kinetic values over time. It is calculated as the slope of the line drawn in the range of exponential phase of the growth when log (OD_{600}) is plotted over time. The specific growth rate (μ) values of the cells harboring circuits for benzoate actuator **b**, hippurate **c**, cocaine **d** and benzaldehyde **e** transducers presented in Figure 10.1. The OD data were collected from cells exposed to the input metabolite for 2-4 hours and growing at 37 °C in a 96-well plate using a plate reader (Biotek Synergy HTX). All data points and the error bars are the mean and standard deviation of normalized values from measurements taken from three different colonies on the same day.

10.6.4 Analysing growth rates

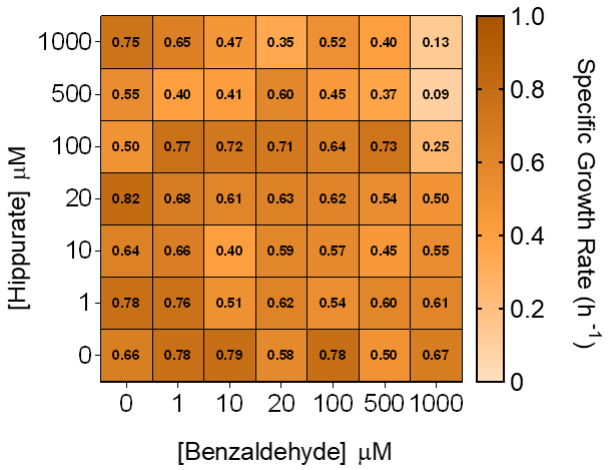


Figure 10.13 **The specific growth rate (μ) values of the whole-cell adder presented in Figure 10.2b.** The OD data were collected from cells exposed to the input metabolites for 2-4 hours and growing at 37 °C in a 96-well plate using a plate reader (Biotek Synergy HTX). The schematic of the calculation of the specific growth rate (μ) values from OD_{600} kinetic values over time is presented in Figure 10.12a. It is calculated as the slope of the line drawn in the range of the exponential phase of growth when log (OD_{600}) is plotted over time. All data points are the mean of normalized values from measurements taken from three different colonies on the same day.

10.6.5 Weighted cell-free transducers

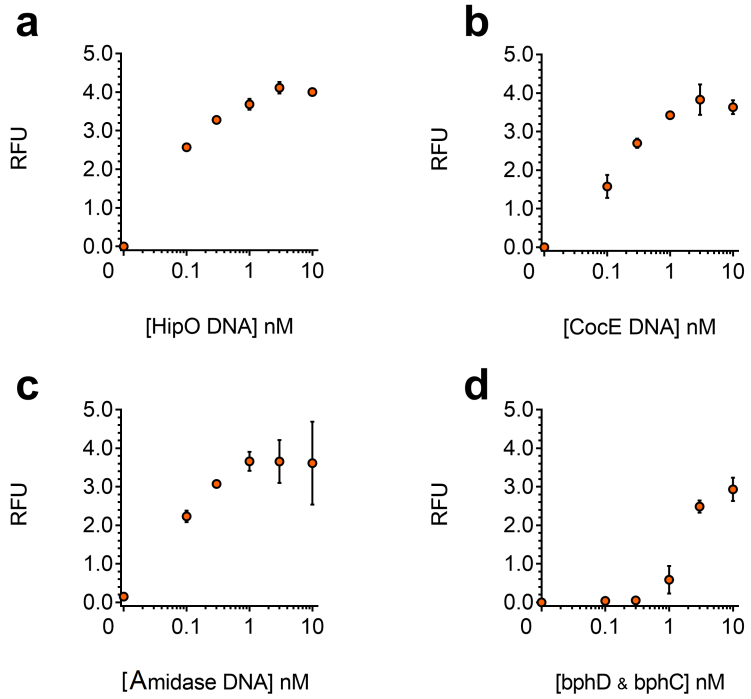


Figure 10.14 **The dose-response of cell-free transducers to different concentrations of the associated enzyme DNA (weights) for weighted transducers.** The behavior of the cell-free transducers at constant concentration of inputs (100 µM) while the weights (concentration of the enzyme DNA) are varied for hippurate (a), cocaine (b), benzamide (c) and biphenyl-2,3-diol (d) transducers. These are plotted using the data in the third column of the heat-maps in Figure 10.4 as the average, and the error bars as SD from measurements taken from three independent cell-free reactions on the same day (RFU: Relative Fluorescence Unit).

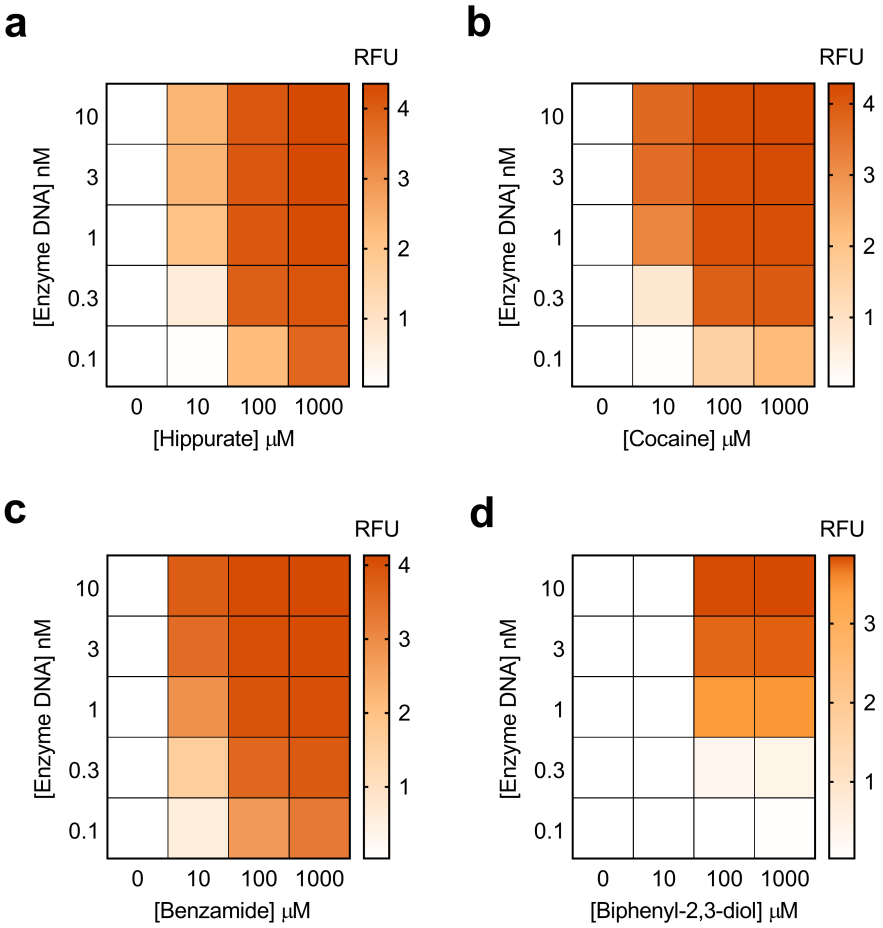


Figure 10.15 **Weighted transducers model results.** The model simulations for experimental conditions presented in Figure 10.4. (a,b,c,d) Heat-maps representing model simulations for weighted transducers at different concentrations of input molecules and enzymes DNA for hippurate (a), cocaine (b), benzamide (c) and biphenyl-2,3-diol (d).

10.6.6 Binary clustering experiments

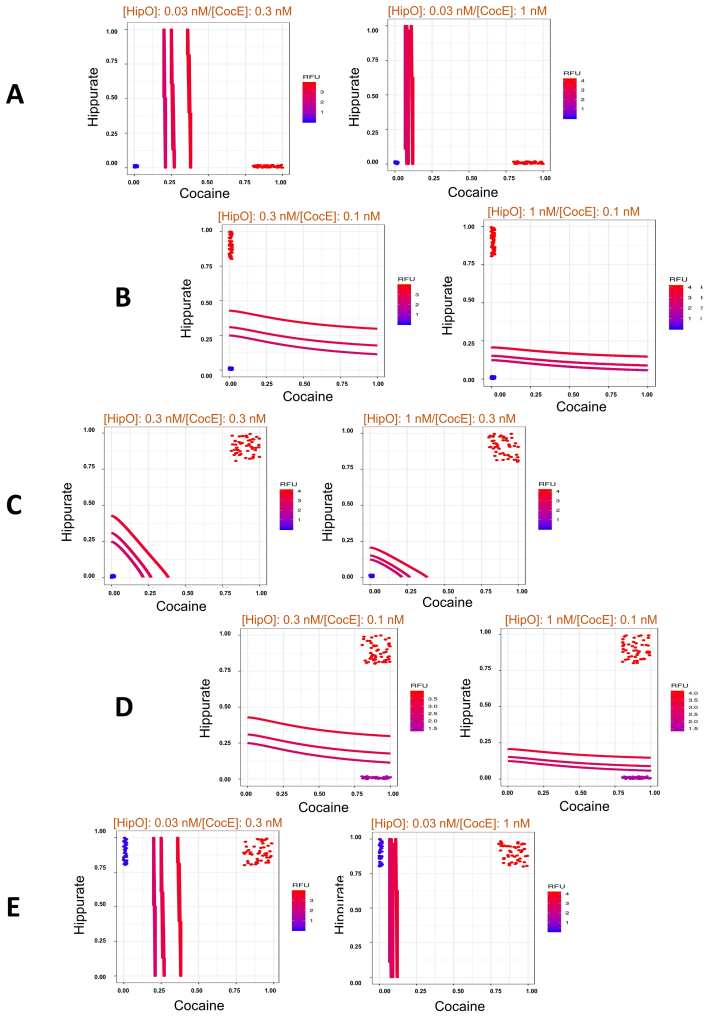


Figure 10.16 **Five different binary classification problems using a metabolic perceptron for hippurate and cocaine.** (**A** to **E**). For each problem, the scatter plot shows multiple data points that represent a combination of input values of cocaine and hippurate. The concentrations for those points are sampled between 0 and 2 µM for low values and 80 µM and 100 µM for high values. The data points in each problem belong to two different sets that can be separated by a threshold line into two separate clusters. The trained model is then used to identify weights needed to be applied to the weighted transducers such that a decision threshold 'd' classifies the two clusters into red (ON, >d) or blue (OFF, <= d). The threshold lines shown in the plots represent three iso-fluorescence lines that successfully classify the data into the binary categories: ON and OFF.

10.6.7 Classifier modeling

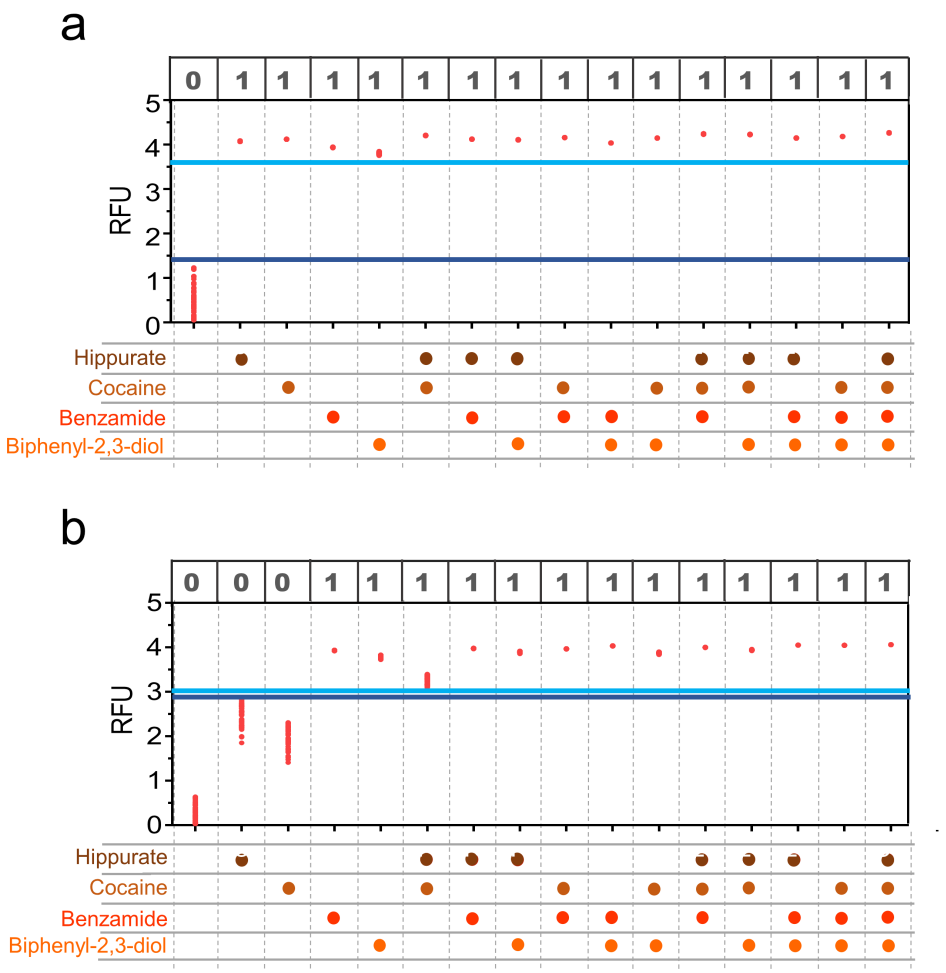


Figure 10.17 **Model simulations for classifiers in Figure 10.6.** Predictions associated with **a** the full OR classifier (Figure 10.6c) and **b** the first calculation for "[cocaine (C) AND hippurate (H)] OR benzamide (B) OR biphenyl-2,3-diol (F)" classifier with 0.1 nM HipO weight with (instead of 0.03 as experimentally tested and presented in Figure 10.6d). In order to perform the clustering, we sampled values uniformly within the stated ranges ([0, 2 µM] for low values and [80 µM, 100 µM] for high values). We then simulated the results to assess the robustness of our designs. Two blue lines refer to the thresholds separating "OFF" and "ON" states. The panel of "OFF" and "ON" at the top of the plots are the expected outputs. (RFU: Relative Fluorescence Unit).

10.6.8 Further experimental characterization

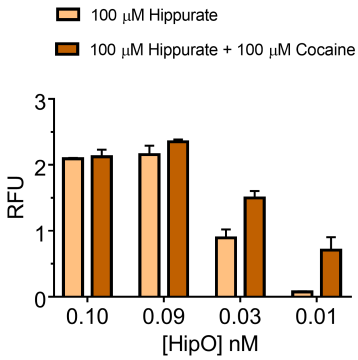


Figure 10.18 **Further characterization of HipO enzyme (hippurate transforming enzyme) at lower concentrations of the enzyme and** 100 µM **hippurate.** HipO enzyme which for its weight led to higher signals than predicted, needed to be further characterized at concentrations lower than the minimum concentration used for the weighted metabolic circuits (0.1 nM). For this characterization, this figure shows the effect of 100 µM hippurate input alone and its additive effect when coupled with 100 µM cocaine at the weight (CocE enzyme concentration) of 0.1 nM. All data are the mean and the error bars are the standard deviation of normalized values from measurements taken from two or three independent cell-free reactions on the same day. (RFU: Relative Fluorescence Unit).

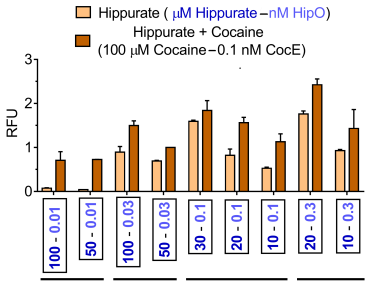


Figure 10.19 **Exploring Hippurate-Cocaine ON-OFF behavior with different weights and input concentrations for hippurate.** All these experiments were done while Cocaine is at a concentration of 100 µM and weight of 0.1 nM CocE. The beige bars are for hippurate (µM Hippurate – nM HipO) and the orange bars are for Hippurate (µM Hippurate – nM HipO) + Cocaine (100 µM Cocaine – 0.1 nM CocE) as inputs. All data are the mean and the error bars are the standard deviation of normalized values from measurements taken from two independent cell-free reactions on the same day. (RFU: Relative Fluorescence Unit).

10.6.9 Design of a multi-layer perceptron

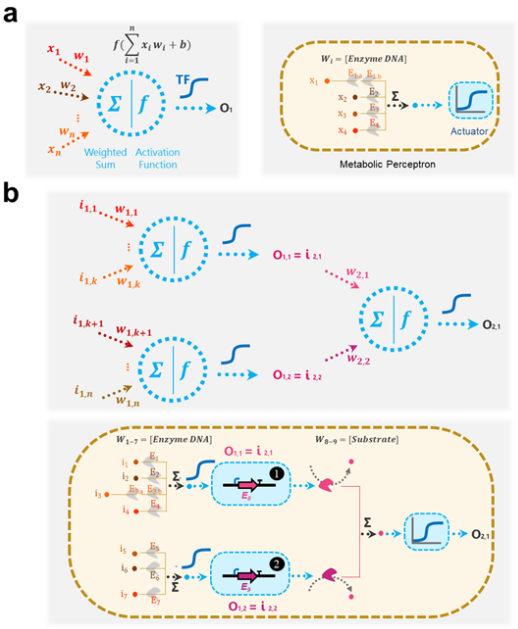


Figure 10.20 **Strategies for multi-layer perceptron implementation.** The schematic presents how computation is performed in a single-layer perceptron: inputs (x_{i-n}) are converted into a common metabolite using enzymes that allow for weighting (w_i) each input (x_i) individually. The common metabolite is then converted into output O_1 using a non-linear activation layer (using a transcription factor =TF). *Right:* A single-layer metabolic perceptron composed of multiple input metabolites (x_{1-4}) and metabolic enzymes (E_{1-4}) transforming the inputs into a common metabolite. The common metabolite then activates the gene expression, representing the actuator function. (b) The schematic presents how computation is performed in a multi-layer perceptron (Top) and a possible implementation of a multi-layer metabolic perceptron (Bottom). In a multi-layer perceptron, the outputs of the first perceptron layer are used as inputs for the second layer. We suggest a potential strategy for such implementation. (1) A TF actuator outputs enzyme E_8 ($O_{1,1}$) from the first layer that behaves as an input ($I_{2,1}$) for the second layer, in turn producing a metabolite needed as effector in the next perceptron layer. (2) Similarly, another TF actuator outputs enzyme E_9 ($O_{1,2}$) from the first layer that behaves as an input ($I_{2,2}$) for the second layer, also producing the same effector metabolite needed in the next perceptron layer. Weights on the second perceptron layer can be applied by tuning the concentrations of the substrate metabolites for E_8 and E_9. This strategy is the converse of what we did in the first layer, where enzyme DNA concentrations were weights and input metabolites were '0' or '1'. Here, the enzymes E_8 and E_9 are '0' or '1', as they are outputs from sigmoidal functions, whereas the metabolite concentrations are the weights.

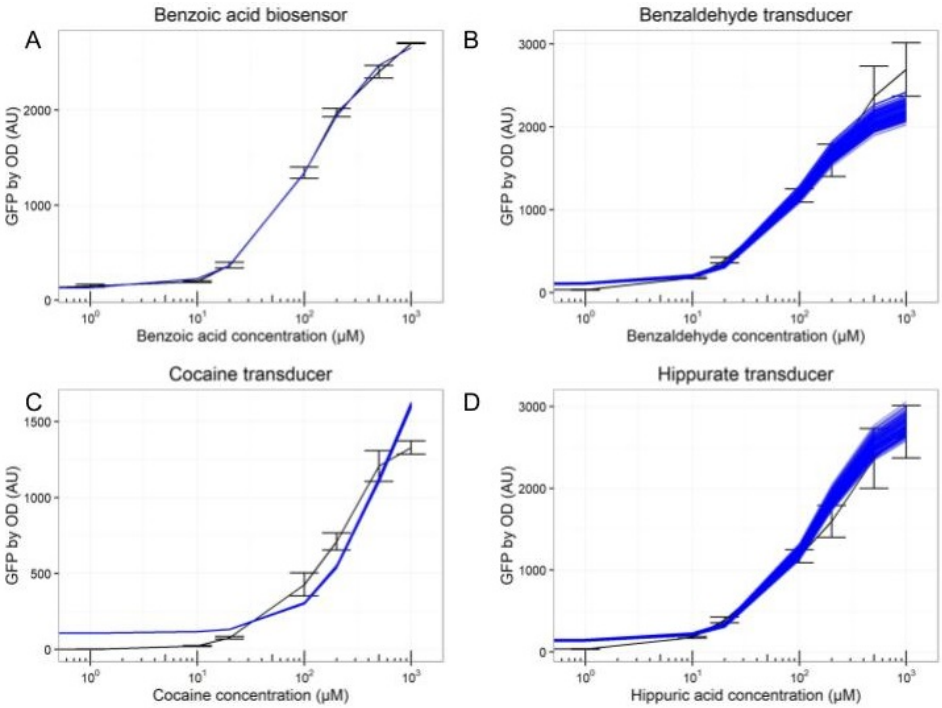


Figure 10.21 **Simulations from the random sampling of estimated parameters in whole-cell system.** Representation of the experimental data with SEM (n = 3) in black, and in blue, the results from 100 simulations of the model with parameters drawn from the final parameters estimation without refitting. The combination of various parameters within our estimations correctly recapitulates the data. **A** benzoate actuator, **B** benzaldehyde transducer, **C** cocaine transducer, and **D** hippurate transducer. Scripts provided in GitHub also allow for visualization of those results for each axis of the adder in Figure 10.2.

10.6.10 Random sampling for model evaluation

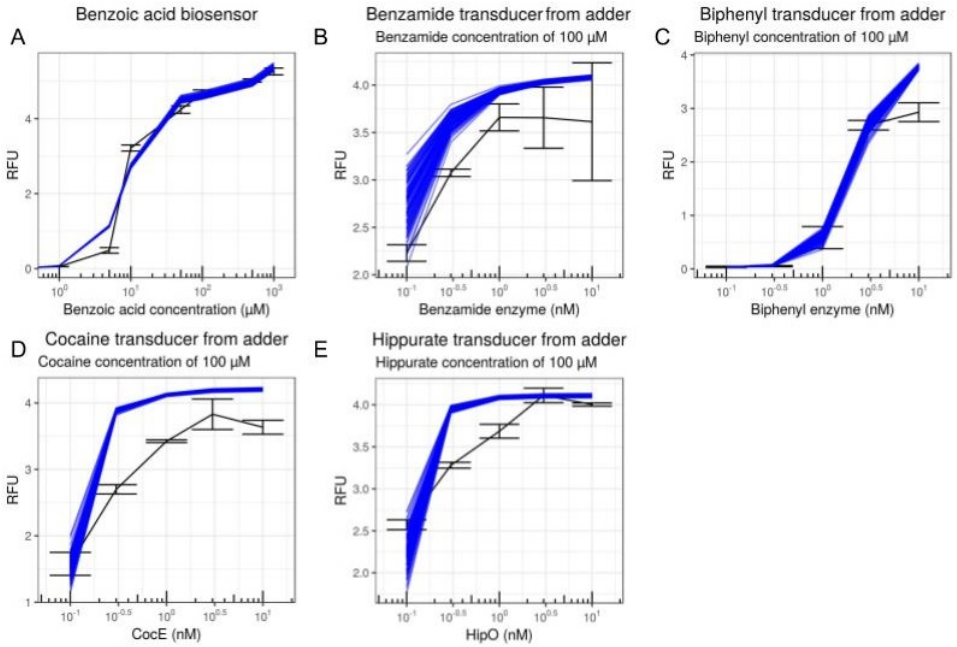


Figure 10.22 **Simulations from the random sampling of estimated parameters in the cell-free system.** Representation of the experimental data with SEM (n = 3) in black, and in blue, the results from 100 simulations of the model with parameters drawn from the final parameters estimation without refitting. The combination of various parameters within our estimations correctly recapitulates the data. **A** benzoate actuator, **B** benzamide transducer, **C** biphenyl-2,3-diol transducer, **D** cocaine transducer, and **E** hippurate transducer. The simulation of the transducers were performed with 100 µM of the input metabolites as will be used in the classifier experiments. Scripts provided in GitHub also allow for the visualization of those results for other axis of the various heat-maps in Figure 10.4. (RFU: Relative Fluorescent/expression Unit of GFP).

10.6.11 Goodness of fit scores

Score	Correlation	Weighted R squared	R squared	Error percentage	Fit or prediction
Actuator	0.999	0.999	0.999	NA	Fit
Benzaldehyde transducer	0.995	0.992	0.980	NA	Fit
Hippurate transducer	0.997	0.990	0.983	NA	Fit
Cocaine transducer	0.965	0.950	0.924	NA	Fit
Adder - complete	0.958	0.982	0.916	16.8 %	Fit (on inducer =0) and prediction
Adder - both inputs present	0.947	0.931	0.889	15.3 %	Prediction

Table 10.3 **Goodness of fit scores for the whole-cell models.** The correlation (from the R cor function), Weighted R squared and R squared between the experimental data and the model. Exact definition of the weighted R squared and the R squared are provided in the Methods section, as well as the Root-Mean-Square Deviation (RMSD) that is used to compare models.

Score	Correlation	Weighted R squared	R squared	Error percentage	Fit or prediction
Actuator	0.990	0.999	0.980	NA	Fit
Cocaine Transducer	0.923	0.999	0.574	NA	Fit
Hippurate Transducer	0.984	0.999	0.962	NA	Fit
Benzamide Transducer	0.946	0.991	0.659	NA	Fit
2,3 biphenyl Transducer	0.965	0.998	0.762	NA	Fit

Table 10.4 **Goodness of fit scores for the cell-free models.** The correlation (from the R cor function), Weighted R squared and R squared between the experimental data and the model. Exact definition of the weighted R squared and the R squared are provided in the Methods section, as well as the RMSD that is used to compare models.

10.6.12 Model parameters

Parameter	**Mean Value ± 95% Confidence Interval**
$Hill_a$	2.2 ± 0.1
Km	$8.40 \pm 9.e^{-3}$
Fc	$137 \pm 1.84(sd : 9.41)$
$Basal$	$3.29.e^{-2} \pm 4.e^{-4}(sd : 2.e^{-3})$
Lin	$8.19 \pm 9.3.e^{-2}$
$Range_{HipO}$	488 ± 35
K_{HipO}	0.396 ± 0.022
$K_{hippurate}$	245 ± 29
n_{HipO}	1.82 ± 0.052
$n_{hippurate}$	1.205 ± 0.046
$Range_{CocE}$	337 ± 28
K_{CocE}	0.799 ± 0.00017
$K_{cocaine}$	54.4 ± 5.04
n_{CocE}	1.713 ± 0.055
$n_{cocaine}$	1.44 ± 0.047
$range_{benzamid.enz}$	234 ± 20
$K_{benzamid.enz}$	3.73 ± 0.27
$K_{benzamid}$	48.6 ± 5.5
$n_{benzamid.enz}$	0.683 ± 0.072
$n_{benzamid}$	0.906 ± 0.087
$range_{biphenyl.enz}$	63.7 ± 4.79
$K_{biphenyl.enz}$	8.63 ± 0.31
$K_{biphenyl}$	56.3 ± 4.92
$n_{biphenyl.enz}$	1.25 ± 0.067
$n_{biphenyl}$	3.05 ± 0.192

Table 10.5 **Parameter estimations for cell-free model** Parameters with value ± 95% Confidence Interval (Standard Deviation for fold change (fc) and baseline)

Parameter	Mean Value ± 95% Confidence Interval
$Hill_a$	$1.34 \pm 1.e^{-6}$
Km	$114 \pm 1.e^{-4}$
Fc	$20.6 \pm 3.e^{-5}$
$Basal$	$130 \pm 2.e^{-4}$
$Range_{BenZ}$	$1.1 \pm 1.e^{-6}$
$Range_{HipO}$	$0.787 \pm 1.e^{-6}$
$Range_{CocE}$	$0.201 \pm 2.97.e^{-3}$
E	4.22 ± 0.193
$Ratio$	$0.776 \pm 3.7.e^{-3}$
nr	$1.956 \pm 4.56.e^{-2}$
$Range_{res}$	1.973 ± 0.107

Table 10.6 **Parameter estimations for *in vivo* model** Parameters with value ± 95% Confidence Interval.

Conclusion & perspectives

Design tools for metabolic circuits

The first Part of this thesis was focused on utilizing efficient algorithms for navigating complex combinatorial spaces, applied to the design of synthetic metabolic circuits.
The first application case of interest was bio-retrosynthesis, which tackles the following problem: given a target compound one wishes to produce, what enzymes (and therefore chemical reactions) should be encoded in the genome of the chassis of interest to produce it from the chassis' metabolism? Retrosynthesis algorithms work backwards, by iteratively simplifying a compound into smaller structures that are simpler to produce, until all starting compounds are found to belong to an ensemble of interest (a database of chemicals that can be bought for chemical retrosynthesis, and the metabolism of the chassis organism of interest for bioretrosynthesis). Methods and algorithms from tools previously developed by the team were presented in the Methods Chapter *Enzyme Discovery: Enzyme Selection and Pathway Design* (more specifically in the Pathway Design part which is my main contribution) and Chapter *Molecular structures enumeration and virtual screening in the chemical space with RetroPath2.0* (in which I developed a set of reaction rules for organic molecules isomer enumeration). In those two examples, the biochemical reactions catalyzed by enzymes are described using reaction rules, learned from data taking into account more or less of the chemical context around the reaction center, allowing us to encode enzymatic promiscuity. One of the main limits of the tools presented in these two chapters is the brute force algorithm that is used to perform the search, as such a huge combinatorial space necessitates better algorithms.
This is the whole aim of the RetroPath3.0 software. While the way the reactions are encoded remained identical, the search algorithm was drastically changed using a similarity-guided Monte Carlo Tree Search (MCTS). MCTS works by building its search tree iteratively, balancing exploration and exploitation and evaluating potential pathways of interest through random sampling to assess their value for the aim of the search. Our algorithm was guided by a score that encompasses both the likelihood that the chemical reaction can apply to the new substrate of interest, and the likelihood that an enzyme exists that can catalyze this reaction. This algorithm was successfully applied to bio-retrosynthesis, allowing us to retrieve pathways described in the literature for 19 compounds out of 20 tested, and

finding a pathway for 83% of cases on a dataset of successful metabolic engineering projects. This tool was designed to be as versatile as possible, with various ways to guide or bias the search towards favorable compounds, chemical reactions or pathways. This modularity and versatility was designed to help expert users input their knowledge into the retrosynthetic search. This is a way to acknowledge the current limitation of purely automatic workflows for metabolic engineering projects, as it is well-known from industrial ventures that expert knowledge from biologists and years of strain optimization are necessary to successfully produce a compound of interest. Moreover, the two datasets that accompany the article (the manually curated golden dataset of exact experimental pathways and the computationally curated lit of successful metabolic engineering projects) will hopefully give a basis to build standards to test new bio-retrosynthesis algorithms in the future of the field. The CASP (Critical Assessment of protein Structure Prediction) challenge is a great example from the field of protein structure prediction that should be taken up by the metabolic engineering community as an example of collaborative community projects for better science.

Despite the algorithmic advances showcased in RetroPath 3.0, a number of challenges still need to be addressed for improving bio-retrosynthesis tools. Two such examples are the quantification of enzyme availability and enzyme promiscuity. More precisely, while RetroPath 3.0 is guided by a score conceived to describe enzyme availability, this metric is broad and does not account for the availability of the enzymatically-catalyzed chemical reaction within the chassis of interest, which could be overcome by accounting for phylogenetic distance to the best candidates, or the quantity of articles describing this function as the end-user of retrosynthetic tools is a biologist that needs precise biological information from referenced publications. Prediction of enzyme promiscuity is another major issue for current bioinformaticians interested in metabolic engineering. Being able to reliably evaluate such a feature would allow retrosynthetic tools to select the allowed degree of promiscuity based on knowledge rather than choosing the most conservative solution. Obviously, reaching this goal would also necessitate intense data reporting and curation efforts from the community in order to have a standard to test promiscuity prediction tools on.

Alternative algorithms to what has been implemented in this thesis could be used, notably to guide the search. In our MCTS implementation, best chemical reactions are scored through chemical similarity, but should ideally be ranked according to their capacity to help produce compounds of interest: rather than evaluating whether the rule is realistic or not as was done here, something more relevant to retrosynthesis would be to know if the rule allows us to solve the search for this compound. An example of other guiding strategies comes from the game of Go. The latest version of AlphaGo [82] combines two major evolution: it uses MCTS to evaluate the state of a Go board, and reinforcement learning to guide the search. For guiding the selection of moves to apply in a given position of pieces on the board, the authors use a neural network to guide towards the most promising moves. More precisely, in the latest version, they use reinforcement learning to train this neural network: moves that lead to wins for the players are more likely to be played later in the game, contrary to the previous version that trained this neural network on a dataset of Go games played by human players. The reinforcement-guided version

beats the human games-guided one 100 to 0. However, both those approaches are complicated to implement in metabolic engineering at the moment. First of all, given our chemical knowledge on enzymatic reactions, it is not because a move applies *in silico* that the chemical reaction can be performed experimentally, whereas this issue does not arise in Go or chess. Secondly, huge datasets for training are not available. A dataset similar to the US patent database could already provide a huge leap forward in the metabolic engineering community.
In order to run such retrosynthesis algorithms for designing metabolic circuits, which require a detectable output, a detectable compounds dataset is required. I therefore provide, make publicly available and regularly update a dataset of detectable metabolites, presented in Chapter 4.
With Chapter 5, I tackled another aspect of circuit design and combinatorial space exploration. The question of this Chapter was the following: given a protein of interest (but this could be a metabolite, as long as it is detectable in a high throughput manner), how do we select components of the buffer to maximize its production? While the combinatorial space is much too large to be explored exhaustively, active learning algorithms, which suggest the next round of experiments to optimize a metric of interest, are perfectly adapted to such a problem. Therefore, I developed an active learning method, coupled with liquid handling robots for experiments, to optimize protein production in cell-free extracts. Such methods will become more and more common as robots become easier to pilot and cheaper. I therefore hope the method I helped develop will become broadly adopted to tackle the lysate efficiency issue of cell-free extracts. The conceptual question tackled here was algorithmically similar to the problem of retrosynthesis, as presented in the introduction. Moreover, from the practical standpoint, improving lysate quality leads to better experimental results when testing the designed circuits in cell-free systems. The method presented in this thesis can therefore be used to improve metabolic circuits implemented in cell-free extracts, as well as production of any protein that is detectable in a high-throughput manner.

Analysis and modeling tools for metabolic circuits

While the first Part of this thesis was focused on building different tools to design metabolic circuits, the second Part was focused on analysis and modeling tools for such circuits.
The first issue for analyzing circuits is that their output needs to be detected. This has been tackled in the design part as I ensure that the last compound of the circuits is a metabolite present in my detectable compounds dataset. The first Chapter of this second Part reviews methods to build biosensors for such detectable output, and notably transcriptional biosensors, to allow reporting of signal from the circuit. In Chapter 7, we saw the development and modeling of an *in vivo* biosensor for pinocembrin and naringenin. More precisely, I observed in our data that chang-

ing plasmid copy number of our biosensors not only modified the fold change of the biosensor, as is to be expected, but also their sensitivity. I therefore chose an adapted modeling strategy based on a modified Hill equation, that accounts for plasmid copy number through transcription factor and binding sites numbers, which is our system's degree of freedom. After fitting this model on available data and verifying parameter consistency using sampling from parameter estimations, the model was then used to suggest modifications for changing the biosensor behavior to attain desired criteria, such as higher or lower sensitivity, by modifying plasmid copy number, DNA - transcription factor binding strength or transcription factor - inducer binding strength. One of the factors of interest that was observed during this work but not modeled was the effect of resource competition, where very high plasmid copy number seemed to be too demanding on cellular resources. This was investigated more in depth in the chapters that followed.

While the work presented in the previous Chapter was performed *in vivo*, my colleagues and I identified several advantages of working in cell-free systems for the rest of the design and analysis of synthetic metabolic circuits, and notably the fine-tuning of enzyme concentration that proved paramount to the success of our most complex circuits. I therefore presented in Chapter 8 a review on cell-free models for state of the art modeling in cell-free systems before my last 2 chapters that implemented metabolic circuits in cell-free systems.

The next step before building complex metabolic circuits is to implement simple functions, which we call transducers, that convert an input of interest into a detectable one, using tools presented in Part I. I present results from this approach in Chapter 9. First, a biosensor is developed and optimized, tuning both transcription factor DNA and reporter DNA. My first contribution to this work was to model this assay using Hill equations. Then, two transducers were implemented in those cell-free systems, for hippuric acid and cocaine. A major point of interest in those two transducers for modeling was that the signal peaked before decreasing at intermediate enzyme DNA concentrations, showing that resource competition has an important role in our observed effects. Therefore, I proceeded in two steps, by first modeling the hippuric acid transducer including resource competition, and applying the same framework to cocaine, accounting for differences in promoter strengths. This modeling was then verified by analyzing a shift in peak signal of the hippurate transducer when varying transcription factor DNA, thereby increasing resource competition. Therefore, this Chapter presents two interesting additions to the biosensor modeling presented in Chapter 7, as it accounts for the effect of resource competition, and includes and models novel steps towards the building of our synthetic circuits.

Finally, the last Chapter of this thesis builds on all previous developments to develop synthetic metabolic circuits, implementing more sophisticated circuitry and calculations. Design was performed using tools presented in the first Part of this thesis. In a first modeling step, the biosensors and metabolic transducers were modeled using custom Hill functions. Then, in a second step, behavior when combining those transducers was predicted *in silico* and tested *in vivo*. Our aim was first to build weighted adders, which can then be modified to make a perceptron, the most basic machine learning algorithm, which can essentially be implemented with a weighted sum followed by a sigmoidal function. The model was used to predict

the enzyme quantities necessary to achieve the classifiers we wished to implement in our cell-free perceptrons, before experimental implementation of those classifiers in cell-free systems. Therefore, my contribution to this project involved data analysis, modeling and prediction for further experimental designs. This project builds on previously developed tools to suggest a novel way to perform computation in synthetic biology.

In Chapters presented in the Analysis and Modeling part of my thesis, different methods were chosen, each best adapted to the question at stake: effect of copy number, of burden, of enzyme DNA concentration... The main idea driving each of those implementations was, as presented in the Introduction of this thesis, to model the system's degree of freedom that I wanted to work on in the project of interest. While one could argue all those projects could have been merged into one, using detailed mechanistic modeling, I believe the questions arising from such detailed modeling - parameters choice, fitting, constraining - render it inconvenient at the moment for design of practical applications as was the aim in this thesis. While processes such as binding, unbinding of inducer to transcription factor, transcription factor to DNA, and all other phenomenons we currently know in transcription and translation could (and often have) been modeled, from a practical standpoint, those could not be controlled in our setting of choice, and including them would therefore surely add more uncertainty to our predictions while not necessarily improving prediction quality. For example, many different measures of enzyme catalytic strength are available in enzymatic databases, spanning order of magnitudes in some cases, and usually measured *in vitro*: while making the best of this information is a scientific endeavor that is both interesting and useful, methods based on empirical modeling do seem more practical when predicting circuit outcomes is the objective. This is however a very pragmatic point of view. For example, while circuit burden and resource usage were formerly not included in modeling projects, it is now becoming more and more common place to do so as it has been recognized a major problem in circuit development in the synthetic biology community. A number of topics otherwise interesting are not currently investigated by the community, in part because of lack of data in published articles on this topic. For example, while we know resource competition affects short-term circuit output, we also know it affects long term circuit behavior throughout generations due to genetic drift. However, since we do not have data of circuit duration (in generations) in most publications, deriving design principles and modeling for this topic is possible but can hardly be evaluated.

The modeling tools presented in this thesis present limitations, as we have seen previously, both in scope of modeling and predictive power. However, I would like to point out the importance of modeling for reasons beyond the aim of this thesis. Current modeling efforts in the synthetic biology community often lead to useful insights, and probably just as often do not. However, modeling should be valued for itself, as it is a way of formalizing knowledge that allows us as a community to present in a concise and mathematical manner our current knowledge on a topic. Failures and lacks in models allow us to identify our current knowledge gaps and could be a major driver of biological research, by identifying where the limits to our knowledge lie.

Acronyms

E. coli *Escherichia coli*.
4CL coumarate-CoA ligase.

AAM atom–atom mapping.
amu atomic mass units.
aTc anhydrotetracycline.
ATP adenosine triphosphate.
AU Arbitrary Units.

BDA 1,4-Butanediol diacrylate.
BLAST Basic Local Alignment Search Tool.
BRENDA Braunschweig Enzyme Database.

CAS Chemical Abstract Service.
CHI chalcone isomerase.
CHS chalcone synthase.
CoMFA Comparative MolecularField Analysis.
CRISPR clustered regularly interspaced short palindromic repeats.

DBD DNA Binding Domain.
DBTL Design–Build–Test–Learn.
DNA deoxyribonucleic acid.
dNTP deoxyribonucleotide triphosphate.
DOGS Design of Genuine Structures.

EC Enzyme Commission.
ECFP Extended Connectivity FingerPrint.

FBA Flux Balance Analysis.
FIFO First In First Out.
FRET Förster Resonance Energy Transfer.

GFP green fluorescent protein.
GPCR G protein-coupled receptor.
GRAS Generally Recognized As Safe.

HMDB Human Metabolome Database.

InChI IUPAC International Chemical Identifier.
IPTG Isopropyl β-D-1-thiogalactopyranoside.

LB Luria Broth.
LBD Ligand Binding Domain.
LC-MS liquid chromatography-mass spectrometry.
LOO Leave-One-Out.

MCS Maximum Common Substructure.
MCTS Monte-Carlo Tree Search.
MDL Molecular Design Limited.
MIC Minimum Inhibitory Concentration.
mRNA messenger RNA.
MS Mass Spectrometry.

NAD nicotinamide adenine dinucleotide.
NADP Nicotinamide Adenine Dinucleotide Phosphate.
NTP nucleoside triphosphate.

OD optical density.
ODE ordinary differential equation.

PAL phenylalanine ammonia lyase.
PEG polyethylene glycol.
PSSM position-specific scoring matrices.

QSAR Quantitative Structure Activity Relationship.
QSPR Quantitative Structure Property Relationship.

RBS ribosome binding site.
RDT Reaction Decoder Tool.
REU relative expression unit.
RFP red fluorescent protein.

RMSD Root-Mean-Square Deviation.
RNA ribonucleic acid.
RNAP RNA polymerase.

SBML Systems Biology Markup Language.
SBOL Synthetic Biology Open Language.
SEMP Sensing-Enabling Metabolic Pathways.
sfGFP super-folder GFP.
SMARTS SMILES arbitrary target specification.
SMILES Simplified Molecular Input Line Entry System.
SQL Structured Query Language.

TAL tyrosine ammonia lyase.
TCA tricarboxylic acid cycle.
TF transcription factor.
tRNA transfer ribonucleic acid.
TX-TL transcription-translation.

List of Tables

1.1 Selenzyme results for EC 1.1.1.37 . . . 37

2.1 Number of generated alkane isomers by canonical augmentation algorithm and isomer transformation algorithm. . . . 56
2.2 Metabolome completion. . . . 68
2.3 List of the 19 SMARTS rules that were used in this study. . . . 78
2.4 Solution space reduction arguments or rules. . . . 79

3.1 Average branching factor for chosen sets . . . 85
3.2 Average branching factor for individual diameters . . . 104
3.3 Toxicity biased results . . . 104
3.4 RP3 configuration . . . 105
3.5 Golden dataset structures . . . 107

4.1 Detectable compounds dataset specifications . . . 119
4.2 Detectable compounds data sources . . . 123

5.1 Sequence of the plasmid used in this study . . . 141

6.1 Successful biosensors . . . 156

7.1 Pinocembrin and naringenin parameters . . . 167
7.2 Parameters for the time-course model . . . 168
7.4 Biosensors characteristics . . . 173
7.3 Flavonoids similarity to pinocembrin . . . 182
7.5 Strains and plasmids for pinocembrin biosensor . . . 183
7.6 Primers list for pinocembrin biosensor . . . 183
7.7 Copy numbers of the used plasmids for pinocembrin biosensor . . . 184
7.8 Pinocembrin-sensor constructs list . . . 185

8.1 Deterministic models developed to understand cell-free . . . 193

9.1 *In vivo* transcription and translation rates. . . . 226
9.2 Number of RNAP/ ribosomes per DNA/ mRNA strand . . . 227
9.3 Effective translation rates *in vivo* and in cell-free . . . 227
9.4 Enzymes' catalytic constants. . . . 228
9.5 Numerical parameters used during simulations . . . 229
9.6 Fluorescence results from calibration of TF and reporter plasmids. 239
9.8 Benzoate concentration in commercial beverages determined from three replicates of our cell-free biosensor and LC-MS. . . . 240
9.9 Benzoic acid sensor shows minimal activation in response to human urine without HipO metabolic transducer. . . . 240
9.7 Fluorescence results from calibration of HipO and CocE metabolic transducer plasmids. . . . 241

9.10 Endogenous hippuric acid concentration in human urine samples determined from three replicates of our cell-free biosensor and LC-MS. 241

10.1 Chemical structures for compounds used in this work. 268
10.2 Sequences identifiers for parts used in this work. 269
10.3 Goodness of fit scores for the whole-cell models 283
10.4 Goodness of fit scores for the cell-free models 283
10.5 Parameter estimations for cell-free model 284
10.6 Parameter estimations for *in vivo* model 285

List of Figures

0.1 Price per base of DNA Sequencing and synthesis 2
0.2 Evolution of the number of publications in synthetic biology . . . 5
0.3 Principles of active learning 10
0.4 Principles of Monte Carlo Tree Search 10

1.1 Enzyme selection, cluster discovery, and pathway design 27
1.2 Computing reaction similarity 30
1.3 Selenzyme result for EC 1.1.1.37 39
1.4 Rules and SMARTS for reaction 1.1.1.37 42
1.5 Enumerated pathways for 1,4-Butanediol production 47

2.1 Execution time for each tested software on the enumeration of alkane isomers. 54
2.2 Reaction rules for canonical augmentation of carbon skeletons . . 55
2.3 Identical rules 57
2.4 Isomer transformation rule set 58
2.5 Rules before solution space reduction due to valence and structure considerations 59
2.6 Distributions of predicted T_g values for enumerated isomers and for isomers found in PubChem with varying Tanimoto threshold. . . . 63
2.7 Distributions of predicted T_g values for enumerated isomers and for isomers found in PubChem 64
2.8 Representation of monomers and isomers in the chemical space . . 74
2.9 Evolution of predicted activities 75
2.10 Reduced isomer transformation rule set. 75
2.11 RetroPath2.0 rules and corresponding SMARTS for reaction 2.6.1.93 at various diameters. 76
2.12 RetroPath2.0 KNIME workflow. Inner view of the "Core" node where the computation takes place. 77

3.1 Presentation of the algorithm and the chemical scoring scheme employed 86
3.2 Results of the RetroPath suite against the golden dataset to identify the experimental pathway. 88
3.3 Impact of guidance scheme on retrieval performance of RetroPath 3.0 89
3.4 Database sped-up retrosynthetic search 92
3.5 Extending a previous search 93
3.6 Impact of biological score cut-off on retrieval performance of RetroPath 3.0 108
3.7 Impact of chemical score cut-off on retrieval performance of RetroPath 3.0 109

3.8 Impact of biochemical score cut-off on retrieval performance of RetroPath 3.0 . . . 110
3.9 Impact of allowed rule diameters on retrieval performance of RetroPath 3.0 . . . 110
3.10 Impact of expansion width on retrieval performance of RetroPath 3.0 111
3.11 Impact of minimal visits count on retrieval performance of RetroPath 3.0 . . . 112
3.12 Impact of rollout depth on retrieval performance of RetroPath 3.0 113
3.13 Impact of exploration constant value on retrieval performance of RetroPath 3.0 . . . 113
3.14 Impact of virtual visits on retrieval performance of RetroPath 3.0 114
3.15 Impact of penalty on retrieval performance of RetroPath 3.0 . . . 115
3.16 Impact of reward on retrieval performance of RetroPath 3.0 . . . 116

4.1 Experimental and sensing characteristics of detectable compounds 123

5.1 Active learning loop to explore the composition of a cell-free system 129
5.2 One-step method to predict protein yield in cell-free systems . . . 131
5.3 Preliminary calibration of the cell-free composition . . . 140
5.4 The choice of 102 cell-free compositions for training and testing of our model. . . . 140
5.5 Mutual information analysis . . . 141
5.6 Global comparison between the yields obtained with different lysates 142
5.7 Comparison between the behavior of the local yields measured with different lysates and the yields measured with the $lysate_{ORI}$ . . . 142
5.8 A decrease in ribosome availability is sufficient to explain the saturation of the yields with $Lysate_{Spectinomycin}$ . . . 143
5.9 Absolute measurements in cell-free reaction . . . 144

6.1 Graphical abstract for biosensor review . . . 152
6.2 Different strategies to develop a TF based biosensor for a given metabolite. . . . 154
6.3 Pinocembrin cell-free biosensor. . . . 159

7.1 Pinocembrin biosynthesis pathway . . . 164
7.2 Pinocembrin biosensor module . . . 165
7.3 Dose responses of different biosensor constructs . . . 171
7.4 Effect of copy number variations of fold change . . . 172
7.5 Model fitting of pinocembrin data . . . 176
7.6 Model fitting of naringenin data - no correction . . . 176
7.7 Model fitting of naringenin data . . . 177
7.8 Copy number model predictions . . . 179
7.9 Time course modeling of construct 357 . . . 186
7.10 Growth model fitting to construct 357 . . . 187
7.11 Growth rate of constructs with varying resistance markers . . . 188
7.12 Time-course model for pinocembrin. . . . 190
7.13 Time-course model for naringenin. . . . 190

8.1 Production of a constitutively expressed gene in cell-free . . . 194
8.2 Resource competition in cell-free . . . 196

9.1 A modular design workflow for engineering scalable cell-free biosensors. . . . 202
9.2 Calibration of sensor and output modules for benzoate detection. 203
9.3 Modeling titration of transcription factor and reporter plasmids. . 204

9.4 Expanding the chemical detection space of cell-free biosensors by plugging various metabolic transducers into an optimized sensor module. . . . 206
9.5 Detecting benzoic acid, hippuric acid, and cocaine in complex samples. . . . 209
9.6 Modeling metabolic transducer behavior for HipO and CocE . . . 217
9.7 Superfolder-GFP expression with J23101 and pBEST promoter (OR2-OR1-Pr) . . . 218
9.8 Model-predicted shift in HipO concentration for peak biosensor signal at high concentrations of TF plasmid and inducer. . . . 219
9.9 Time course of the benzoic acid biosensor response to varying concentrations of inducer. . . . 230
9.10 Time course of the hippuric acid biosensor response to varying concentrations of inducer. . . . 230
9.11 Time course of the cocaine biosensor response to varying concentrations of inducer. . . . 231
9.12 Time course of the benzoic biosensor response to 1x and 0.1x beverages. . . . 232
9.13 Interference of 0.1x and 1x beverages on cell-free reaction with constitutive sfGFP plasmid. . . . 233
9.14 Hill plot fit of a standard gradient of benzoic acid to calibrate sensor. 234
9.15 Interference of human urine on cell-free reaction with constitutive sfGFP plasmid. . . . 234
9.16 Hill plot fit of a standard gradient of hippuric acid to calibrate sensor. 235
9.17 Correlation between cell-free biosensor and LC-MS measurements of endogenous hippuric acid levels in human urine. . . . 235
9.18 Detection of cocaine spiked into clinical urine samples with sfGFP output module. . . . 236
9.19 Cell-free reactions accumulate autofluorescent products in the GFP channel even in the absence of DNA. . . . 236
9.20 Use of firefly luciferase as an output module enhances benzoic acid sensor fold change. . . . 237
9.21 Comparison of benzoic acid and cocaine biosensor expression in response to urinary cocaine gradient. . . . 238

10.1 Whole-cell actuator and metabolic transducers. . . . 247
10.2 Whole-cell metabolic adder of hippurate and benzaldehyde. . . . 249
10.3 Cell-free actuator and metabolic transducers. . . . 251
10.4 Cell-free weighted transducers characterized by varying the concentration of the enzyme DNA. . . . 254
10.5 Cell-free adder . . . 255
10.6 Cell-free perceptron enabling development of classifiers. . . . 258
10.7 Feedback-loop circuit design of the benzoate actuator. . . . 270
10.8 Comparison of the maximum signals of whole-cell circuits. . . . 270
10.9 2D plots for the data presented in heat-map in Figure 10.2b . . . 271
10.10 Examining the effect of resource competition versus enzyme efficiency on the whole-cell cocaine transducer. . . . 272
10.11 Examining the effect of resource competition versus enzyme efficiency on the whole-cell metabolic adder. . . . 272
10.12 The specific growth rate (μ) values of the whole-cell circuits presented in Figure 10.1. . . . 273
10.13 The specific growth rate (μ) values of the whole-cell adder presented in Figure 10.2b. . . . 274
10.14 The dose-response of cell-free transducers to different concentrations of the associated enzyme DNA (weights) for weighted transducers. 275
10.15 Weighted transducers model results. . . . 276

10.16 Five different binary classification problems using a metabolic perceptron for hippurate and cocaine. 277
10.17 Model simulations for classifiers in Figure 10.6. 278
10.18 Further characterization of HipO enzyme 279
10.19 Exploring Hippurate-Cocaine ON-OFF behavior with different weights and input concentrations for hippurate. 279
10.20 Strategies for multi-layer perceptron implementation. 280
10.21 Simulations from the random sampling of estimated parameters in whole-cell system. 281
10.22 Simulations from the random sampling of estimated parameters in the cell-free system. 282

Bibliography

[1] J.D. Watson and F.H.C. Crick. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature*, 171:737–738, 1953.

[2] Erwin Schrödinger. *What is Life? The Physical Aspect of the Living Cell*. Cambridge University Press, 1944.

[3] Human Genome Project. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, February 2001.

[4] Rob Carlson. Time for new dna synthesis and sequencing cost curves, 2014.

[5] William M. Shaw, Hitoshi Yamauchi, Jack Mead, Glen-Oliver F. Gowers, David J. Bell, David Öling, Niklas Larsson, Mark Wigglesworth, Graham Ladds, and Tom Ellis. Engineering a model cell for rational tuning of GPCR signaling. *Cell*, 177(3):782–796, April 2019.

[6] J Monod. The growth of bacterial cultures. *Annual Review of Microbiology*, 3(1):371–394, October 1949.

[7] D. Ewen Cameron, Caleb J. Bashor, and James J. Collins. A brief history of synthetic biology. *Nature Reviews Microbiology*, 12(5):381–390, April 2014.

[8] Michael Elowitz and Stanislas Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335–338, 2000.

[9] Timothy S Gardner, Charles R Cantor, and James J Collins. Construction of a genetic toggle switch in Escherichia coli. *Nature*, 403(20), 2000.

[10] Jean-Baptiste Lugagne, Sebastián Sosa Carrillo, Melanie Kirch, Agnes Köhler, Gregory Batt, and Pascal Hersen. Balancing a genetic toggle switch by real-time feedback control and periodic forcing. *Nature Communications*, 8(1), November 2017.

[11] Attila Becskei and Luis Serrano. Engineering stability in gene networks by autoregulation. *Nature*, 405(6786):590–593, June 2000.

[12] D. G. Gibson, G. A. Benders, C. Andrews-Pfannkoch, E. A. Denisova, H. Baden-Tillson, J. Zaveri, T. B. Stockwell, A. Brownley, D. W. Thomas, M. A. Algire, C. Merryman, L. Young, V. N. Noskov, J. I. Glass, J. C. Venter, C. A. Hutchison, and H. O. Smith. Complete chemical synthesis, assembly, and cloning of a mycoplasma genitalium genome. *Science*, 319(5867):1215–1220, February 2008.

[13] Julius Fredens, Kaihang Wang, Daniel de la Torre, Louise F. H. Funke, Wesley E. Robertson, Yonka Christova, Tiongsun Chia, Wolfgang H. Schmied, Daniel L. Dunkelmann, Václav Beránek, Chayasith Uttamapinant, Andres Gonzalez Llamazares, Thomas S. Elliott, and Jason W. Chin. Total synthesis of escherichia coli with a recoded genome. *Nature*, 569(7757):514–518, May 2019.

[14] Jee Loon Foo and Matthew Wook Chang. Synthetic yeast genome reveals its versatility. *Nature*, 557(7707):647–648, May 2018.

[15] Youngha Ryu and Peter G Schultz. Efficient incorporation of unnatural amino acids into proteins in escherichia coli. *Nature Methods*, 3(4):263–265, March 2006.

[16] Dong Niu, Hong-Jiang Wei, Lin Lin, Haydy George, Tao Wang, I-Hsiu Lee, Hong-Ye Zhao, Yong Wang, Yinan Kan, Ellen Shrock, Emal Lesha, Gang Wang, Yonglun Luo, Yubo Qing, Deling Jiao, Heng Zhao, Xiaoyang Zhou, Shouqi Wang, Hong Wei, Marc Güell, George M. Church, and Luhan Yang. Inactivation of porcine endogenous retrovirus in pigs using CRISPR-cas9. *Science*, 357(6357):1303–1307, August 2017.

[17] George M. Church, Michael B. Elowitz, Christina D. Smolke, Christopher A. Voigt, and Ron Weiss. Realizing the potential of synthetic biology. *Nature Reviews Molecular Cell Biology*, 15(4):289–294, March 2014.

[18] David L. Shis, Faiza Hussain, Sarah Meinhardt, Liskin Swint-Kruse, and Matthew R. Bennett. Modular, multi-input transcriptional logic gating with orthogonal LacI/GalR family chimeras. *ACS Synthetic Biology*, 3(9):645–651, July 2014.

[19] Tae Seok Moon, Chunbo Lou, Alvin Tamsir, Brynne C. Stanton, and Christopher A. Voigt. Genetic programs constructed from layered logic gates in single cells. *Nature*, 491(7423):249–253, October 2012.

[20] Nina Buffi, Davide Merulla, Julien Beutier, Fanny Barbaud, Siham Beggah, Harald van Lintel, Philippe Renaud, and Jan Roelof van der Meer. Miniaturized bacterial biosensor system for arsenic detection holds great promise for making integrated measurement device. *Bioengineered Bugs*, 2(5):296–298, September 2011.

[21] Xinyi Wan, Francesca Volpetti, Ekaterina Petrova, Chris French, Sebastian J. Maerkl, and Baojun Wang. Cascaded amplifying circuits enable ultrasensitive cellular sensors for toxic metals. *Nature Chemical Biology*, 15(5):540–548, March 2019.

[22] Alexis Courbet, Drew Endy, Eric Renard, Franck Molina, and Jérôme Bonnet. Detection of pathological biomarkers in human clinical samples via amplifying genetic switches and logic gates. *Science Translational Medicine*, 7:289ra83, May 2015.

[23] Ke Yan Wen, Loren Cameron, James Chappell, Kirsten Jensen, David J. Bell, Richard Kelwick, Margarita Kopniczky, Jane C. Davies, Alain Filloux, and Paul S. Freemont. A cell-free biosensor for detecting quorum sensing molecules in p. aeruginosa-infected respiratory samples. *ACS Synthetic Biology*, 6(12):2293–2301, October 2017.

[24] Christian Kemmer, Marc Gitzinger, Marie Daoud-El Baba, Valentin Djonov, Jörg Stelling, and Martin Fussenegger. Self-sufficient control of urate homeostasis in mice by a synthetic circuit. *Nature Biotechnology*, 28(4):355–360, March 2010.

[25] Vincent M Isabella, Binh N Ha, Mary Joan Castillo, David J Lubkowicz, Sarah E Rowe, Yves A Millet, Cami L Anderson, Ning Li, Adam B Fisher, Kip A West, Philippa J Reeder, Munira M Momin, Christopher G Bergeron, Sarah E Guilmain, Paul F Miller, Caroline B Kurtz, and Dean Falb. Development of a synthetic live bacterial therapeutic for the human metabolic disease phenylketonuria. *Nature Biotechnology*, 36(9):857–864, August 2018.

[26] Fuzhong Zhang, James M Carothers, and Jay D Keasling. Design of a dynamic sensor-regulator system for production of chemicals and fuels derived from fatty acids. *Nature Biotechnology*, 30(4):354–359, March 2012.

[27] Stephanie J. Doong, Apoorv Gupta, and Kristala L. J. Prather. Layered dynamic regulation for improving metabolic pathway productivity in escherichia coli. *Proceedings of the National Academy of Sciences*, 115(12):2964–2969, March 2018.

[28] Apoorv Gupta, Irene M Brockman Reizman, Christopher R Reisch, and Kristala L J Prather. Dynamic regulation of metabolic flux in engineered bacteria using a pathway-independent quorum-sensing circuit. *Nature Biotechnology*, 35(3):273–279, February 2017.

[29] A. A. K. Nielsen, B. S. Der, J. Shin, P. Vaidyanathan, V. Paralanov, E. A. Strychalski, D. Ross, D. Densmore, and C. A. Voigt. Genetic circuit design automation. *Science*, 352(6281):aac7341, March 2016.

[30] Amin Espah Borujeni, Daniel Cetnar, Iman Farasat, Ashlee Smith, Natasha Lundgren, and Howard M. Salis. Precise quantification of translation inhibition by mRNA structures that overlap with the ribosomal footprint in n-terminal coding sequences. *Nucleic Acids Research*, 45(9):5437–5448, February 2017.

[31] Priscilla E. M. Purnick and Ron Weiss. The second wave of synthetic biology: from modules to systems. *Nature Reviews Molecular Cell Biology*, 10(6):410–422, June 2009.

[32] Jennifer A N Brophy and Christopher A Voigt. Principles of genetic circuit design. *Nature Methods*, 11(5):508–520, April 2014.

[33] Oliver Purcell and Timothy K Lu. Synthetic analog and digital circuits for cellular computation and memory. *Current Opinion in Biotechnology*, 29:146–155, October 2014.

[34] Mingqi Xie and Martin Fussenegger. Designing cell function: assembly of synthetic gene circuits for cell biology applications. *Nature Reviews Molecular Cell Biology*, 19(8):507–525, June 2018.

[35] Meriem El Karoui, Monica Hoyos-Flight, and Liz Fletcher. Future trends in synthetic biology—a report. *Frontiers in Bioengineering and Biotechnology*, 7, August 2019.

[36] Eleni Karamasioti, Claude Lormeau, and Jörg Stelling. Computational design of biological circuits: putting parts into context. *Molecular Systems Design & Engineering*, 2(4):410–421, 2017.

[37] Stefano Cardinale and Adam Paul Arkin. Contextualizing context for synthetic biology - identifying causes of failure of synthetic biological systems. *Biotechnology Journal*, 7(7):856–866, May 2012.
[38] Domitilla Del Vecchio, Alexander J Ninfa, and Eduardo D Sontag. Modular cell biology: retroactivity and insulation. *Molecular Systems Biology*, 4, February 2008.
[39] M. Carbonell-Ballestero, E. Garcia-Ramallo, R. Montañez, C. Rodriguez-Caso, and J. Macía. Dealing with the genetic load in bacterial synthetic biology circuits: convergences with the ohm's law. *Nucleic Acids Research*, 44(1):496–507, December 2015.
[40] Deepak Mishra, Phillip M Rivera, Allen Lin, Domitilla Del Vecchio, and Ron Weiss. A load driver device for engineering modularity in biological networks. *Nature Biotechnology*, 32(12):1268–1275, November 2014.
[41] Arthur Prindle, Jangir Selimkhanov, Howard Li, Ivan Razinkov, Lev S. Tsimring, and Jeff Hasty. Rapid and tunable post-translational coupling of genetic circuits. *Nature*, 508(7496):387–391, April 2014.
[42] Yu-Yu Cheng, Andrew J. Hirning, Krešimir Josić, and Matthew R. Bennett. The timing of transcriptional regulation in synthetic gene circuits. *ACS Synthetic Biology*, 6(11):1996–2002, September 2017.
[43] Francesca Ceroni, Alice Boo, Simone Furini, Thomas E Gorochowski, Olivier Borkowski, Yaseen N Ladak, Ali R Awan, Charlie Gilbert, Guy-Bart Stan, and Tom Ellis. Burden-driven feedback control of gene expression. *Nature Methods*, 15(5):387–393, March 2018.
[44] Olivier Borkowski, Carlos Bricio, Michela Murgiano, Brooke Rothschild-Mancinelli, Guy-Bart Stan, and Tom Ellis. Cell-free prediction of protein expression costs for growing cells. *Nature Communications*, 9(1), April 2018.
[45] Domitilla Del Vecchio, Aaron J. Dy, and Yili Qian. Control theory meets synthetic biology. *Journal of The Royal Society Interface*, 13(120):20160380, July 2016.
[46] Angel Goñi-Moreno and Pablo I. Nikel. High-performance biocomputing in synthetic biology–integrated transcriptional and metabolic circuits. *Frontiers in Bioengineering and Biotechnology*, 7, March 2019.
[47] Alexis Courbet, Patrick Amar, François Fages, Eric Renard, and Franck Molina. Computer-aided biochemical programming of synthetic microreactors as diagnostic devices. *Molecular Systems Biology*, 14(4):e7845, April 2018.
[48] Rafael Silva-Rocha, Javier Tamames, Vitor Martins dos Santos, and Víctor de Lorenzo. The logicome of environmental bacteria: merging catabolic and regulatory events with boolean formalisms. *Environmental Microbiology*, 13(9):2389–2402, March 2011.
[49] Friedrich C. Simmel, Bernard Yurke, and Hari R. Singh. Principles and applications of nucleic acid strand displacement reactions. *Chemical Reviews*, February 2019.
[50] Jackson O'Brien and Arvind Murugan. Temporal pattern recognition through analog molecular computation. *ACS Synthetic Biology*, 8(4):826–832, March 2019.
[51] L. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, November 1994.
[52] Herbert M. Sauro and Kyung Kim. It's an analog world. *Nature*, 497(7451):572–573, May 2013.
[53] Ramiz Daniel, Sung Sik Woo, Lorenzo Turicchia, and Rahul Sarpeshkar. Analog transistor models of bacterial genetic circuits. In *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, November 2011.
[54] Ramiz Daniel, Jacob R. Rubens, Rahul Sarpeshkar, and Timothy K. Lu. Synthetic analog computation in living cells. *Nature*, 497(7451):619–623, May 2013.
[55] D. Chandran, W.B. Copeland, S.C. Sleight, and H.M. Sauro. Mathematical modeling and synthetic biology. *Drug Discovery Today: Disease Models*, 5(4):299–309, December 2008.
[56] Daniel G Gibson, Lei Young, Ray-Yuan Chuang, J Craig Venter, Clyde A Hutchison, and Hamilton O Smith. Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nature Methods*, 6(5):343–345, April 2009.
[57] Carola Engler and Sylvestre Marillonnet. Golden gate cloning. In *DNA Cloning and Assembly Methods*, pages 119–131. Humana Press, December 2013.
[58] Carola Engler, Ramona Gruetzner, Romy Kandzia, and Sylvestre Marillonnet. Golden gate shuffling: A one-pot DNA shuffling method based on type IIs restriction enzymes. *PLoS ONE*, 4(5):e5553, May 2009.
[59] Philip Shapira, Seokbeom Kwon, and Jan Youtie. Tracking the emergence of synthetic biology. *Scientometrics*, 112(3):1439–1469, July 2017.
[60] Ross Cloney. Building an alliance of biofoundries q and a with paul freemont, 2019.

[61] Nicholas Roehner, Jacob Beal, Kevin Clancy, Bryan Bartley, Goksel Misirli, Raik Grünberg, Ernst Oberortner, Matthew Pocock, Michael Bissell, Curtis Madsen, Tramy Nguyen, Michael Zhang, Zhen Zhang, Zach Zundel, Douglas Densmore, John H. Gennari, Anil Wipat, Herbert M. Sauro, and Chris J. Myers. Sharing structure and function in biological design with SBOL 2.0. *ACS Synthetic Biology*, 5(6):498–506, May 2016.

[62] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, , the rest of the SBML Forum:, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J.-H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novere, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, March 2003.

[63] Ellis Whitehead, Fabian Rudolf, Hans-Michael Kaltenbach, and Jörg Stelling. Automated planning enables complex protocols on liquid-handling robots. *ACS Synthetic Biology*, 7(3):922–932, February 2018.

[64] Filippo Caschera, Ashty S. Karim, Gianluca Gazzola, Anne E. d'Aquino, Norman H. Packard, and Michael C. Jewett. High-throughput optimization cycle of a cell-free ribosome assembly and protein synthesis system. *ACS Synthetic Biology*, 7(12):2841–2853, October 2018.

[65] Jarosław M. Granda, Liva Donina, Vincenza Dragone, De-Liang Long, and Leroy Cronin. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature*, 559(7714):377–381, July 2018.

[66]

[67] V. Noireaux, R. Bar-Ziv, and A. Libchaber. Principles of cell-free genetic circuit assembly. *Proceedings of the National Academy of Sciences*, 100(22):12672–12677, October 2003.

[68] Zachary Z. Sun, Clarmyra A. Hayes, Jonghyeon Shin, Filippo Caschera, Richard M. Murray, and Vincent Noireaux. Protocols for Implementing an Escherichia coli Based TX-TL Cell-Free Expression System for Synthetic Biology. *Journal of Visualized Experiments*, 79, sep 2013.

[69] Ho-Cheol Kim and Dong-Myung Kim. Methods for energizing cell-free protein synthesis. *Journal of Bioscience and Bioengineering*, 108(1):1–4, July 2009.

[70] Emmanuel L. C. de los Santos, Joseph T. Meyerowitz, Stephen L. Mayo, and Richard M. Murray. Engineering transcriptional regulator effector specificity using computational design and in vitro rapid prototyping: Developing a vanillin sensor. *ACS Synthetic Biology*, 5(4):287–295, August 2015.

[71] Quentin M. Dudley, Kim C. Anderson, and Michael C. Jewett. Cell-free mixing of escherichia coli crude extracts to prototype and rationally engineer high-titer mevalonate synthesis. *ACS Synthetic Biology*, 5(12):1578–1588, August 2016.

[72] Ashty S. Karim, Jacob T. Heggestad, Samantha A. Crowe, and Michael C. Jewett. Controlling cell-free metabolism through physiochemical perturbations. *Metabolic Engineering*, 45:86–94, January 2018.

[73] Andrew D. Halleran and Richard M. Murray. Cell-free and in vivo characterization of lux, las, and rpa quorum activation systems in e. coli. *ACS Synthetic Biology*, 7(2):752–755, November 2017.

[74] Keith Pardee, Alexander A. Green, Tom Ferrante, D. Ewen Cameron, Ajay DaleyKeyser, Peng Yin, and James J. Collins. Paper-based synthetic gene networks. *Cell*, 159(4):940–954, November 2014.

[75] Mathilde Koch, Amir Pandi, Olivier Borkowski, Angelo Cardoso Batista, and Jean-Loup Faulon. Custom-made transcriptional biosensors for metabolic engineering. *Current Opinion in Biotechnology*, 59:78–84, October 2019.

[76] Mathilde Koch, Amir Pandi, Baudoin Delépine, and Jean-Loup Faulon. A dataset of small molecules triggering transcriptional and translational cellular responses. *Data in Brief*, 17:1374–1378, April 2018.

[77] Baudoin Delépine, Vincent Libis, Pablo Carbonell, and Jean-Loup Faulon. SensiPath: computer-aided design of sensing-enabling metabolic pathways. *Nucleic Acids Research*, 44(W1):W226–W231, April 2016.

[78] E. J. Corey and W. T. Wipke. Computer-assisted design of complex organic syntheses. *Science*, 166(3902):178–192, October 1969.

[79] R. Pool. Chemistry "grand master" garners a nobel prize: E. j. corey developed a logical methods for synthesizing molecules by working backward from the desired product. *Science*, 250(4980):510–511, October 1990.

[80] Elias James Corey. The logic of chemical synthesis: Multistep synthesis of complex carbogenic molecules(nobel lecture). *Angewandte Chemie International Edition in English*, 30(5):455–465, May 1991.

[81] Yutaka Saito, Misaki Oikawa, Hikaru Nakazawa, Teppei Niide, Tomoshi Kameda, Koji Tsuda, and Mitsuo Umetsu. Machine-learning-guided mutagenesis for directed evolution of fluorescent proteins. *ACS Synthetic Biology*, 7(9):2014–2022, August 2018.

[82] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, October 2017.

[83] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, December 2018.

[84] C. Cheng, N. Carver, and S. Rahimi. Constraint based staff scheduling optimization using single player monte carlo tree search. pages 633–638, 2014.

[85] Frederik Frydenberg, Kasper R. Andersen, Sebastian Risi, and Julian Togelius. Investigating MCTS modifications in general video game playing. In *2015 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, August 2015.

[86] Alan Levinovitz. The mystery of go, the ancient game that computers still can't win, 2014.

[87] Marwin H. S. Segler, Mike Preuss, and Mark P. Waller. Planning chemical syntheses with deep neural networks and symbolic AI. *Nature*, 555(7698):604–610, March 2018.

[88] Wikipedia. Reinforcement learning, 2019.

[89] L. P. Kaelbling, M. L. Littman, and A. W. Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, May 1996.

[90] I. Arel, C. Liu, T. Urbanik, and A.G. Kohls. Reinforcement learning-based multi-agent system for network traffic signal control. *IET Intelligent Transport Systems*, 4(2):128, 2010.

[91] Y. Ganin, T. Kulkarni, I. Babuschkin, S. Eslami, and O. Vinyals. Synthesizing programs for images using reinforced adversarial learning. *arXiv preprint*, 1804.01118, 2018.

[92] Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management - CIKM '18*. ACM Press, 2018.

[93] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. DRN. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*. ACM Press, 2018.

[94] Garychl. Applications of reinforcement learning in real world, 2018.

[95] B. Settles. Active learning literature survey. *Computer Sciences Technical Reports*, 1648, 2009.

[96] Fredrik Olsson. A literature survey of active machine learning in the context of natural language processing. 2009.

[97] Zachary Wu, S. B. Jennifer Kan, Russell D. Lewis, Bruce J. Wittmann, and Frances H. Arnold. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences*, 116(18):8852–8858, April 2019.

[98] Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *Computers and Games*, pages 72–83. Springer Berlin Heidelberg, 2007.

[99] Charles Krauthammer. Be afraid, 1997.

[100] James Somers. The man who would teach machines to think, 2013.

[101] Adrian Cho. 'huge leap forward': Computer that mimics human brain beats professional at game of go, 2016.

[102] Bruno Bouzy and Tristan Cazenave. Computer go: An AI oriented survey. *Artificial Intelligence*, 132(1):39–103, October 2001.

[103] Dana Mackenzie. Why this week's man-versus-machine go match doesn't matter (and what does), 2016.

[104] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016.

[105] Science News Staff. From ai to protein folding: Our breakthrough runners-up, 2016.
[106] Leon Watson. Tech mate? top grandmaster claims chess is riddled with cheats using smartphones, 2015.
[107] Mike Cassidy. Centaur chess shows power of teaming human and machine, 2014.
[108] Daylight. Daylight theory manual, version 4.9, 2017. Accessed September 1, 2017.
[109] Stephen R Heller, Alan McNaught, Igor Pletnev, Stephen Stein, and Dmitrii Tchekhovskoi. InChI, the IUPAC international chemical identifier. *Journal of Cheminformatics*, 7(1), May 2015.
[110] Eric Gifford, Mark Johnson, and Chun che Tsai. A graph-theoretic approach to modeling metabolic pathways. *Journal of Computer-Aided Molecular Design*, 5(4):303–322, August 1991.
[111] Jean-Loup Faulon, Donald P. Visco, and Ramdas S. Pophale. The signature molecular descriptor. 1. using extended valence sequences in QSAR and QSPR studies. *Journal of Chemical Information and Computer Sciences*, 43(3):707–720, May 2003.
[112] Pablo Carbonell, Anne-Gaëlle Planson, Davide Fichera, and Jean-Loup Faulon. A retrosynthetic biology approach to metabolic pathway design for therapeutic production. *BMC Systems Biology*, 5(1):122, aug 2011.
[113] Pablo Carbonell, Pierre Parutto, Claire Baudier, Christophe Junot, and Jean-Loup Faulon. Retropath: Automated pipeline for embedded metabolic circuits. *ACS Synthetic Biology*, 3(8):565–577, October 2013.
[114] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, April 2010.
[115] M. Kotera, Y. Tabei, Y. Yamanishi, T. Tokimatsu, and S. Goto. Supervised de novo reconstruction of metabolic pathways from metabolome-scale compound sets. *Bioinformatics*, 29(13):i135–i144, June 2013.
[116] Thomas Duigou, Melchior du Lac, Pablo Carbonell, and Jean-Loup Faulon. RetroRules: a database of reaction rules for engineering biology. *Nucleic Acids Research*, 47(D1):D1229–D1235, October 2018.
[117] Chunhui Li, Christopher S. Henry, Matthew D. Jankowski, Justin A. Ionita, Vassily Hatzimanikatis, and Linda J. Broadbelt. Computational discovery of biochemical routes to specialty chemicals. *Chemical Engineering Science*, 59(22-23):5051–5060, November 2004.
[118] Mina Oh, Takuji Yamada, Masahiro Hattori, Susumu Goto, and Minoru Kanehisa. Systematic analysis of enzyme-catalyzed reaction patterns and prediction of microbial biodegradation pathways. *Journal of Chemical Information and Modeling*, 47(4):1702–1712, May 2007.
[119] M. J. L. de Groot, R. J. P. van Berlo, W. A. van Winden, P. J. T. Verheijen, M. J. T. Reinders, and D. de Ridder. Metabolite and reaction inference based on enzyme specificities. *Bioinformatics*, 25(22):2975–2982, August 2009.
[120] Joseph L. Durant, Burton A. Leland, Douglas R. Henry, and James G. Nourse. Reoptimization of MDL keys for use in drug discovery. *Journal of Chemical Information and Computer Sciences*, 42(6):1273–1280, November 2002.
[121] Shelley D Copley. Shining a light on enzyme promiscuity. *Current Opinion in Structural Biology*, 47:167–175, December 2017.
[122] Sergio Hernández, Gabriela Ferragut, Isaac Amela, JosepAntoni Perez-Pons, Jaume Piñol, Angel Mozo-Villarias, Juan Cedano, and Enrique Querol. MultitaskProtDB: a database of multitasking proteins. *Nucleic Acids Research*, 42(D1):D517–D520, November 2013.
[123] Jonathan D Tyzack, Nicholas Furnham, Ian Sillitoe, Christine M Orengo, and Janet M Thornton. Understanding enzyme function evolution from a computational perspective. *Current Opinion in Structural Biology*, 47:131–139, December 2017.
[124] R A Jensen. Enzyme recruitment in evolution of new function. *Annual Review of Microbiology*, 30(1):409–425, October 1976.
[125] Takuji Yamada and Peer Bork. Evolution of biomolecular networks — lessons from metabolic and protein interactions. *Nature Reviews Molecular Cell Biology*, 10(11):791–803, November 2009.
[126] Pablo Carbonell, Guillaume Lecointre, and Jean-Loup Faulon. Origins of specificity and promiscuity in metabolic networks. *Journal of Biological Chemistry*, 286(51):43994–44004, November 2011.
[127] Lianet Noda-Garcia, Wolfram Liebermeister, and Dan S. Tawfik. Metabolite–enzyme coevolution: From single enzymes to metabolic pathways and networks. *Annual Review of Biochemistry*, 87(1):187–216, June 2018.
[128] Sandeep Chakraborty, Renu Minda, Lipika Salaye, Abhaya M. Dandekar, Swapan K. Bhattacharjee, and Basuthkar J. Rao. Promiscuity-based enzyme selection for rational directed evolution experiments.

In *Methods in Molecular Biology*, pages 205–216. Humana Press, 2013.

[129] Masahito Hosokawa, Yuri Hoshino, Yohei Nishikawa, Tomotada Hirose, Dong Hyun Yoon, Tetsushi Mori, Tetsushi Sekiguchi, Shuichi Shoji, and Haruko Takeyama. Droplet-based microfluidics for high-throughput screening of a metagenomic library for isolation of microbial enzymes. *Biosensors and Bioelectronics*, 67:379–385, May 2015.

[130] H. Nam, N. E. Lewis, J. A. Lerman, D.-H. Lee, R. L. Chang, D. Kim, and B. O. Palsson. Network context and selection in the evolution to enzyme specificity. *Science*, 337(6098):1101–1104, August 2012.

[131] Gabriela I. Guzmán, José Utrilla, Sergey Nurk, Elizabeth Brunk, Jonathan M. Monk, Ali Ebrahim, Bernhard O. Palsson, and Adam M. Feist. Model-driven discovery of underground metabolic functions inEscherichia coli. *Proceedings of the National Academy of Sciences*, 112(3):929–934, January 2015.

[132] Sara A. Amin, Elizabeth Chavez, Vladimir Porokhin, Nikhil U. Nair, and Soha Hassoun. Towards creating an extended metabolic model (EMM) for e. coli using enzyme promiscuity prediction and metabolomics data. *Microbial Cell Factories*, 18(1), June 2019.

[133] Lisa Jeske, Sandra Placzek, Ida Schomburg, Antje Chang, and Dietmar Schomburg. BRENDA in 2019: a european ELIXIR core data resource. *Nucleic Acids Research*, 47(D1):D542–D549, November 2019.

[134] Abhinav Nath and William M. Atkins. A quantitative index of substrate promiscuity†. *Biochemistry*, 47(1):157–166, January 2008.

[135] International Union of Biochemistry and Molecular Biology Nomenclature Committee and Edwin Clifford Webb. Enzyme nomenclature 1992: recommendations of the nomenclature committee of the international union of biochemistry and molecular biology on the nomenclature and classification of enzymes. In *International Union of Biochemistry and Molecular Biology*, page 888. Academic Press, 1992.

[136] Noushin Hadadi, Jasmin Hafner, Adrian Shajkofci, Aikaterini Zisaki, and Vassily Hatzimanikatis. ATLAS of biochemistry: A repository of all possible biochemical reactions for synthetic biology and metabolic engineering studies. *ACS Synthetic Biology*, 5(10):1155–1166, July 2016.

[137] Christopher S. Henry, Linda J. Broadbelt, and Vassily Hatzimanikatis. Discovery and analysis of novel metabolic pathways for the biosynthesis of industrial chemicals: 3-hydroxypropanoate. *Biotechnology and Bioengineering*, pages n/a–n/a, 2010.

[138] Baudoin Delépine, Thomas Duigou, Pablo Carbonell, and Jean-Loup Faulon. RetroPath2.0: A retrosynthesis workflow for metabolic engineers. *Metabolic Engineering*, 45:158–170, January 2018.

[139] Yiannis N Kaznessis. Models for synthetic biology. *BMC Systems Biology*, 1(1), November 2007.

[140] Jens Nielsen. Systems biology of metabolism. *Annual Review of Biochemistry*, 86(1):245–275, June 2017.

[141] L. Michaelis and M. L. Menten. Die kinetik der invertinwirkung. *Biochem. Z*, 49(333-369):352, 1913.

[142] Sara Berthoumieux, Matteo Brilli, Daniel Kahn, Hidde de Jong, and Eugenio Cinquemani. On the identifiability of metabolic network models. *Journal of Mathematical Biology*, 67(6-7):1795–1832, November 2012.

[143] Yves Berset, Davide Merulla, Aurélie Joublin, Vassily Hatzimanikatis, and Jan R. van der Meer. Mechanistic modeling of genetic circuits for ArsR arsenic regulation. *ACS Synthetic Biology*, 6(5):862–874, February 2017.

[144] Jonathan R. Karr, Jayodita C. Sanghvi, Derek N. Macklin, Miriam V. Gutschow, Jared M. Jacobs, Benjamin Bolival, Nacyra Assad-Garcia, John I. Glass, and Markus W. Covert. A whole-cell computational model predicts phenotype from genotype. *Cell*, 150(2):389–401, July 2012.

[145] Carlotta Martelli, Andrea De Martino, Enzo Marinari, Matteo Marsili, and Isaac Pérez Castillo. Identifying essential genes inEscherichia colifrom a metabolic optimization principle. *Proceedings of the National Academy of Sciences*, 106(8):2607–2611, February 2009.

[146] Nathan E. Lewis, Harish Nagarajan, and Bernhard O. Palsson. Constraining the metabolic genotype–phenotype relationship using a phylogeny of in silico methods. *Nature Reviews Microbiology*, 10(4):291–305, February 2012.

[147] Ana Bulović, Stephan Fischer, Marc Dinh, Felipe Golib, Wolfram Liebermeister, Christian Poirier, Laurent Tournier, Edda Klipp, Vincent Fromion, and Anne Goelzer. Automated generation of bacterial resource allocation models. *Metabolic Engineering*, 55:12–22, September 2019.

[148] Anne Goelzer, Jan Muntel, Victor Chubukov, Matthieu Jules, Eric Prestel, Rolf Nölker, Mahendra Mariadassou, Stéphane Aymerich, Michael Hecker, Philippe Noirot, Dörte Becher, and Vincent Fromion. Quantitative prediction of genome-wide resource allocation in bacteria. *Metabolic Engineering*, 32:232–

243, November 2015.
[149] Christopher S. Henry, Linda J. Broadbelt, and Vassily Hatzimanikatis. Thermodynamics-based metabolic flux analysis. *Biophysical Journal*, 92(5):1792–1805, March 2007.
[150] R. L. Chang, K. Andrews, D. Kim, Z. Li, A. Godzik, and B. O. Palsson. Structural systems biology evaluation of metabolic thermotolerance in escherichia coli. *Science*, 340(6137):1220–1223, June 2013.
[151] Ove Øyås and Jörg Stelling. Genome-scale metabolic networks in time and space. *Current Opinion in Systems Biology*, 8:51–58, April 2018.
[152] Renana Sabi and Tamir Tuller. Modelling and measuring intracellular competition for finite resources during gene expression. *Journal of The Royal Society Interface*, 16(154):20180887, May 2019.
[153] Yili Qian, Hsin-Ho Huang, José I. Jiménez, and Domitilla Del Vecchio. Resource competition shapes the response of genetic circuits. *ACS Synthetic Biology*, 6(7):1263–1272, April 2017.
[154] Andras Gyorgy, José I. Jiménez, John Yazbek, Hsin-Ho Ho Huang, Hattie Chung, Ron Weiss, and Domitilla Del Vecchio. Isocost Lines Describe the Cellular Economy of Genetic Circuits. *Biophysical Journal*, 109(3):639–646, aug 2015.
[155] J N Weiss. The hill equation revisited: uses and misuses. *The FASEB Journal*, 11(11):835–841, September 1997.
[156] Evangelos-Marios Nikolados, Andrea Y. Weiße, Francesca Ceroni, and Diego A. Oyarzún. Growth defects and loss-of-function in synthetic gene circuits. *ACS Synthetic Biology*, 8(6):1231–1240, May 2019.
[157] Pencho Yordanov and Jörg Stelling. Steady-state differential dose response in biological systems. *Biophysical Journal*, 114(3):723–736, February 2018.
[158] Susanna Zucca, Lorenzo Pasotti, Giuliano Mazzini, Maria Gabriella Cusella De Angelis, and Paolo Magni. Characterization of an inducible promoter in different DNA copy number conditions. *BMC Bioinformatics*, 13(Suppl 4):S11, 2012.
[159] Lorenzo Pasotti, Massimo Bellato, Davide De Marchi, and Paolo Magni. Mechanistic models of inducible synthetic circuits for joint description of DNA copy number, regulatory protein level, and cell load. *Processes*, 7(3):119, February 2019.
[160] Timo Lubitz and Wolfram Liebermeister. Parameter balancing: consistent parameter sets for kinetic metabolic models. *Bioinformatics*, February 2019.
[161] Claudia Schillings, Mikael Sunnåker, Jörg Stelling, and Christoph Schwab. Efficient characterization of parametric uncertainty of complex (bio)chemical networks. *PLOS Computational Biology*, 11(8):e1004457, August 2015.
[162] Claude Lormeau, Mikołaj Rybiński, and Jörg Stelling. Multi-objective design of synthetic biological circuits. *IFAC-PapersOnLine*, 50(1):9871–9876, July 2017.
[163] Ido Golding, Johan Paulsson, Scott M. Zawilski, and Edward C. Cox. Real-time kinetics of gene activity in individual bacteria. *Cell*, 123(6):1025–1036, December 2005.
[164] Lekshmi Dharmarajan, Hans-Michael Kaltenbach, Fabian Rudolf, and Joerg Stelling. A simple and flexible computational framework for inferring sources of heterogeneity from single-cell dynamics. *Cell Systems*, 8(1):15–26.e11, January 2019.
[165] Artémis Llamosi, Andres M. Gonzalez-Vargas, Cristian Versari, Eugenio Cinquemani, Giancarlo Ferrari-Trecate, Pascal Hersen, and Gregory Batt. What population reveals about individual cell identity: Single-cell parameter estimation of models of gene expression in yeast. *PLOS Computational Biology*, 12(2):e1004706, February 2016.
[166] Javier Macía, Francesc Posas, and Ricard V. Solé. Distributed computation: the new wave of synthetic biology devices. *Trends in Biotechnology*, 30(6):342–349, June 2012.
[167] Eric J. Leaman, Brian Q. Geuther, and Bahareh Behkam. Quantitative investigation of the role of intra-/intercellular dynamics in bacterial quorum sensing. *ACS Synthetic Biology*, 7(4):1030–1042, March 2018.
[168] Thomas E. Gorochowski. Agent-based modelling in synthetic biology. *Essays In Biochemistry*, 60(4):325–336, November 2016.
[169] Pablo Carbonell, Mathilde Koch, Thomas Duigou, and Jean-Loup Faulon. Enzyme discovery: Enzyme selection and pathway design. In *Methods in Enzymology*, pages 3–27. Elsevier, 2018.
[170] Mathilde Koch, Thomas Duigou, Pablo Carbonell, and Jean-Loup Faulon. Molecular structures enumeration and virtual screening in the chemical space with RetroPath2.0. *Journal of Cheminformatics*, 9(1), December 2017.

[171] Heykel Trabelsi, Mathilde Koch, and Jean-Loup Faulon. Building a minimal and generalizable model of transcription factor–based biosensors: Showcasing flavonoids. *Biotechnology and Bioengineering*, 115(9):2292–2304, May 2018.

[172] Mathilde Koch, Jean-Loup Faulon, and Olivier Borkowski. Models for cell-free synthetic biology: Make prototyping easier, better, and faster. *Frontiers in Bioengineering and Biotechnology*, 6, November 2018.

[173] Peter L. Voyvodic, Amir Pandi, Mathilde Koch, Ismael Conejero, Emmanuel Valjent, Philippe Courtet, Eric Renard, Jean-Loup Faulon, and Jerome Bonnet. Plug-and-play metabolic transducers expand the chemical detection space of cell-free biosensors. *Nature Communications*, 10(1), April 2019.

[174] Amir Pandi, Mathilde Koch, Peter L. Voyvodic, Paul Soudier, Jerome Bonnet, Manish Kushwaha, and Jean-Loup Faulon. Metabolic perceptrons for neural computing in biological systems. *Nature Communications*, 10(1), August 2019.

[175] Pablo Carbonell, Jerry Wong, Neil Swainston, Eriko Takano, Nicholas J Turner, Nigel S Scrutton, Douglas B Kell, Rainer Breitling, and Jean-Loup Faulon. Selenzyme: enzyme selection tool for pathway design. *Bioinformatics*, 34(12):2153–2154, feb 2018.

[176] Pablo Carbonell, Andrew Currin, Adrian J. Jervis, Nicholas J. W. Rattray, Neil Swainston, Cunyu Yan, Eriko Takano, and Rainer Breitling. Bioinformatics for the synthetic biology of natural products: integrating across the design–build–test cycle. *Natural Product Reports*, 33(8):925–932, 2016.

[177] Tilmann Weber, Kai Blin, Srikanth Duddela, Daniel Krug, Hyun Uk Kim, Robert Bruccoleri, Sang Yup Lee, Michael A. Fischbach, Rolf Müller, Wolfgang Wohlleben, Rainer Breitling, Eriko Takano, and Marnix H. Medema. antiSMASH 3.0—a comprehensive resource for the genome mining of biosynthetic gene clusters. *Nucleic Acids Research*, 43(W1):W237–W243, may 2015.

[178] Syed Asad Rahman, Sergio Martinez Cuesta, Nicholas Furnham, Gemma L Holliday, and Janet M Thornton. EC-BLAST: a tool to automatically search and compare enzyme reactions. *Nature Methods*, 11(2):171–174, jan 2014.

[179] Y. Yamanishi, M. Hattori, M. Kotera, S. Goto, and M. Kanehisa. E-zyme: predicting potential EC numbers from the chemical transformation pattern of substrate-product pairs. *Bioinformatics*, 25(12):i179–i186, may 2009.

[180] Yu Li, Sheng Wang, Ramzan Umarov, Bingqing Xie, Ming Fan, Lihua Li, and Xin Gao. DEEPre: sequence-based enzyme EC number prediction by deep learning. *Bioinformatics*, 34(5):760–769, oct 2018.

[181] Joseph Mellor, Ioana Grigoras, Pablo Carbonell, and Jean-Loup Faulon. Semisupervised gaussian process for automated enzyme search. *ACS Synthetic Biology*, 5(6):518–528, mar 2016.

[182] Noushin Hadadi and Vassily Hatzimanikatis. Design of computational retrobiosynthesis tools for the design of de novo synthetic pathways. *Current Opinion in Chemical Biology*, 28:99–104, oct 2015.

[183] Sang Yup Lee and Hyun Uk Kim. Systems strategies for developing industrial microbial strains. *Nature biotechnology*, 33(10):1061–72, oct 2015.

[184] Marnix H. Medema, Renske van Raaphorst, Eriko Takano, and Rainer Breitling. Computational tools for the synthetic design of biochemical pathways. *Nature Reviews Microbiology*, 10(3):191–202, January 2012.

[185] Y. Moriya, D. Shigemizu, M. Hattori, T. Tokimatsu, M. Kotera, S. Goto, and M. Kanehisa. PathPred: an enzyme-catalyzed metabolic pathway prediction server. *Nucleic Acids Research*, 38(Web Server):W138–W143, apr 2010.

[186] Guillermo Rodrigo, Javier Carrera, Kristala Jones Prather, and Alfonso Jaramillo. DESHARKY: Automatic design of metabolic pathways for optimal cell growth. *Bioinformatics*, 24(21):2554–2556, nov 2008.

[187] Miguel A Campodonico, Barbara A Andrews, Juan A Asenjo, Bernhard Ø Palsson, and Adam M Feist. Generation of an atlas for commodity chemical production in Escherichia coli and a novel pathway prediction algorithm, GEM-Path. *Metabolic Engineering*, 25:140–158, 2014.

[188] Pablo Carbonell, Pierre Parutto, Joan Herisson, Shashi Bhushan Pandit, and Jean-Loup Loup Faulon. XTMS: Pathway design in an eXTended metabolic space. *Nucleic Acids Research*, 42(Web Server):W389–W394, jul 2014.

[189] Noushin Hadadi, Jasmin Hafner, Keng Cher Soh, and Vassily Hatzimanikatis. Reconstruction of biological pathways and metabolic networks from in silico labeled metabolites. *Biotechnology Journal*, 12(1):1600464, jan 2017.

[190] Mengjin Liu, Bruno Bienfait, Oliver Sacher, Johann Gasteiger, Roland J. Siezen, Arjen Nauta, and Jan M.W. W. Geurts. Combining chemoinformatics with bioinformatics: In silico prediction of bacterial flavor-forming pathways by a chemical systems biology approach "Reverse Pathway Engineering". *PLoS ONE*, 9(1):e84769, jan 2014.

[191] Harry Yim, Robert Haselbeck, Wei Niu, Catherine Pujol-Baxley, Anthony Burgard, Jeff Boldt, Julia Khandurina, John D Trawick, Robin E Osterhout, Rosary Stephen, Jazell Estadilla, Sy Teisan, H Brett Schreyer, Stefan Andrae, Tae Hoon Yang, Sang Yup Lee, Mark J Burk, and Stephen Van Dien. Metabolic engineering of Escherichia coli for direct production of 1,4-butanediol. *Nature chemical biology*, 7(7):445–52, may 2011.

[192] Pablo Carbonell and Jean-Yves Trosset. Computational protein design methods for synthetic biology. In *Methods in Molecular Biology*, pages 3–21. Springer New York, nov 2015.

[193] Noel M O'Boyle, Michael Banck, Craig A James, Chris Morley, Tim Vandermeersch, and Geoffrey R Hutchison. Open babel: An open chemical toolbox. *Journal of Cheminformatics*, 3(10):33, 2011.

[194] Greg Landrum. Rdkit documentation, 2017. Accessed September 1, 2017.

[195] Syed Asad Rahman, Gilliean Torrance, Lorenzo Baldacci, Sergio Martínez Cuesta, Franz Fenninger, Nimish Gopal, Saket Choudhary, John W. May, Gemma L. Holliday, Christoph Steinbeck, and Janet M. Thornton. Reaction decoder tool (RDT): extracting features from chemical reactions. *Bioinformatics*, 32(13):2065–2066, feb 2016.

[196] Gerald M. Maggiora and Veerabahu Shanmugasundaram. Molecular similarity measures. In *Methods in Molecular Biology*, pages 39–100. Humana Press, aug 2011.

[197] Alexander Fillbrunn, Christian Dietz, Julianus Pfeuffer, René Rahn, Gregory A. Landrum, and Michael R. Berthold. KNIME for reproducible cross-domain analysis of life science data. *Journal of Biotechnology*, 261:149–156, nov 2017.

[198] Tomer Altman, Michael Travers, Anamika Kothari, Ron Caspi, and Peter D Karp. A systematic comparison of the MetaCyc and KEGG pathway databases. *BMC Bioinformatics*, 14(1):112, 2013.

[199] Matthew D. Jankowski, Christopher S. Henry, Linda J. Broadbelt, and Vassily Hatzimanikatis. Group contribution method for thermodynamic analysis of complex metabolic networks. *Biophysical Journal*, 95(3):1487–1499, aug 2008.

[200] R. Caspi, T. Altman, K. Dreher, C. A. Fulcher, P. Subhraveti, I. M. Keseler, A. Kothari, M. Krummenacker, M. Latendresse, L. A. Mueller, Q. Ong, S. Paley, A. Pujar, A. G. Shearer, M. Travers, D. Weerasinghe, P. Zhang, and P. D. Karp. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Research*, 40(D1):D742–D753, nov 2012.

[201] Adam M Feist, Christopher S Henry, Jennifer L Reed, Markus Krummenacker, Andrew R Joyce, Peter D Karp, Linda J Broadbelt, Vassily Hatzimanikatis, and Bernhard Ø Palsson. A genome-scale metabolic reconstruction for escherichia coli k-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Molecular Systems Biology*, 3(1):121, jun 2007.

[202] Sébastien Moretti, Olivier Martin, T. Van Du Tran, Alan Bridge, Anne Morgat, and Marco Pagni. MetaNetX/MNXref – reconciliation of metabolites and biochemical reactions to bring together genome-scale metabolic networks. *Nucleic Acids Research*, 44(D1):D523–D526, nov 2015.

[203] Neil Swainston, Riza Batista-Navarro, Pablo Carbonell, Paul D. Dobson, Mark Dunstan, Adrian J. Jervis, Maria Vinaixa, Alan R. Williams, Sophia Ananiadou, Jean-Loup Faulon, Pedro Mendes, Douglas B. Kell, Nigel S. Scrutton, and Rainer Breitling. biochem4j: Integrated and extensible biochemical knowledge through graph databases. *PLOS ONE*, 12(7):e0179130, jul 2017.

[204] W. Li and A. Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, may 2006.

[205] Jean-Francois Taly, Cedrik Magis, Giovanni Bussotti, Jia-Ming Chang, Paolo Di Tommaso, Ionas Erb, Jose Espinosa-Carrasco, Carsten Kemena, and Cedric Notredame. Using the t-coffee package to build multiple sequence alignments of protein, RNA, DNA sequences and 3d structures. *Nature Protocols*, 6(11):1669–1682, oct 2011.

[206] P. Rice, I. Longden, and A. Bleasby. Emboss: The european molecular biology open software suite. *Trends in Genetics*, 16(6):276–277, 2000.

[207] S. Kawashima and M. Kanehisa. AAindex: Amino acid index database. *Nucleic Acids Research*, 28(1):374–374, jan 2000.

[208] Guy Yachdav, Sebastian Wilzbach, Benedikt Rauscher, Robert Sheridan, Ian Sillitoe, James Procter, Suzanna E. Lewis, Burkhard Rost, and Tatyana Goldberg. MSAViewer: interactive JavaScript

visualization of multiple sequence alignments. *Bioinformatics*, 32(22):3501–353, jul 2016.

[209] P. Carbonell, A. Currin, M. Dunstan, D. Fellows, A. Jervis, N. J. W. Rattray, C. J. Robinson, N. Swainston, M. Vinaixa, A. Williams, C. Yan, P. Barran, R. Breitling, G. G.-Q. Chen, J.-L. Faulon, C. Goble, R. Goodacre, D. B. Kell, R. L. Feuvre, J. Micklefield, N. S. Scrutton, P. Shapira, E. Takano, and N. J. Turner. SYNBIOCHEM-a SynBio foundry for the biosynthesis and sustainable production of fine and speciality chemicals. *Biochemical Society Transactions*, 44(3):675–677, jun 2016.

[210] Pablo Carbonell, Davide Fichera, Shashi B Pandit, and Jean-Loup Faulon. Enumerating metabolic pathways for the production of heterologous target chemicals in chassis organisms. *BMC Systems Biology*, 6(1):10, feb 2012.

[211] Anne Gaëlle Planson, Pablo Carbonell, Elodie Paillard, Nicolas Pollet, and Jean Loup Faulon. Compound toxicity screening and structure-activity relationship modeling in Escherichia coli. *Biotechnology and Bioengineering*, 109(3):846–850, 2012.

[212] Ayoun Cho, Hongseok Yun, Jin Park, Sang Lee, and Sunwon Park. Prediction of novel synthetic pathways for the production of desired chemicals. *BMC Systems Biology*, 4(1):35, 2010.

[213] Shardul Paricharak, Tom Klenka, Martin Augustin, Umesh A Patel, and Andreas Bender. Are phylogenetic trees suitable for chemogenomics analyses of bioactivity data sets: the importance of shared active compounds and choosing a suitable data embedding method, as exemplified on kinases. *Journal of Cheminformatics*, 5(1):49, 2013.

[214] James G Jeffryes, Ricardo L Colastani, Mona Elbadawi-Sidhu, Tobias Kind, Thomas D Niehaus, Linda J Broadbelt, Andrew D Hanson, Oliver Fiehn, Keith E J Tyo, and Christopher S Henry. MINEs: open access databases of computationally predicted enzyme promiscuity products for untargeted metabolomics. *Journal of Cheminformatics*, 7(1), August 2015.

[215] Stefan Schuster, David A. Fell, and Thomas Dandekar. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature Biotechnology*, 18(3):326–332, mar 2000.

[216] Jeffrey D. Orth, Tom M. Conrad, Jessica Na, Joshua A. Lerman, Hojung Nam, Adam M. Feist, and Bernhard Ø Palsson. A comprehensive genome-scale reconstruction of Escherichia coli metabolism–2011. *Molecular systems biology*, 7:535, oct 2011.

[217] CASREACT. Chemicals abstracts service, 2017. Accessed June 28, 2017.

[218] Wendy A. Warr. A short review of chemical reaction database systems, computer-aided synthesis design, reaction prediction and synthetic feasibility. *Molecular Informatics*, 33(6-7):469–476, June 2014.

[219] Sara Szymkuć, Ewa P. Gajewska, Tomasz Klucznik, Karol Molga, Piotr Dittwald, Michał Startek, Michał Bajczyk, and Bartosz A. Grzybowski. Computer-assisted synthetic planning: The end of the beginning. *Angewandte Chemie International Edition*, 55(20):5904–5937, April 2016.

[220] Gisbert Schneider. Future de novo drug design. *Molecular Informatics*, 33(6-7):397–402, May 2014.

[221] Ivar Ugi, Johannes Bauer, Klemens Bley, Alf Dengler, Andreas Dietz, Eric Fontain, Bernhard Gruber, Rainer Herges, Michael Knauer, Klaus Reitsam, and Natalie Stein. Computer-assisted solution of chemical problems—the historical development and the present state of the art of a new discipline of chemistry. *Angewandte Chemie International Edition in English*, 32(2):164–189, February 1993.

[222] Jean-Loup Faulon and Andreas Bender. *Reaction network generation*. Handbook of Chemoinformatics Algorithms, April 2010.

[223] Michael R. Berthold, Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Christoph Sieb, Kilian Thiel, and Bernd Wiswedel. KNIME: The Konstanz Information Miner. pages 319–326. Springer, Berlin, Heidelberg, 2008.

[224] Tiago Rodrigues, Daniel Reker, Martin Welin, Michael Caldera, Cyrill Brunner, Gisela Gabernet, Petra Schneider, Björn Walse, and Gisbert Schneider. De novo fragment design for drug discovery and chemical biology. *Angewandte Chemie International Edition*, 54(50):15079–15083, October 2015.

[225] Tamás Fehér, Anne-Gaëlle Planson, Pablo Carbonell, Alfred Fernández-Castané, Ioana Grigoras, Ekaterina Dariy, Alain Perret, and Jean-Loup Faulon. Validation of RetroPath, a computer-aided design tool for metabolic pathway engineering. *Biotechnology Journal*, 9(11):1446–1457, October 2014.

[226] Hannes L. Röst, Uwe Schmitt, Ruedi Aebersold, and Lars Malmström. pyOpenMS: A python-based interface to the OpenMS mass-spectrometry algorithm library. *Proteomics*, 14(1):74–77, January 2014.

[227] Dheivya Thiagarajan and Dinesh P. Mehta. Faster algorithms for isomer network generation. *Journal of Chemical Information and Modeling*, 56(12):2310–2319, November 2016.

[228] Julio E Peironcely, Miguel Rojas-Chertó, Davide Fichera, Theo Reijmers, Leon Coulier, Jean-Loup Faulon, and Thomas Hankemeier. OMG: Open molecule generator. *Journal of Cheminformatics*,

4(1):21, 2012.
[229] Mohammad Mahdi Jaghoori, Sung-Shik T.Q. Jongmans, Frank de Boer, Julio Peironcely, Jean-Loup Faulon, Theo Reijmers, and Thomas Hankemeier. PMG: Multi-core metabolite identification. *Electronic Notes in Theoretical Computer Science*, 299:53–60, December 2013.
[230] Brendan D McKay. Isomorph-free exhaustive generation. *Journal of Algorithms*, 26(2):306–324, February 1998.
[231] José-Manuel Gally, Stéphane Bourg, Quoc-Tuan Do, Samia Aci-Sèche, and Pascal Bonnet. VSPrep: A general KNIME workflow for the preparation of molecules for virtual screening. *Molecular Informatics*, 36(10):1–11, June 2017.
[232] Lars Ruddigkeit, Ruud van Deursen, Lorenz C. Blum, and Jean-Louis Reymond. Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17. *Journal of Chemical Information and Modeling*, 52(11):2864–2875, November 2012.
[233] Univ Bayreuth. Isomers of alkanes, 2017. Accessed June 28, 2017.
[234] Peter Willett. Similarity-based virtual screening using 2d fingerprints. *Drug Discovery Today*, 11(23-24):1046–1053, December 2006.
[235] W. Michael Brown, Shawn Martin, Mark D. Rintoul, and Jean-Loup Faulon. Designing novel polymers with targeted properties using the signature molecular descriptor. *Journal of Chemical Information and Modeling*, 46(2):826–835, March 2006.
[236] Muralisrinivasan Natamai Subramanian. Polymer properties. *Polymer Blends and Composites*, 2017.
[237] James C. Gerdeen and Ronald A.L. Rorrer. Engineering design with polymers and composites. *CRC Press*, 30, 2011.
[238] Carla J Churchwell, Mark D Rintoul, Shawn Martin, Donald P Visco, Archana Kotu, Richard S Larson, Laurel O Sillerud, David C Brown, and Jean-Loup Faulon. The signature molecular descriptor. 3 inverse quantitative structure-activity relationships of icam-1 inhibitory peptides. *Journal of Molecular Graphics and Modelling*, 22(4):263–273, March 2004.
[239] Shawn Martin. Lattice enumeration for inverse molecular design using the signature descriptor. *Journal of Chemical Information and Modeling*, 52(7):1787–1797, June 2012.
[240] Piotr Setny and Joanna Trylska. Search for novel aminoglycosides by combining fragment-based virtual screening and 3d-QSAR scoring. *Journal of Chemical Information and Modeling*, 49(2):390–400, January 2009.
[241] Kentaro Kawai, Naoya Nagata, and Yoshimasa Takahashi. De novo design of drug-like molecules by a fragment-based molecular evolutionary approach. *Journal of Chemical Information and Modeling*, 54(1):49–56, January 2014.
[242] David S. Wishart, Timothy Jewison, An Chi Guo, Michael Wilson, Craig Knox, Yifeng Liu, Yannick Djoumbou, Rupasri Mandal, Farid Aziat, Edison Dong, Souhaila Bouatra, Igor Sinelnikov, David Arndt, Jianguo Xia, Philip Liu, Faizath Yallou, Trent Bjorndahl, Rolando Perez-Pineiro, Roman Eisner, Felicity Allen, Vanessa Neveu, Russ Greiner, and Augustin Scalbert. HMDB 3.0—the human metabolome database in 2013. *Nucleic Acids Research*, 41(D1):D801–D807, November 2012.
[243] MTBLS229. Metabolights, 2017. Accessed June 28, 2017.
[244] Patrick Kiefer, Uwe Schmitt, Jonas E. N. Müller, Johannes Hartl, Fabian Meyer, Florian Ryffel, and Julia A. Vorholt. DynaMet: A fully automated pipeline for dynamic LC–MS data. *Analytical Chemistry*, 87(19):9679–9686, September 2015.
[245] Donald P. Visco, Ramdas S. Pophale, Mark D. Rintoul, and Jean-Loup Faulon. Developing a methodology for an inverse quantitative structure-activity relationship using the signature molecular descriptor. *Journal of Molecular Graphics and Modelling*, 20(6):429–438, June 2002.
[246] Ruud van Deursen and Jean-Louis Reymond. Chemical space travel. *ChemMedChem*, 2(5):636–640, May 2007.
[247] Melvin J. Yu. Natural product-like virtual libraries: Recursive atom-based enumeration. *Journal of Chemical Information and Modeling*, 51(3):541–557, March 2011.
[248] David Hoksza, Petr Škoda, Milan Voršilák, and Daniel Svozil. Molpher: a software framework for systematic chemical space exploration. *Journal of Cheminformatics*, 6(1):7, March 2014.
[249] Aaron M. Virshup, Julia Contreras-García, Peter Wipf, Weitao Yang, and David N. Beratan. Stochastic voyages into uncharted chemical space produce a representative library of all possible drug-like compounds. *Journal of the American Chemical Society*, 135(19):7296–7303, May 2013.
[250] KEGG. Kegg pathway, neomycin, kanamycin and gentamicin biosynthesis, 2017. Accessed June 28, 2017.

[251] M Walzer, T Sachsenberg, r F Aichele, M Rurik, J Veit, B Isabell, P Pedrioli, J Pfeuffer, X Liang, K Reinert, and O Kohlbacher. Openms tutorial, 2017. Accessed June 28, 2017.

[252] Akhil Kumar, Lin Wang, Chiam Yu Ng, and Costas D. Maranas. Pathway design using de novo steps through uncharted biochemical spaces. *Nature Communications*, 9(1), January 2018.

[253] Milenko Tokic, Noushin Hadadi, Meric Ataman, Dário Neves, Birgitta E. Ebert, Lars M. Blank, Ljubisa Miskovic, and Vassily Hatzimanikatis. Discovery and evaluation of biosynthetic pathways for the production of five methyl ethyl ketone precursors. *ACS Synthetic Biology*, 7(8):1858–1873, July 2018.

[254] Arturo Casini, Fang-Yuan Chang, Raissa Eluere, Andrew M. King, Eric M. Young, Quentin M. Dudley, Ashty Karim, Katelin Pratt, Cassandra Bristol, Anthony Forget, Amar Ghodasara, Robert Warden-Rothman, Rui Gan, Alexander Cristofaro, Amin Espah Borujeni, Min-Hyung Ryu, Jian Li, Yong-Chan Kwon, He Wang, Evangelos Tatsis, Carlos Rodriguez-Lopez, Sarah O'Connor, Marnix H. Medema, Michael A. Fischbach, Michael C. Jewett, Christopher Voigt, and D. Benjamin Gordon. A pressure test to make 10 molecules in 90 days: External evaluation of methods to engineer biology. *Journal of the American Chemical Society*, 140(12):4302–4316, February 2018.

[255] Sang Yup Lee, Hyun Uk Kim, Tong Un Chae, Jae Sung Cho, Je Woong Kim, Jae Ho Shin, Dong In Kim, Yoo-Sung Ko, Woo Dae Jang, and Yu-Sin Jang. A comprehensive metabolic map for production of bio-based chemicals. *Nature Catalysis*, 2(1):18–33, January 2019.

[256] Noushin Hadadi, Homa MohammadiPeyhani, Ljubisa Miskovic, Marianne Seijo, and Vassily Hatzimanikatis. Enzyme annotation for orphan and novel reactions using knowledge of substrate reactive sites. *Proceedings of the National Academy of Sciences*, 116(15):7298–7307, March 2019.

[257] Connor W. Coley, Luke Rogers, William H. Green, and Klavs F. Jensen. Computer-assisted retrosynthesis based on molecular similarity. *ACS Central Science*, 3(12):1237–1245, November 2017.

[258] James D. Winkler, Andrea L. Halweg-Edwards, and Ryan T. Gill. The LASER database: Formalizing design rules for metabolic engineering. *Metabolic Engineering Communications*, 2:30–38, December 2015.

[259] James D. Winkler, Andrea L. Halweg-Edwards, and Ryan T. Gill. Quantifying complexity in metabolic engineering using the LASER database. *Metabolic Engineering Communications*, 3:227–233, December 2016.

[260] Michael G. Bramucci, Carol M. McCutchen, Vasantha Nagarajan, and Stuart M. Thomas. Microbial production of terephthalic acid and isophthalic acid, 2001.

[261] Kate Thodey, Stephanie Galanie, and Christina D Smolke. A microbial biomanufacturing platform for natural and semisynthetic opioids. *Nature Chemical Biology*, 10(10):837–844, August 2014.

[262] Vincent Libis, Baudoin Delépine, and Jean-Loup Faulon. Expanding biosensing abilities through computer-aided design of metabolic pathways. *ACS Synthetic Biology*, 5(10):1076–1085, mar 2016.

[263] Geng-Min Lin, Robert Warden-Rothman, and Christopher A. Voigt. Retrosynthetic design of metabolic pathways to chemicals not found in nature. *Current Opinion in Systems Biology*, 14:82–107, April 2019.

[264] John S. Schreck, Connor W. Coley, and Kyle J. M. Bishop. Learning retrosynthetic planning through simulated experience. *ACS Central Science*, 5(6):970–981, May 2019.

[265] Stephen Heller, Alan McNaught, Stephen Stein, Dmitrii Tchekhovskoi, and Igor Pletnev. InChI - the worldwide chemical structure identifier standard. *Journal of Cheminformatics*, 5(1), January 2013.

[266] Aric Hagberg, Pieter Swart, and Daniel Chult. Exploring network structure, dynamics, and function using networkx. 01 2008.

[267] Dávid Bajusz, Anita Rácz, and Károly Héberger. Why is tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, 7(1), May 2015.

[268] The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic Acids Research*, 46(5):2699–2699, February 2018.

[269] Anne Morgat, Thierry Lombardot, Kristian B. Axelsen, Lucila Aimo, Anne Niknejad, Nevila Hyka-Nouspikel, Elisabeth Coudert, Monica Pozzato, Marco Pagni, Sébastien Moretti, Steven Rosanoff, Joseph Onwubiko, Lydie Bougueleret, Ioannis Xenarios, Nicole Redaschi, and Alan Bridge. Updates in rhea – an expert curated resource of biochemical reactions. *Nucleic Acids Research*, 45(D1):D415–D418, October 2016.

[270] Ron Caspi, Richard Billington, Carol A Fulcher, Ingrid M Keseler, Anamika Kothari, Markus Krummenacker, Mario Latendresse, Peter E Midford, Quang Ong, Wai Kit Ong, Suzanne Paley, Pallavi Subhraveti, and Peter D Karp. The MetaCyc database of metabolic pathways and enzymes. *Nucleic Acids Research*, 46(D1):D633–D639, October 2017.

[271] Antonio Fabregat, Steven Jupe, Lisa Matthews, Konstantinos Sidiropoulos, Marc Gillespie, Phani Garapati, Robin Haw, Bijay Jassal, Florian Korninger, Bruce May, Marija Milacic, Corina Duenas Roca, Karen Rothfels, Cristoffer Sevilla, Veronica Shamovsky, Solomon Shorser, Thawfeek Varusai, Guilherme Viteri, Joel Weiser, Guanming Wu, Lincoln Stein, Henning Hermjakob, and Peter D'Eustachio. The reactome pathway knowledgebase. *Nucleic Acids Research*, 46(D1):D649–D655, November 2017.

[272] Ali Ebrahim, Joshua A Lerman, Bernhard O Palsson, and Daniel R Hyduke. COBRApy: COnstraints-based reconstruction and analysis for python. *BMC Systems Biology*, 7(1):74, 2013.

[273] Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, Leonid Zaslavsky, Jian Zhang, and Evan E Bolton. PubChem 2019 update: improved access to chemical data. *Nucleic Acids Research*, 47(D1):D1102–D1109, October 2018.

[274] Jonathan M Monk, Colton J Lloyd, Elizabeth Brunk, Nathan Mih, Anand Sastry, Zachary King, Rikiya Takeuchi, Wataru Nomura, Zhen Zhang, Hirotada Mori, Adam M Feist, and Bernhard O Palsson. iML1515, a knowledgebase that computes escherichia coli traits. *Nature Biotechnology*, 35(10):904–908, October 2017.

[275] Jeffrey D. Orth, Bernhard Ø. Palsson, and R. M. T. Fleming. Reconstruction and use of microbial metabolic networks: the core escherichia coli metabolic model as an educational guide. *EcoSal Plus*, 4(1), September 2010.

[276] Zachary A. King, Justin Lu, Andreas Dräger, Philip Miller, Stephen Federowicz, Joshua A. Lerman, Ali Ebrahim, Bernhard O. Palsson, and Nathan E. Lewis. BiGG models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Research*, 44(D1):D515–D522, October 2015.

[277] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning series. MIT Press, 2018.

[278] J. Wang, J. Tian, and J.H. Wu. A method for producing terephthalic acid by comamonas testosteroni dsm6577. *Chinese Journal of Catalysis*, 27(4):297–298, 2006.

[279] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

[280] Johannes Köster and Sven Rahmann. Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics*, 34(20):3600–3600, May 2018.

[281] M. R. Connor and J. C. Liao. Engineering of an escherichia coli strain for the production of 3-methyl-1-butanol. *Applied and Environmental Microbiology*, 74(18):5769–5775, August 2008.

[282] Hee Jin Hwang, Jin Hwan Park, Jin Ho Kim, Min Kyung Kong, Jin Won Kim, Jin Woo Park, Kwang Myung Cho, and Pyung Cheon Lee. Engineering of a butyraldehyde dehydrogenase of-Clostridium saccharoperbutylacetonicumto fit an engineered 1, 4-butanediol pathway inEscherichia coli. *Biotechnology and Bioengineering*, 111(7):1374–1384, February 2014.

[283] Yuchang Luo, Qinqin Zhao, Qian Liu, and Yan Feng. An artificial biosynthetic pathway for 2-amino-1, 3-propanediol production using metabolically engineered escherichia coli. *ACS Synthetic Biology*, 8(3):548–556, February 2019.

[284] Xiaolin Shen, Jia Wang, Bradley K. Gall, Eric M. Ferreira, Qipeng Yuan, and Yajun Yan. Establishment of novel biosynthetic pathways for the production of salicyl alcohol and gentisyl alcohol in engineered escherichia coli. *ACS Synthetic Biology*, 7(4):1012–1017, March 2018.

[285] Shawn Pugh, Rebekah McKenna, Ibrahim Halloum, and David R. Nielsen. Engineering escherichia coli for renewable benzyl alcohol production. *Metabolic Engineering Communications*, 2:39–45, December 2015.

[286] Jianming Yang and Lizhong Guo. Biosynthesis of β-carotene in engineered e. coli using the MEP and MVA pathways. *Microbial Cell Factories*, 13(1), November 2014.

[287] Yuheng Lin, Xinxiao Sun, Qipeng Yuan, and Yajun Yan. Extending shikimate pathway for the production of muconic acid and its precursor salicylic acid in escherichia coli. *Metabolic Engineering*, 23:62–69, May 2014.

[288] Miso Park, Shen-Long Tsai, and Wilfred Chen. Microbial biosensors: Engineered microorganisms as the sensing machinery. *Sensors*, 13(5):5777–5795, May 2013.

[289] Tian Ma, Bin Shi, Ziling Ye, Xiaowei Li, Min Liu, Yun Chen, Jiang Xia, Jens Nielsen, Zixin Deng, and Tiangang Liu. Lipid engineering combined with systematic metabolic engineering of saccharomyces cerevisiae for high-yield production of lycopene. *Metabolic Engineering*, 52:134–142, March 2019.

[290] Jingyu Wang and Kechun Zhang. Production of mesaconate in escherichia coli by engineered glutamate mutase pathway. *Metabolic Engineering*, 30:190–196, July 2015.

[291] Christine Nicole S. Santos, Mattheos Koffas, and Gregory Stephanopoulos. Optimization of a heterologous pathway for the production of flavonoids from glucose. *Metabolic Engineering*, 13(4):392–400, July 2011.

[292] Yu Ping, Xiaodong Li, Baofu Xu, Wei Wei, Wenping Wei, Guoyin Kai, Zhihua Zhou, and Youli Xiao. Building microbial hosts for heterologous production of n-methylpyrrolinium. *ACS Synthetic Biology*, 8(2):257–263, January 2019.

[293] W Qi, T Vannelli, S Breinig, A Benbassat, A Gatenby, S Haynie, and F Sariaslani. Functional expression of prokaryotic and eukaryotic genes in escherichia coli for conversion of glucose to pp-hydroxystyrene. *Metabolic Engineering*, 9(3):268–276, May 2007.

[294] Anil Shrestha, Ramesh Prasad Pandey, and Jae Kyung Sohng. Biosynthesis of resveratrol and piceatannol in engineered microbial strains: achievements and perspectives. *Applied Microbiology and Biotechnology*, 103(7):2959–2972, February 2019.

[295] Bong Gyu Kim, Hyejin Lee, , and Joong-Hoon Ahn. Biosynthesis of pinocembrin from glucose using engineered escherichia coli. *Journal of Microbiology and Biotechnology*, 24(11):1536–41, 2014.

[296] Rebekah McKenna and David R. Nielsen. Styrene biosynthesis from glucose by engineered e. coli. *Metabolic Engineering*, 13(5):544–554, September 2011.

[297] Aditya M. Kunjapur, Yekaterina Tarasova, and Kristala L. J. Prather. Synthesis and accumulation of aromatic aldehydes in an engineered strain ofEscherichia coli. *Journal of the American Chemical Society*, 136(33):11644–11654, August 2014.

[298] J. Andrew Jones, Victoria R. Vernacchio, Daniel M. Lachance, Matthew Lebovich, Li Fu, Abhijit N. Shirke, Victor L. Schultz, Brady Cress, Robert J. Linhardt, and Mattheos A. G. Koffas. ePathOptimize: A combinatorial approach for transcriptional balancing of metabolic pathways. *Scientific Reports*, 5(1), June 2015.

[299] Tsutomu Hoshino. Violacein and related tryptophan metabolites produced by chromobacterium violaceum: biosynthetic mechanism and pathway for construction of violacein core. *Applied Microbiology and Biotechnology*, 91(6):1463–1475, July 2011.

[300] Tomas Kozelek. Methods of mcts and the game arimaa. *Charles Univ., Prague*, 2009.

[301] Guillermo Carbajosa, Almudena Trigo, Alfonso Valencia, and Ildefonso Cases. Bionemo: molecular information on biodegradation metabolism. *Nucleic acids research*, 37(Database issue):D598–602, jan 2009.

[302] Michael J Cipriano, Pavel N Novichkov, Alexey E Kazakov, Dmitry A Rodionov, Adam P Arkin, Mikhail S Gelfand, and Inna Dubchak. RegTransBase – a database of regulatory sequences and interactions based on literature: a resource for investigating transcriptional regulation in prokaryotes. *BMC Genomics*, 14(1):213, apr 2013.

[303] Socorro Gama-Castro, Heladia Salgado, Alberto Santos-Zavaleta, Daniela Ledezma-Tejeida, Luis Muñiz-Rascado, Jair Santiago García-Sotelo, Kevin Alquicira-Hernández, Irma Martínez-Flores, Lucia Pannier, Jaime Abraham Castro-Mondragón, Alejandra Medina-Rivera, Hilda Solano-Lira, César Bonavides-Martínez, Ernesto Pérez-Rueda, Shirley Alquicira-Hernández, Liliana Porrón-Sotelo, Alejandra López-Fuentes, Anastasia Hernández-Koutoucheva, Víctor Del Moral-Chávez, Fabio Rinaldi, and Julio Collado-Vides. RegulonDB version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic acids research*, 44(D1):D133–143, jan 2016.

[304] Pavel S Novichkov, Alexey E Kazakov, Dmitry A Ravcheev, Semen A Leyn, Galina Y Kovaleva, Roman A Sutormin, Marat D Kazanov, William Riehl, Adam P Arkin, Inna Dubchak, and Dmitry A Rodionov. RegPrecise 3.0–a resource for genome-scale exploration of transcriptional regulation in bacteria. *BMC genomics*, 14(1):745, nov 2013.

[305] Akanksha Rajput, Karambir Kaur, and Manoj Kumar. SigMol: repertoire of quorum sensing signaling molecules in prokaryotes. *Nucleic acids research*, 44(D1):D634–639, jan 2016.

[306] Cédric Orelle, Erik D. Carlson, Teresa Szal, Tanja Florin, Michael C. Jewett, and Alexander S. Mankin. Protein synthesis by ribosomes with tethered subunits. *Nature*, 524(7563):119–124, July 2015.

[307] E. Karzbrun, A. M. Tayar, V. Noireaux, and R. H. Bar-Ziv. Programmable on-chip DNA compartments as artificial cells. *Science*, 345(6198):829–832, August 2014.

[308] Simon J. Moore, James T. MacDonald, Sarah Wienecke, Alka Ishwarbhai, Argyro Tsipa, Rochelle Aw, Nicolas Kylilis, David J. Bell, David W. McClymont, Kirsten Jensen, Karen M. Polizzi, Rebekka Biedendieck, and Paul S. Freemont. Rapid acquisition and model-based analysis of cell-free transcrip-

tion–translation reactions from nonmodel bacteria. *Proceedings of the National Academy of Sciences*, 115(19):E4340–E4349, apr 2018.

[309] Melissa K. Takahashi, James Chappell, Clarmyra A. Hayes, Zachary Z. Sun, Jongmin Kim, Vipul Singhal, Kevin J. Spring, Shaima Al-Khabouri, Christopher P. Fall, Vincent Noireaux, Richard M. Murray, and Julius B. Lucks. Rapidly Characterizing the Fast Dynamics of RNA Genetic Circuitry with Cell-Free Transcription-Translation (TX-TL) Systems. *ACS Synthetic Biology*, 4(5):503–515, may 2015.

[310] C. Eric Hodgman and Michael C. Jewett. Cell-free synthetic biology: Thinking outside the cell. *Metabolic Engineering*, 14(3):261–269, 2012.

[311] Eyal Karzbrun, Jonghyeon Shin, Roy H. Bar-Ziv, and Vincent Noireaux. Coarse-grained dynamics of protein synthesis in a cell-free system. *Phys. Rev. Lett.*, 106:048104, Jan 2011.

[312] Jennifer A. Schoborg and Michael C. Jewett. Cell-free protein synthesis: An emerging technology for understanding, harnessing, and expanding the capabilities of biological systems. In *Synthetic Biology*, pages 307–330. Wiley-VCH Verlag GmbH & Co. KGaA, March 2018.

[313] Balbas, Lorence, James R. Swartz, Michael C. Jewett, and Kim A. Woodrow. Cell-free protein synthesis with prokaryotic combined transcription-translation. In *Recombinant Gene Expression*, pages 169–182. Humana Press, 2004.

[314] Diogo M. Camacho, Katherine M. Collins, Rani K. Powers, James C. Costello, and James J. Collins. Next-generation machine learning for biological networks. *Cell*, 173(7):1581–1592, June 2018.

[315] S. G. Grant, J. Jessee, F. R. Bloom, and D. Hanahan. Differential plasmid rescue from transgenic mouse DNAs into escherichia coli methylation-restriction mutants. *Proceedings of the National Academy of Sciences*, 87(12):4645–4649, June 1990.

[316] Daniel J. Wiegand, Henry H. Lee, Nili Ostrov, and George M. Church. Establishing a cell-free vibrio natriegens expression system. *ACS Synthetic Biology*, 7(10):2475–2479, August 2018.

[317] Richard Kelwick, Alexander J. Webb, James T. MacDonald, and Paul S. Freemont. Development of a bacillus subtilis cell-free transcription-translation system for prototyping regulatory elements. *Metabolic Engineering*, 38:370–381, November 2016.

[318] Stephanie D. Cole, Kathryn Beabout, Kendrick B. Turner, Zachary K. Smith, Vanessa L. Funk, Svetlana V Harbaugh, Alvin T. Liem, Pierce A. Roth, Brian A. Geier, Peter A. Emanuel, Scott A. Walper, Jorge Luis Chávez, and Matthew W. Lux. Quantification of interlaboratory cell-free protein synthesis variability. *ACS Synthetic Biology*, August 2019.

[319] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E*, 69(6), June 2004.

[320] Jonghyeon Shin and Vincent Noireaux. Efficient cell-free expression with the endogenous E. Coli RNA polymerase and sigma factor 70. *Journal of Biological Engineering*, 4(1):8, jun 2010.

[321] Di Liu, Trent Evans, and Fuzhong Zhang. Applications and advances of metabolite biosensors for metabolic engineering. *Metabolic engineering*, 31:35–43, sep 2015.

[322] Ulysses Amancio de Frias, Greicy Kelly Bonifacio Pereira, María-Eugenia Guazzaroni, and Rafael Silva-Rocha. Boosting secondary metabolite production and discovery through the engineering of novel microbial biosensors. *BioMed Research International*, 2018:1–11, jul 2018.

[323] Yang Liu, Ye Liu, and Meng Wang. Design, Optimization and Application of Small Molecule Biosensor in Metabolic Engineering. *Frontiers in microbiology*, 8:2012, 2017.

[324] Seema Ameen, Mohammad Ahmad, Mohd. Mohsin, M. Irfan Qureshi, Mohamed M. Ibrahim, Malik Z. Abdin, and Altaf Ahmad. Designing, construction and characterization of genetically encoded FRET-based nanosensor for real time monitoring of lysine flux in living cells. *Journal of Nanobiotechnology*, 14(1):49, dec 2016.

[325] Yu Xiu, Sungho Jang, J. Andrew Jones, Nicholas A. Zill, Robert J. Linhardt, Qipeng Yuan, Gyoo Yeol Jung, and Mattheos A. G. Koffas. Naringenin-responsive riboswitch-based fluorescent biosensor module for Escherichia coli co-cultures. *Biotechnology and Bioengineering*, 114(10):2235–2244, oct 2017.

[326] Annamaria Ruscito, Erin M. McConnell, Anna Koudrina, Ranganathan Velu, Christopher Mattice, Vernon Hunt, Maureen McKeague, and Maria C. DeRosa. In Vitro Selection and Characterization of DNA Aptamers to a Small Molecule Target. In *Current Protocols in Chemical Biology*, volume 9, pages 233–268. John Wiley & Sons, Inc., Hoboken, NJ, USA, dec 2017.

[327] Alexander Carpenter, Ian Paulsen, and Thomas Williams. Blueprints for Biosensors: Design, Limitations, and Applications. *Genes*, 9(8):375, jul 2018.

[328] Jameson K. Rogers, Noah D. Taylor, George M. Church, Christoph Wittmann, and Ramon Gonzalez. Biosensor-based engineering of biosynthetic pathways. *Current Opinion in Biotechnology*, 42:84–91, 2016.

[329] Shuobo Shi, Yook Wah Choi, Huimin Zhao, Meng How Tan, and Ee Lui Ang. Discovery and engineering of a 1-butanol biosensor in saccharomyces cerevisiae. *Bioresource Technology*, 245:1343–1351, December 2017.

[330] Jingwei Zhang, Jesus F. Barajas, Mehmet Burdu, Thomas L. Ruegg, Bryton Dias, and Jay D. Keasling. Development of a Transcription Factor-Based Lactam Biosensor. *ACS Synthetic Biology*, 6(3):439–445, mar 2017.

[331] Vincent Libis, B Delepine, and Jean-Loup Faulon. Sensing new chemicals with bacterial transcription factors. *Curr. Opin. Microbiol.*, 33:105–112, 2016.

[332] Erik K. R. Hanko, Nigel Peter Minton, and Naglis Malys. A transcription factor-based biosensor for detection of itaconic acid. *ACS Synthetic Biology*, 7:1436–1446, apr 2018.

[333] Joe C. H. Ho, Sandip V. Pawar, Steven J. Hallam, and Vikramaditya G. Yadav. An improved whole-cell biosensor for the discovery of lignin-transforming enzymes in functional metagenomic screens. *ACS Synthetic Biology*, 7(2):392–398, December 2017.

[334] Tim Snoek, David Romero-Suarez, Jie Zhang, Francesca Ambri, Mette L. Skjoedt, Suresh Sudarsan, Michael K. Jensen, and Jay D. Keasling. An orthogonal and pH-tunable sensor-selector for muconic acid biosynthesis in yeast. *ACS Synthetic Biology*, 7(4):995–1003, April 2018.

[335] Yuriy Rebets, Stefan Schmelz, Oleksandr Gromyko, Stepan Tistechok, Lutz Petzke, Andrea Scrima, and Andriy Luzhetskyy. Design, development and application of whole-cell based antibiotic-specific biosensor. *Metabolic Engineering*, 47:263–270, 2018.

[336] Solvej Siedler, Narendar K. Khatri, Andrea Zsohár, Inge Kjærbølling, Michael Vogt, Petter Hammar, Christian F. Nielsen, Jan Marienhagen, Morten O.A. Sommer, and Haakan N. Joensson. Development of a Bacterial Biosensor for Rapid Screening of Yeast p-Coumaric Acid Production. *ACS Synthetic Biology*, 6(10):1860–1869, 2017.

[337] Benjamin M. Woolston, Timothy Roth, Ishwar Kohale, David R. Liu, and Gregory Stephanopoulos. Development of a formaldehyde biosensor with application to synthetic methylotrophy. *Biotechnology and Bioengineering*, 115(1):206–215, sep 2018.

[338] Gert Peters, Brecht De Paepe, Lien De Wannemaeker, Dries Duchi, Jo Maertens, Jeroen Lammertyn, and Marjan De Mey. Development of N-acetylneuraminic acid responsive biosensors based on the transcriptional regulator NanR. *Biotechnology and Bioengineering*, 115:1855–1865, 2018.

[339] Xue-Feng Chen, Xiao-Xia Xia, Sang Yup Lee, and Zhi-Gang Qian. Engineering Tunable Biosensors for Monitoring Putrescine in Escherichia coli. *Biotechnology and Bioengineering*, 115:1014–1027, dec 2017.

[340] Yongfei Liu, Yinyin Zhuang, Dongqin Ding, Yiran Xu, Jibin Sun, and Dawei Zhang. Biosensor-Based Evolution and Elucidation of a Biosynthetic Pathway in Escherichia coli. *ACS Synthetic Biology*, 6(5):837–848, may 2017.

[341] Chang Liu, Bo Zhang, Yi-Ming Liu, Ke-Qian Yang, and Shuang-Jiang Liu. New intracellular shikimic acid biosensor for monitoring shikimate synthesis in corynebacterium glutamicum. *ACS Synthetic Biology*, 7(2):591–601, November 2018.

[342] Kil Koang Kwon, Soo-Jin Yeom, Dae-Hee Lee, Ki Jun Jeong, and Seung-Goo Lee. Development of a novel cellulase biosensor that detects crystalline cellulose hydrolysis using a transcriptional regulator. *Biochemical and Biophysical Research Communications*, 495(1):1328–1334, jan 2018.

[343] Brecht De Paepe, Jo Maertens, Bartel Vanholme, and Marjan De Mey. Modularization and response curve engineering of a naringenin-responsive transcriptional biosensor. *ACS Synthetic Biology*, 7(5):1303–1314, April 2018.

[344] Mette L Skjoedt, Tim Snoek, Kanchana R Kildegaard, Dushica Arsovska, Michael Eichenberger, Tobias J Goedecke, Arun S Rajkumar, Jie Zhang, Mette Kristensen, Beata J Lehka, Solvej Siedler, Irina Borodina, Michael K Jensen, and Jay D Keasling. Engineering prokaryotic transcriptional activators as metabolite biosensors in yeast. *Nature Chemical Biology*, 12(11):951–958, sep 2016.

[345] Leopoldo F. M. Machado and Neil Dixon. Development and substrate specificity screening of an in vivo biosensor for the detection of biomass derived aromatic chemical building blocks. *Chem. Commun.*, 52(76):11402–11405, sep 2016.

[346] Christian M. Kasey, Mounir Zerrad, Yiwei Li, T. Ashton Cropp, and Gavin J. Williams. Development of transcription factor-based designer macrolide biosensors for metabolic engineering and synthetic biology. *ACS Synthetic Biology*, 7(1):227–239, October 2018.

[347] Abayomi Oluwanbe Johnson, Miriam Gonzalez-Villanueva, Lynn Wong, Alexander Steinbüchel, Kang Lan Tee, Peng Xu, and Tuck Seng Wong. Design and application of genetically-encoded malonyl-CoA biosensors for metabolic engineering of microbial cell factories. *Metabolic Engineering*, 44(September):253–264, 2017.

[348] Francesca Ambri, Tim Snoek, Mette L. Skjoedt, Michael K. Jensen, and Jay D. Keasling. Design, engineering, and characterization of prokaryotic ligand-binding transcriptional activators as biosensors in yeast. In *Methods in Molecular Biology*, pages 269–290. Springer New York, 2018.

[349] Adam J. Meyer, Thomas H. Segall-Shapiro, Emerson Glassey, Jing Zhang, and Christopher A. Voigt. Escherichia coli "marionette" strains with 12 highly optimized small-molecule sensors. *Nature Chemical Biology*, 15(2):196–204, nov 2018.

[350] Jieyuan Wu, Peixia Jiang, Wei Chen, Dandan Xiong, Linglan Huang, Junying Jia, Yuanyuan Chen, Jian-Ming Jin, and Shuang-Yan Tang. Design and application of a lactulose biosensor. *Scientific Reports*, 7(1):45994, dec 2017.

[351] Noah D Taylor, Alexander S Garruss, Rocco Moretti, Sum Chan, Mark A Arbing, Duilio Cascio, Jameson K Rogers, Farren J Isaacs, Sriram Kosuri, David Baker, Stanley Fields, George M Church, and Srivatsan Raman. Engineering an allosteric transcription factor to respond to new ligands. *Nature methods*, 13(2):177–83, feb 2016.

[352] Andrew K. D. Younger, Neil C. Dalvie, Austin G. Rottinghaus, and Joshua N. Leonard. Engineering modular biosensors to confer metabolite-responsive regulation of transcription. *ACS Synthetic Biology*, 6(2):311–325, October 2017.

[353] Javier F. Juárez, Begoña Lecube-Azpeitia, Stuart L. Brown, Christopher D. Johnston, and George M. Church. Biosensor libraries harness large classes of binding domains for construction of allosteric transcriptional regulators. *Nature communications*, 9(1):3101, aug 2018.

[354] Andrew K D Younger, Peter Y Su, Andrea J Shepard, Shreya V Udani, Thaddeus R Cybulski, Keith E J Tyo, and Joshua N Leonard. Development of novel metabolite-responsive transcription factors via transposon-mediated protein fusion. *Protein Engineering, Design and Selection*, 31(February):55–63*, feb 2018.

[355] Benjamin M. Brandsen, Jordan M. Mattheisen, Teia Noel, and Stanley Fields. A biosensor strategy for e. coli based on ligand-dependent stabilization. *ACS Synthetic Biology*, 7(9):1990–1999, jul 2018.

[356] Benjamin W. Jester, Christine E. Tinberg, Matthew S. Rich, David Baker, and Stanley Fields. Engineered biosensors from dimeric ligand-binding domains. *ACS Synthetic Biology*, 7(10):2457–2467, sep 2018.

[357] Vidhya Selvamani, Irisappan Ganesh, Murali kannan Maruthamuthu, Gyeong Tae Eom, and Soon Ho Hong. Engineering chimeric two-component system into Escherichia coli from Paracoccus denitrificans to sense methanol. *Biotechnology and Bioprocess Engineering*, 22(3):225–230, jun 2017.

[358] Hung-Ju Chang, Pauline Mayonove, Agustin Zavala, Angelique De Visch, Philippe Minard, Martin Cohen-Gonsaud, and Jerome Bonnet. A modular receptor platform to expand the sensing repertoire of bacteria. *ACS Synthetic Biology*, 7(1):166–175, October 2018.

[359] Jameson K Rogers and George M Church. Genetically encoded sensors enable real-time observation of metabolite production. *Proceedings of the National Academy of Sciences of the United States of America*, 113(9):2388–93, mar 2016.

[360] Ye Chen, Joanne M. L. Ho, David L. Shis, Chinmaya Gupta, James Long, Daniel S. Wagner, William Ott, Krešimir Josić, and Matthew R. Bennett. Tuning the dynamic range of bacterial promoters regulated by ligand-inducible transcription factors. *Nature Communications*, 9(1):64, dec 2018.

[361] Ahmad A. Mannan, Di Liu, Fuzhong Zhang, and Diego A. Oyarzún. Fundamental Design Principles for Transcription-Factor-Based Metabolite Biosensors. *ACS Synthetic Biology*, 6(10):1851–1859, oct 2017.

[362] Brian P. Landry, Rohan Palanki, Nikola Dyulgyarov, Lucas A. Hartsough, and Jeffrey J. Tabor. Phosphatase activity tunes two-component system sensor detection threshold. *Nature Communications*, 9(1):1433, dec 2018.

[363] Phuc H.B. Nguyen, Yong Wu, Shaobin Guo, and Richard M Murray. Design Space Exploration of the Violacein Pathway in Escherichia coli Based Transcription Translation Cell-Free System (TX-TL). *bioRxiv*, jan 2016.

[364] Yong Y Wu, Stephanie Culler, Julia Khandurina, Stephen Van Dien, and Richard M Murray. Prototyping 1,4-butanediol (BDO) biosynthesis pathway in a cell-free transcription-translation (TX-TL) system. *bioRxiv*, apr 2015.

[365] Richard Kelwick, Luca Ricci, Soo Mei Chee, David Bell, Alexander J Webb, and Paul S Freemont. Cell-free prototyping strategies for enhancing the sustainable production of polyhydroxyalkanoates bioplastics. *bioRxiv*, nov 2017.

[366] Simon J Moore, Tommaso Tosi, Yonek B Hleba, David Bell, Karen Polizzi, and Paul Freemont. A cell-free synthetic biochemistry platform for raspberry ketone production. *bioRxiv*, oct 2017.

[367] Lihong Jiang, Jiarun Zhao, Jiazhang Lian, and Zhinan Xu. Cell-free protein synthesis enabled rapid prototyping for metabolic engineering and synthetic biology. *Synthetic and Systems Biotechnology*, 3(2):90–96, jun 2018.

[368] Ashty S. Karim and Michael C. Jewett. Cell-Free Synthetic Biology for Pathway Prototyping. *Methods in Enzymology*, 608:31–58, 2018.

[369] Mehran Soltani, Brady R. Davis, Hayley Ford, J. Andrew D. Nelson, and Bradley C. Bundy. Reengineering cell-free protein synthesis as a biosensor: Biosensing with transcription, translation, and protein-folding. *Biochemical Engineering Journal*, 138:165–171, oct 2018.

[370] Mitchell Tai and Gregory Stephanopoulos. Engineering the push and pull of lipid biosynthesis in oleaginous yeast yarrowia lipolytica for biofuel production. *Metabolic Engineering*, 15:1–9, January 2013.

[371] Gregory Stephanopoulos. Synthetic biology and metabolic engineering. *ACS Synthetic Biology*, 1(11):514–525, November 2012.

[372] E. I. Lan and J. C. Liao. ATP drives direct photosynthetic production of 1-butanol in cyanobacteria. *Proceedings of the National Academy of Sciences*, 109(16):6018–6023, April 2012.

[373] Amarjeet Singh, Keng Cher Soh, Vassily Hatzimanikatis, and Ryan T. Gill. Manipulating redox and ATP balancing for improved production of succinate in e. coli. *Metabolic Engineering*, 13(1):76–81, January 2011.

[374] Mary J. Dunlop, Jay D. Keasling, and Aindrila Mukhopadhyay. A model for improving microbial biofuel production using a synthetic feedback loop. *Systems and Synthetic Biology*, 4(2):95–104, February 2010.

[375] Mary E. Harrison and Mary J. Dunlop. Synthetic feedback loop model for increasing microbial biofuel production using a biosensor. *Frontiers in Microbiology*, 3, 2012.

[376] P. Xu, L. Li, F. Zhang, G. Stephanopoulos, and M. Koffas. Improving fatty acids production by engineering dynamic pathway regulation and metabolic control. *Proceedings of the National Academy of Sciences*, 111(31):11299–11304, July 2014.

[377] Brian F. Pfleger, Douglas J. Pitera, Jack D. Newman, Vincent J.J. Martin, and Jay D. Keasling. Microbial sensors for small molecules: Development of a mevalonate biosensor. *Metabolic Engineering*, 9(1):30–38, January 2007.

[378] Azhar Rasul, Faya Martin Millimouno, Wafa Ali Eltayb, Muhammad Ali, Jiang Li, and Xiaomeng Li. Pinocembrin: A novel natural compound with versatile pharmacological and biological activities. *BioMed Research International*, 2013:1–9, 2013.

[379] Roderick J Weston, Kevin R Mitchell, and Kerry L Allen. Antibacterial phenolic components of new zealand manuka honey. *Food Chemistry*, 64(3):295–301, February 1999.

[380] Litao Peng, Shuzhen Yang, Yun Jiang Cheng, Feng Chen, Siyi Pan, and Gang Fan. Antifungal activity and action mode of pinocembrin from propolis against penicillium italicum. *Food Science and Biotechnology*, 21(6):1533–1539, December 2012.

[381] Nana Yang, Shucun Qin, Mengzan Wang, Bin Chen, Na Yuan, Yongqi Fang, Shutong Yao, Peng Jiao, Yang Yu, Ying Zhang, and Jiafu Wang. Pinocembrin, a major flavonoid in propolis, improves the biological functions of EPCs derived from rat bone marrow through the PI3k-eNOS-NO signaling pathway. *Cytotechnology*, 65(4):541–551, November 2012.

[382] Rui Liu, Cai xia Wu, Dan Zhou, Fan Yang, Shuo Tian, Li Zhang, Tian tai Zhang, and Guan hua Du. Pinocembrin protects against β-amyloid-induced toxicity in neurons through inhibiting receptor for advanced glycation end products (RAGE)-independent signaling pathways and regulating mitochondrion-mediated apoptosis. *BMC Medicine*, 10(1):105, September 2012.

[383] Rui Liu, Mei Gao, Zhi-Hong Yang, and Guan-Hua Du. Pinocembrin protects rat brain against oxidation and apoptosis induced by ischemia–reperfusion both in vivo and in vitro. *Brain Research*, 1216:104–115, June 2008.

[384] Junjun Wu, Guocheng Du, Jingwen Zhou, and Jian Chen. Metabolic engineering of escherichia coli for (2s)-pinocembrin production from glucose by a modular metabolic strategy. *Metabolic Engineering*, 16:48–55, March 2013.

[385] A. M. Marin, E. M. Souza, F. O. Pedrosa, L. M. Souza, G. L. Sassaki, V. A. Baura, M. G. Yates, R. Wassem, and R. A. Monteiro. Naringenin degradation by the endophytic diazotroph herbaspirillum seropedicae SmR1. *Microbiology*, 159:167–175, November 2013.

[386] Solvej Siedler, Steen G. Stahlhut, Sailesh Malla, Jérôme Maury, and Ana Rute Neves. Novel biosensors based on flavonoid-responsive transcriptional regulators introduced into escherichia coli. *Metabolic Engineering*, 21:2–8, January 2014.

[387] Sunghwan Kim, Paul A. Thiessen, Evan E. Bolton, Jie Chen, Gang Fu, Asta Gindulyte, Lianyi Han, Jane He, Siqian He, Benjamin A. Shoemaker, Jiyao Wang, Bo Yu, Jian Zhang, and Stephen H. Bryant. PubChem substance and compound databases. *Nucleic Acids Research*, 44(D1):D1202–D1213, September 2015.

[388] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2016.

[389] Karline Soetaert, Thomas Petzoldt, and R. Woodrow Setzer. Solving differential equations in r: Package desolve. *Journal of Statistical Software*, 33(9):1–25, 2010.

[390] John E. Dennis, David M. Gay, and Roy E. Walsh. An adaptive nonlinear least-squares algorithm. *ACM Transactions on Mathematical Software*, 7(3):348–368, September 1981.

[391] Srivatsan Raman, Jameson K. Rogers, Noah D. Taylor, and George M. Church. Evolution-guided optimization of biosynthetic pathways. *Proceedings of the National Academy of Sciences*, 111(50):17803–17808, December 2014.

[392] Lacramioara Bintu, Nicolas E Buchler, Hernan G Garcia, Ulrich Gerland, Terence Hwa, Jané Kondev, Thomas Kuhlman, and Rob Phillips. Transcriptional regulation by the numbers: applications. *Current Opinion in Genetics & Development*, 15(2):125–135, April 2005.

[393] Mattias Rydenfelt, Robert Sidney Cox, Hernan Garcia, and Rob Phillips. Statistical mechanical model of coupled transcription from multiple promoters due to transcription factor titration. *Physical Review E-Statistical, Nonlinear, and Soft Matter Physics,*, 89(1), January 2014.

[394] Jeffrey A. Dietrich, Adrienne E. McKee, and Jay D. Keasling. High-throughput metabolic engineering: Advances in small-molecule screening and selection. *Annual Review of Biochemistry*, 79(1):563–590, June 2010.

[395] Lothar Eggeling, Michael Bott, and Jan Marienhagen. Novel screening methods—biosensors. *Current Opinion in Biotechnology*, 35:30–36, December 2015.

[396] Xiulai Chen and Liming Liu. Gene Circuits for Dynamically Regulating Metabolism. *Trends in Biotechnology*, xx:1–4, 2018.

[397] Jameson K. Rogers, Christopher D. Guzman, Noah D. Taylor, Srivatsan Raman, Kelley Anderson, and George M. Church. Synthetic biosensors for precise gene control and real-time monitoring of metabolites. *Nucleic Acids Research*, 43(15):7648–7660, July 2015.

[398] Loren L. Looger, Mary A. Dwyer, James J. Smith, and Homme W. Hellinga. Computational design of receptor and sensor proteins with novel functions. *Nature*, 423(6936):185–190, May 2003.

[399] Daniel J Mandell and Tanja Kortemme. Computer-aided design of functional protein interactions. *Nature Chemical Biology*, 5(11):797–807, November 2009.

[400] Marcus Schallmey, Julia Frunzke, Lothar Eggeling, and Jan Marienhagen. Looking for the pick of the bunch: high-throughput screening of producing microorganisms with biosensors. *Current Opinion in Biotechnology*, 26:148–154, April 2014.

[401] Shuang-Yan Tang, Hossein Fazelinia, and Patrick C. Cirino. AraC regulatory protein mutants with altered effector specificity. *Journal of the American Chemical Society*, 130(15):5267–5271, April 2008.

[402] Shuang-Yan Tang and Patrick C. Cirino. Design and application of a mevalonate-responsive regulatory protein. *Angewandte Chemie International Edition*, 50(5):1084–1086, December 2011.

[403] Shuang-Yan Tang, Shuai Qian, Olubolaji Akinterinwa, Christopher S. Frei, Joseph A. Gredell, and Patrick C. Cirino. Screening for enhanced triacetic acid lactone production by recombinant escherichia coli expressing a designed triacetic acid lactone reporter. *Journal of the American Chemical Society*, 135(27):10099–10103, July 2013.

[404] Melissa K. Takahashi, Clarmyra A. Hayes, James Chappell, Zachary Z. Sun, Richard M. Murray, Vincent Noireaux, and Julius B. Lucks. Characterizing and prototyping genetic networks with cell-free transcription-translation reactions. *Methods*, 86:60–72, sep 2015.

[405] D M Kim and J R Swartz. Regeneration of adenosine triphosphate from glycolytic intermediates for cell-free protein synthesis. *Biotechnology and bioengineering*, 74(4):309–316, aug 2001.

[406] Kara A. Calhoun and James R. Swartz. Energizing cell-free protein synthesis with glucose metabolism. *Biotechnology and Bioengineering*, 90(5):606–613, jun 2005.

[407] Yong Y. Wu, Hirokazu Sato, Hongjun Huang, Stephanie J. Culler, Julia Khandurina, Harish Nagarajan, Tae Hoon Yang, Stephen Van Dien, Richard M. Murray, Stephen Van Dien, and Richard M. Murray. System-level studies of a cell-free transcription-translation platform for metabolic engineering. *bioRxiv*, pages 1–14, aug 2017.

[408] Evan Spruijt, Ekaterina Sokolova, and Wilhelm T. S. Huck. Complexity of molecular crowding in cell-free enzymatic reaction networks. *Nature Nanotechnology*, 9(6):406–407, 2014.

[409] Tobias Stögbauer, Lukas Windhager, Ralf Zimmer, and Joachim O. Rädler. Experiment and mathematical modeling of gene expression dynamics in a cell-free system. *Integrative Biology*, 4(5):494–501, 2012.

[410] Dan Siegal-Gaskins, Zoltan A. Tuza, Jongmin Kim, Vincent Noireaux, and Richard M. Murray. Gene circuit performance characterization and resource usage in a cell-free "breadboard". *ACS Synthetic Biology*, 3(6):416–425, jun 2014.

[411] Kelly A. Underwood, James R. Swartz, and Joseph D. Puglisi. Quantitative polysome analysis identifies limitations in bacterial cell-free protein synthesis. *Biotechnology and Bioengineering*, 91(4):425–435, 2005.

[412] Andras Gyorgy and Richard M. Murray. Quantifying resource competition and its effects in the TX-TL system. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 3363–3368. IEEE, dec 2016.

[413] Wolfgang Halter, Frank Allgower, Richard M. Murray, and Andras Gyorgy. Optimal experiment design and leveraging competition for shared resources in cell-free extracts. In *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, dec 2018.

[414] Dan Siegal-gaskins, Vincent Noireaux, and Richard M Murray. Biomolecular resource utilization in elementary cell-free gene circuits. *American Control Conference (ACC), 2013*, pages 1531–1536, 2013.

[415] Zoltan A. Tuza, Vipul Singhal, Jongmin Kim, and Richard M. Murray. An in silico modeling toolbox for rapid prototyping of circuits in a biomolecular breadboard system. In *52nd IEEE Conference on Decision and Control*. IEEE, dec 2013.

[416] Alexander Nieß, Jurek Failmezger, Maike Kuschel, Martin Siemann-Herzberg, and Ralf Takors. Experimentally validated model enables debottlenecking of in vitro protein synthesis and identifies a control shift under in vivo conditions. *ACS Synthetic Biology*, 6(10):1913–1921, oct 2017.

[417] Tomoaki Matsuura, Kazufumi Hosoda, and Yoshihiro Shimizu. Robustness of a reconstituted escherichia coli protein translation system analyzed by computational modeling. *ACS Synthetic Biology*, 7(8):1964–1972, jul 2018.

[418] Michael Vilkhovoy, Nicholas Horvath, Che-Hsiao Shih, Joseph A. Wayman, Kara Calhoun, James Swartz, and Jeffrey D. Varner. Sequence specific modeling of e. coli cell-free protein synthesis. *ACS Synthetic Biology*, 7(8):1844–1857, jun 2018.

[419] Michael C Jewett, Kara A Calhoun, Alexei Voloshin, Jessica J Wuu, and James R Swartz. An integrated cell-free metabolic platform for protein production and synthetic biology. *Molecular Systems Biology*, 4:220, oct 2008.

[420] Jurek Failmezger, Robert Nitschel, Andrés Sánchez-Kopper, Michael Kraml, and Martin Siemann-Herzberg. Site-Specific Cleavage of Ribosomal RNA in Escherichia coli-Based Cell-Free Protein Synthesis Systems. *PLOS ONE*, 11(12):e0168764, dec 2016.

[421] D.-M. Kim and J.R. Swartz. Prolonging cell-free protein synthesis by selective reagent additions. *Biotechnology Progress*, 16(3):385–390, jun 2000.

[422] Vijayalakshmi H Nagaraj, James M Greene, Anirvan M Sengupta, and Eduardo D Sontag. Translation inhibition and resource balance in the TX-TL cell-free gene expression system. *Synthetic Biology*, 2(1), jan 2017.

[423] Jun Li, Liangcai Gu, John Aach, and George M. Church. Improved Cell-Free RNA and Protein Synthesis System. *PLOS ONE*, 9(9):e106232, 2014.

[424] Gabriele Lillacci and Mustafa Khammash. Parameter estimation and model selection in computational biology. *PLoS Computational Biology*, 6(3):e1000696, mar 2010.

[425] Michael C. Jewett and James R. Swartz. Mimicking the Escherichia coli Cytoplasmic Environment Activates Long-Lived and Efficient Cell-Free Protein Synthesis. *Biotechnology and Bioengineering*, 86(1):19–26, apr 2004.

[426] Michael C. Jewett and James R. Swartz. Substrate replenishment extends protein synthesis with an in vitro translation system designed to mimic the cytoplasm. *Biotechnology and Bioengineering*, 87(4):465–471, aug 2004.
[427] WHO. Tracking universal health coverage: 2017 global monitoring report. In *World Health Organization and International Bank for Reconstruction and Development*. 2017.
[428] WHO. World health statistics 2018: Monitoring health for the sdgs. In *World Health Organization*. 2018.
[429] Raul Fernandez-López, Raul Ruiz, Fernando de la Cruz, and Gabriel Moncalián. Transcription factor-based biosensors enlightened by the analyte. *Frontiers in Microbiology*, 6:648, July 2015.
[430] Jan Roelof van der Meer and Shimshon Belkin. Where microbiology meets microengineering: design and applications of reporter bacteria. *Nature Reviews Microbiology*, 8(7):511–522, June 2010.
[431] Nilesh Raut, Gregory O'Connor, Patrizia Pasini, and Sylvia Daunert. Engineered cells as biosensing systems in biomedical analysis. *Analytical and Bioanalytical Chemistry*, 402(10):3147–3159, February 2012.
[432] Jonathan Garamella, Ryan Marshall, Mark Rustad, and Vincent Noireaux. The all e. coli TX-TL toolbox 2.0: A platform for cell-free synthetic biology. *ACS Synthetic Biology*, 5(4):344–355, February 2016.
[433] Jonghyeon Shin and Vincent Noireaux. An e. coli cell-free expression toolbox: Application to synthetic gene circuits and artificial cells. *ACS Synthetic Biology*, 1(1):29–41, January 2012.
[434] Roberta Lentini, Silvia Perez Santero, Fabio Chizzolini, Dario Cecchi, Jason Fontana, Marta Marchioretto, Cristina Del Bianco, Jessica L. Terrell, Amy C. Spencer, Laura Martini, Michele Forlin, Michael Assfalg, Mauro Dalla Serra, William E. Bentley, and Sheref S. Mansy. Integrating artificial with natural cells to translate chemical messages that direct e. coli behaviour. *Nature Communications*, 5(1):4012, May 2014.
[435] Keith Pardee, Alexander A. Green, Melissa K. Takahashi, Dana Braff, Guillaume Lambert, Jeong Wook Lee, Tom Ferrante, Duo Ma, Nina Donghia, Melina Fan, Nichole M. Daringer, Irene Bosch, Dawn M. Dudley, David H. O'Connor, Lee Gehrke, and James J. Collins. Rapid, low-cost detection of zika virus using programmable biomolecular components. *Cell*, 165(5):1255–1266, May 2016.
[436] Marnix H. Medema, Rainer Breitling, Roel Bovenberg, and Eriko Takano. Exploiting plug-and-play synthetic biology for drug discovery and production in microorganisms. *Nature Reviews Microbiology*, 9(2):131–137, December 2011.
[437] Christopher T. Walsh and Michael A. Fischbach. Natural products version 2.0: Connecting genes to molecules. *Journal of the American Chemical Society*, 132(8):2469–2493, March 2010.
[438] H. E. Campbell, M. P. Escudier, P. Patel, S. J. Challacombe, J. D. Sanderson, and M. C. E. Lomer. Review article: cinnamon- and benzoate-free diet as a primary treatment for orofacial granulomatosis. *Alimentary Pharmacology & Therapeutics*, 34(7):687–701, August 2011.
[439] Ana del Olmo, Javier Calzada, and Manuel Nuñez. Benzoic acid and its derivatives as naturally occurring compounds in foods and as additives: Uses, exposure, and controversy. *Critical Reviews in Food Science and Nutrition*, 57(14):3084–3103, November 2017.
[440] Lalita K. Gardner and Glen D. Lawrence. Benzene production from decarboxylation of benzoic acid in the presence of ascorbic acid and a transition-metal catalyst. *Journal of Agricultural and Food Chemistry*, 41(5):693–695, May 1993.
[441] Eugenio Aprea, Franco Biasioli, Silvia Carlin, Tilmann D. Märk, and Flavia Gasperi. Monitoring benzene formation from benzoate in model systems by proton transfer reaction-mass spectrometry. *International Journal of Mass Spectrometry*, 275(1-3):117–121, August 2008.
[442] Armand J. Quick. Clinical value of the test for hippuric acid in cases of disease of the liver. *Archives of Internal Medicine*, 57(3):544:556, March 1936.
[443] T Wilczok and G Bieniek. Urinary hippuric acid concentration after occupational exposure to toluene. *Occupational and Environmental Medicine*, 35(4):330–334, November 1978.
[444] John Ambre. The urinary excretion of cocaine and metabolites in humans: A kinetic analysis of published data. *Journal of Analytical Toxicology*, 9(6):241–245, November 1985.
[445] R. H. Williams, J. A. Maggiore, S. M. Shah, T. B. Erickson, and A. Negrusz. Cocaine and its major metabolites in plasma and urine samples from patients in an urban emergency medicine setting. *Journal of Analytical Toxicology*, 24(7):478–481, October 2000.
[446] Amin S. M. Salehi, Miriam J. Shakalli Tang, Mark T. Smith, Jeremy M. Hunt, Robert A. Law, David W. Wood, and Bradley C. Bundy. Cell-free protein synthesis approach to biosensing hTRβ-specific en-

docrine disruptors. *Analytical Chemistry*, 89(6):3395–3401, March 2017.
[447] Amin S.M. Salehi, Seung Ook Yang, Conner C. Earl, Miriam J. Shakalli Tang, J. Porter Hunt, Mark T. Smith, David W. Wood, and Bradley C. Bundy. Biosensing estrogenic endocrine disruptors in human blood and urine: A RAPID cell-free protein synthesis approach. *Toxicology and Applied Pharmacology*, 345:19–25, April 2018.
[448] A. W. Martinez, S. T. Phillips, and G. M. Whitesides. Three-dimensional microfluidic devices fabricated in layered paper and tape. *Proceedings of the National Academy of Sciences*, 105(50):19606–19611, December 2008.
[449] Filippo Caschera and Vincent Noireaux. Synthesis of 2.3 mg/ml of protein with an all escherichia coli cell-free transcription–translation system. *Biochimie*, 99:162–168, April 2014.
[450] Amin Espah Borujeni, Anirudh S. Channarasappa, and Howard M. Salis. Translation rate is controlled by coupled trade-offs between site accessibility, selective RNA unfolding and sliding at upstream standby sites. *Nucleic Acids Research*, 42(4):2646–2659, November 2014.
[451] Patrick P. Dennis and Hans Bremer. Modulation of Chemical Composition and Other Parameters of the Cell at Different Exponential Growth Rates. *EcoSal Plus*, 3(1), 2008.
[452] Gertjan Kramer, Richard R Sprenger, Merel A Nessen, Winfried Roseboom, Dave Speijer, Luitzen de Jong, M Joost Teixeira de Mattos, JaapWillem Back, and Chris G de Koster. Proteome-wide alterations in Escherichia coli translation rates upon anaerobiosis. *Molecular & cellular proteomics : MCP*, 9(11):2508–16, nov 2010.
[453] J. A. Bernstein, A. B. Khodursky, P.-H. Lin, S. Lin-Chao, and S. N. Cohen. Global analysis of mRNA decay and abundance in Escherichia coli at single-gene resolution using two-color fluorescent DNA microarrays. *Proceedings of the National Academy of Sciences*, 99(15):9697–9702, 2002.
[454] Sandra Placzek, Ida Schomburg, Antje Chang, Lisa Jeske, Marcus Ulbrich, Jana Tillack, and Dietmar Schomburg. BRENDA in 2017: new perspectives and new tools in BRENDA. *Nucleic Acids Research*, 45(D1):D380–D388, jan 2017.
[455] Amin Espah Borujeni, Anirudh S. Channarasappa, and Howard M. Salis. Translation rate is controlled by coupled trade-offs between site accessibility, selective RNA unfolding and sliding at upstream standby sites. *Nucleic Acids Research*, 42(4):2646–2659, feb 2014.
[456] RStudio Team. *RStudio: Integrated Development Environment for R*. RStudio, Inc., Boston, MA, 2016.
[457] Hadley Wickham. Reshaping data with the reshape package. *Journal of Statistical Software*, 21(12):1–20, 2007.
[458] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.
[459] David S Wishart, Yannick Djoumbou Feunang, Ana Marcu, An Chi Guo, Kevin Liang, Rosa Vázquez-Fresno, Tanvir Sajed, Daniel Johnson, Carin Li, Naama Karu, Zinat Sayeeda, Elvis Lo, Nazanin Assempour, Mark Berjanskii, Sandeep Singhal, David Arndt, Yonjie Liang, Hasan Badran, Jason Grant, Arnau Serra-Cayuela, Yifeng Liu, Rupa Mandal, Vanessa Neveu, Allison Pon, Craig Knox, Michael Wilson, Claudine Manach, and Augustin Scalbert. HMDB 4.0: the human metabolome database for 2018. *Nucleic Acids Res.*, 46(D1):D608–D617, January 2018.
[460] John Selberg, Marcella Gomez, and Marco Rolandi. The potential for convergence between synthetic biology and bioelectronics. *Cell Systems*, 7(3):231–244, September 2018.
[461] Evgeny Katz. Enzyme-based logic gates and networks with output signals analyzed by various methods. *ChemPhysChem*, 18(13):1688–1713, May 2017.
[462] Christina Kiel, Eva Yus, and Luis Serrano. Engineering signal transduction pathways. *Cell*, 140(1):33–47, January 2010.
[463] F. Farzadfard and T. K. Lu. Genomically encoded analog memory with precise in vivo DNA writing in living cell populations. *Science*, 346(6211):1256272, November 2014.
[464] J. Bonnet, P. Subsoontorn, and D. Endy. Rewritable digital data storage in live cells via engineered control of recombination directionality. *Proceedings of the National Academy of Sciences*, 109(23):8884–8889, May 2012.
[465] J. Bonnet, P. Yin, M. E. Ortiz, P. Subsoontorn, and D. Endy. Amplifying genetic logic gates. *Science*, 340(6132):599–603, March 2013.
[466] Yimeng Zeng, Alicia M. Jones, Emily E. Thomas, Barbara Nassif, Jonathan J. Silberg, and Laura Segatori. A split transcriptional repressor that links protein solubility to an orthogonal genetic circuit. *ACS Synthetic Biology*, page acssynbio.8b00129, aug 2018.
[467] Alexander A. Green, Pamela A. Silver, James J. Collins, and Peng Yin. Toehold switches: De-novo-designed regulators of gene expression. *Cell*, 159(4):925–939, November 2014.

[468] David Bikard, Wenyan Jiang, Poulami Samai, Ann Hochschild, Feng Zhang, and Luciano A. Marraffini. Programmable repression and activation of bacterial gene expression using an engineered CRISPR-cas system. *Nucleic Acids Research*, 41(15):7429–7437, June 2013.
[469] A. A. Nielsen and C. A. Voigt. Multi-input CRISPR/cas genetic circuits that interface host regulatory networks. *Molecular Systems Biology*, 10(11):763–763, November 2014.
[470] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
[471] Jessica G. Perez, Jessica C. Stark, and Michael C. Jewett. Cell-free synthetic biology: Engineering beyond the cell. *Cold Spring Harbor Perspectives in Biology*, 8(12):a023853, October 2016.
[472] Zoe Swank, Nadanai Laohakunakorn, and Sebastian J. Maerkl. Cell-free gene-regulatory network engineering with synthetic transcription factors. *Proceedings of the National Academy of Sciences*, 116(13):5892–5901, March 2019.
[473] C. E. Cowles, N. N. Nichols, and C. S. Harwood. BenR, a XylS homologue, regulates three different pathways of aromatic acid degradation in pseudomonas putida. *Journal of Bacteriology*, 182(22):6339–6346, November 2000.
[474] D. Nevozhay, R. M. Adams, K. F. Murphy, K. Josic, and G. Balazsi. Negative autoregulation linearizes the dose-response and suppresses the heterogeneity of gene expression. *Proceedings of the National Academy of Sciences*, 106(13):5123–5128, March 2009.
[475] Nathaniel Roquet and Timothy K. Lu. Digital and analog gene circuits for biotechnology. *Biotechnology Journal*, 9(5):597–608, February 2014.
[476] David K. Karig, Sukanya Iyer, Michael L. Simpson, and Mitchel J. Doktycz. Expression optimization and synthetic gene networks in cell-free systems. *Nucleic Acids Research*, 40(8):3763–3774, December 2012.
[477] Erich Michel and Kurt Wüthrich. Cell-free expression of disulfide-containing eukaryotic proteins for structural biology. *FEBS Journal*, 279(17):3176–3184, August 2012.
[478] I.-S. Oh, D.-M. Kim, T.-W. Kim, C.-G. Park, and C.-Y. Choi. Providing an oxidizing environment for the cell-free expression of disulfide-containing proteins by exhausting the reducing activity of escherichia coli s30 extract. *Biotechnology Progress*, 22(4):1225–1228, August 2006.
[479] Christopher Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.
[480] Simon O. Haykin. *Neural Networks and Learning Machines*. Pearson, 2009.
[481] A.K. Jain, Jianchang Mao, and K.M. Mohiuddin. Artificial neural networks: a tutorial. *Computer*, 29(3):31–44, March 1996.
[482] Ron Weiss, George E. Homsy, and Thomas F. Knight. Toward in vivo digital circuits. In *Natural Computing Series*, pages 275–295. Springer Berlin Heidelberg, 2002.
[483] Rahul Sarpeshkar. Analog versus digital: Extrapolating from electronics to neurobiology. *Neural Computation*, 10(7):1601–1638, October 1998.
[484] Daniel D. Lewis, Fernando D. Villarreal, Fan Wu, and Cheemeng Tan. Synthetic biology outside the cell: Linking computational tools to cell-free systems. *Frontiers in Bioengineering and Biotechnology*, 2, December 2014.
[485] Leonardo Noriega. *Multilayer perceptron tutorial*. School of Computing. Staffordshire University, 2005.
[486] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, N.J. : Prentice Hall, 1999.
[487] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, December 1989.
[488] Raul Rojas. *Neural Networks: A Systematic Introduction*. Springer Science and Business Media LLC, 2013.
[489] Kevin M. Cherry and Lulu Qian. Scaling up molecular pattern recognition with DNA-based winner-take-all neural networks. *Nature*, 559(7714):370–376, July 2018.
[490] Michael Jahn, Carsten Vorpahl, Thomas Hübschmann, Hauke Harms, and Susann Müller. Copy number variability of expression plasmids determined by cell sorting and droplet digital PCR. *Microbial Cell Factories*, 15(1):211, December 2016.

Modélisation pour la conception et l'analyse de circuits métaboliques synthétiques.

Mots clés : modélisation, bio-informatique, rétrosynthèse, biosenseurs, biologie synthétique.

Résumé : Les buts de cette thèse sont doubles, et concernent les circuits métaboliques synthétiques, qui permettent de détecter des composants chimiques par transmission de signal et de faire du calcul en utilisant des enzymes.
La première partie a consisté à développer des outils d'apprentissage actif et par renforcement pour améliorer la conception de circuits métaboliques et optimiser la biodétection et la bioproduction. Pour atteindre cet objectif, un nouvel algorithme (RetroPath3.0) fondé sur une recherche arborescente de Monte Carlo guidée par similarité est présenté. Cet algorithme, combiné à des règles de réaction apprises sur des données et des niveaux différents de promiscuité enzymatique, permet de focaliser l'exploration sur les composés et les chemins les plus prometteurs en bio-rétrosynthèse. Les chemins obtenus par rétrosynthèse peuvent être implémentés dans des cellules ou des systèmes acellulaires. Afin de concevoir le meilleur milieu pour optimiser la productivité du système, une méthode d'apprentissage actif qui explore efficacement l'espace combinatoire des composants du milieu a été développée.
La deuxième partie a consisté à développer des méthodes d'analyse, pour générer des connaissances à partir de données biologiques, et modéliser les réponses de biocapteurs. Dans un premier temps, l'effet du nombre de copies de plasmides sur la sensibilité d'un biocapteur utilisant un facteur de transcription a été modélisé. Ensuite, en utilisant des systèmes acellulaires qui permettent un meilleur contrôle des variables expérimentales comme la concentration d'ADN, l'utilisation des ressources a été modélisée pour assurer que notre compréhension actuelle des phénomènes sous-jacents est suffisante pour rendre compte du comportement du circuit, en utilisant des modèles empiriques ou mécanistiques. Couplés aux outils de conception de circuits métaboliques, ces modèles ont ensuite permis de développer une nouvelle approche de calcul biologique, appelée perceptrons métaboliques.
Dans l'ensemble, cette thèse présente des outils de conception et d'analyse pour les circuits métaboliques synthétiques. Ces outils ont été utilisés pour développer une nouvelle méthode permettant d'effectuer des calculs en biologie synthétique.

Computational modeling to design and analyze synthetic metabolic circuits.

Keywords: mathematical modeling, bioinformatics, retrosynthesis, biosensors, synthetic biology.

Abstract: The aims of this thesis are two-fold, and centered on synthetic metabolic circuits, which perform sensing and computation using enzymes.
The first part consisted in developing reinforcement and active learning tools to improve the design of metabolic circuits and optimize biosensing and bioproduction. In order to do this, a novel algorithm (RetroPath3.0) based on similarity-guided Monte Carlo Tree Search to improve the exploration of the search space is presented. This algorithm, combined with data-derived reaction rules and varying levels of enzyme promiscuity, allows to focus exploration toward the most promising compounds and pathways for bio-retrosynthesis. As retrosynthesis-based pathways can be implemented in whole cell or cell-free systems, an active learning method to efficiently explore the combinatorial space of components for rational buffer optimization was also developed, to design the best buffer maximizing cell-free productivity.
The second part consisted in developing analysis tools, to generate knowledge from biological data and model biosensor response. First, the effect of plasmid copy number on sensitivity of a transcription-factor based biosensor was modeled. Then, using cell-free systems allowing for broader control over the experimental factors such as DNA concentration, resource usage was modeled to ensure our current knowledge of underlying phenomenons is sufficient to account for circuit behavior, using either empirical models or mechanistic models. Coupled with metabolic circuit design, those models allowed us to develop a new biocomputation approach, called metabolic perceptrons.
Overall, this thesis presents tools to design and analyse synthetic metabolic circuits, which are a novel way to perform computation in synthetic biology.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France