Modélisation de séquences et reconstruction non supervisée de génomes microbiens à partir de donnée métagénomiques.

Kévin Gravouil

UNIVERSITÉ Clermont Auvergne

ÉCOLE DOCTORALE DES SCIENCES DE LA VIE,
SANTÉ, AGRONOMIE ET ENVIRONNEMENT

Numéro d'ordre XXX

THÈSE
Présentée à l'Université Clermont Auvergne
pour l'obtention du grade de

DOCTEUR D'UNIVERSITÉ
Spécialité : Bioinformatique

Présentée et soutenue publiquement le 20/12/2019 par
KÉVIN GRAVOUIL

MODÉLISATION DE SÉQUENCES ET RECONSTRUCTION NON SUPERVISÉE DE GÉNOMES MICROBIENS À PARTIR DE DONNÉES MÉTAGÉNOMIQUES

Composition du jury :

Christine GASPIN Directrice de Recherche	Rapportrice	UR 875 MIAT INRA Toulouse
Pierre PETERLONGO Chargé de Recherche	Rapporteur	GenScale INRIA Rennes
Timothy VOGEL Professeur des Universités	Rapporteur	UMR 5005 AMPÈRE Université de Lyon
Engelbert MEPHU NGUIFO Professeur des Universités	Examinateur	UMR 6158 LIMOS Université Clermont Auvergne
Éric PEYRETAILLADE Maître de Conférences Universitaire	Co-directeur de thèse	UMR 6023 LMGE Université Clermont Auvergne
Didier DEBROAS Professeur des Universités	Co-directeur de thèse	UMR 6023 LMGE Université Clermont Auvergne
Marie PAILLOUX Maître de Conférences Universitaire	Co-encadrante de thèse	UMR 6158 LIMOS Université Clermont Auvergne

CONSEIL RÉGIONAL AUVERGNE

MEDIS Microbiologie, Environnement Digestif et Santé

LIMOS

Laboratoire Microorganismes : Génome et Environnement

Sommaire

Remerciements

Quoi de mieux pour commencer la lecture de ce manuscrit que la mention de toutes les personnes qui, par leur travail passionné, leurs conseils ou leur simple existence, ont pu rendre possible l'aboutissement de cette thèse ?

Je tiens tout d'abord à remercier les différents laboratoires qui m'ont accueilli. Merci à Monique Alric et Pierre Peyret de l'UMR 454 MEDIS (anciennement EA 4678 CIDAM), merci à Farouk Toumani de l'UMR 6158 LIMOS, et merci à Télesphore Sime-Ngando de l'UMR 6023 LMGE pour votre accueil et pour m'avoir donné les moyens de réaliser cette thèse dans de bonnes conditions.

J'exprime également mes plus sincères remerciements à Christine Gaspin, Pierre Peterlongo et Timothy Vogel pour m'avoir fait l'honneur d'accepter d'être rapporteurs de cette thèse ainsi qu'à Engelbert Mephu Nguifo pour avoir accepté d'examiner mon travail et faire partie de mon jury de thèse. Je remercie Sébastien Terrat et Philippe Leroy d'avoir accepté de participer aux différents comités de suivi de ma thèse, de votre disponibilité et de vos retours.

Ce travail n'aurait pas été possible sans la participation de la région Auvergne au travers du Contrat de plan État-Région et de l'Université Clermont-Auvergne, notamment au travers de l'École Doctorale des Sciences de la Vie, Santé, Agronomie, Environnement, et du Mésocentre.

Mes remerciements vont naturellement à mes encadrant-e-s de thèse, Éric Peyretaillade, Didier Debroas et Marie Pailloux pour m'avoir guidé et dirigé tout au long de ces trois années de travail par leur discussion et leur ouverture d'esprit. Travailler avec des personnes aux points de vue différents fut un réel apport dans ce travail et une nécessité pour moi. Merci également à vous pour m'avoir tant apporté sur tous les autres aspects qui rythment la vie d'un chercheur. Et surtout, merci à vous pour votre patience, merci pour m'avoir permis d'explorer des tas de choses – pas toujours fructueuses, mais quelquefois indispensables alors que rien ne le laissait présager – et un immense merci de m'avoir fait confiance.

Merci également aux différentes personnes qui sont intervenues durant mes travaux et qui m'ont ouvert à de nouveaux domaines. Merci à Violaine Antoine pour ses discussions et ressources sur le clustering et merci à Vincent Barra de m'avoir fait mettre un pied dans les réseaux de neurones.

De la même manière, merci aux personnes qui m'ont fait confiance pour intervenir dans la formation des étudiants. Merci à Brigitte Ekpe, à Valérie Polonais et à Éric Peyretaillade pour m'avoir permis de m'essayer à l'enseignement à des étudiants aux appétences variées.

Merci également aux personnes qui rendent la recherche plus ouverte et donc plus belle.

La suite de mes remerciements s'adressent aux membres des différentes équipes que j'ai pu rejoindre. Merci donc à tous les membres de MEDIS, du LIMOS et du LMGE qui permettent de travailler dans d'excellentes conditions aussi bien scientifiques, professionnelles et personnelles.

Merci Clémence Defois, Réjane Beugnot, Cédric Bernarde, Jean-François Brugère, Sandrine Chalancon, Sylvain Denis, Sophie Marre et Manon Martinet pour votre accueil à MEDIS.

Merci Gisèle Bronner, Jean-Christophe Charvy, Marina Chauvet, Nelly Cruvellier, Maxime Fuster, Claire Hennequin, Isabelle Jouan, Cécile Lepère, Anne Mone, Corinne Petit, Viviane Ravet, Agnès Vellet pour votre accueil, ces quelques pauses cafés et repas toujours animés de bonne humeur. J'aurais aimé passer davantage de temps avec vous, mais les multiples déménagements en ont décidé autrement.

Merci enfin à Benjamin Vincent, Nicolas Champeil, Boris Lonjon, Amina Chorfi, Angeline Plaud, Carlos Cepeda, Rafael Colares, Benjamin Dalmas, Matthieu Gondran, Henri Perret du Cray, Béatrice Bourdieu, Martine Cacciopoli et toutes les personnes du LIMOS.

Une mention spéciale est faite à mes co-bureaux Alexis Pereda, David Brevet et Benjamin Bergougnoux. Merci les gars pour vos interminables ratiocinations des sujets pas du tout dans mes cordes. J'ai appris et j'ai compris plein de trucs à cause de vous, bon gré mal gré pour arriver sur la fin. À l'image de ces joyeux bonobos (presque).

Et c'est une transition toute trouvée pour se tourner vers l'allègre bande de bioinfo qui a toujours été présente, aussi bien pour échanger sur nos sujets de recherche respectifs, pour de folles parties de jeux de société ou pour aller explorer la gastronomie locale (et souvent tout ça à la fois !). Merci donc à Bérénice Batut, Corentin Hochart, Tristan Dubos, Hélène Gardon, Romain De Oliveira, Jeanne Bazile et Cécile Hilpert.

Merci aux organisateurs et participants des JeBif Pubs de faire vivre la communauté bioinfo à Clermont. Merci Camille, Anaïs, Jordan, Loïc, Émilie, Marie, Cléa, Medhi, Nadia, Aurore, Anne-Lise et tous les autres que j'ai pu croiser.

Je ne peux oublier de remercier mes camarades de longue date – et malheureusement de longues distances – pour m'avoir aider plus ou moins directement, plus ou moins consciemment mais toujours. Always. Merci donc aux Copains d'Abord *extended edition* : Tuck, Marielle, Piotr, Camille, Pierre, Armelle, Guigui, Bichon, Mag, Émi, Fred, Anaïs, Benji et Adrien. Un immense merci également à mes frères, Yoann et Bastien, à mes parents et à mes grands parents.

Et enfin – et surtout ! – qu'aurais-je pu faire sans elle ? Ça n'aurait en tout cas pas été la même limonade. Un incommensurable merci Charlotte de ne pas m'avoir étrangler après tout ce temps, de m'accompagner et de me pousser à être quelqu'un de meilleur. Merci.

Ce fut une aventure qui n'aurait pu être sans toutes ces personnes. Merci encore et bonne lecture.

Résumé

Les micro-organismes sont ubiquistes et contribuent à tous les cycles biogéochimiques de la planète. Leur étude en laboratoire souffre cependant d'importantes limitations. Les approches métagénomiques permettent d'étudier la matière noire microbienne *in situ*.

Leur étude en laboratoire souffre cependant d'importantes limitations et les approches de type "omique" comme la métagénomique ont ainsi révolutionné l'écologie microbienne. Il est ainsi possible de reconstruire des génomes sans mettre en culture les microorganismes grâce aux recours à des stratégies par des approches de *binning* non supervisé.

Les méthodes existantes sont articulées en deux étapes : une représentation numérique (ou « modélisation ») des séquences métagénomiques (le plus souvent, des contigs) puis un clustering. Ce dernier produit des « bins » représentant les génomes.

Une comparaison objective de ces méthodes restait nécessaire. Cette dernière a été conduite à partir de jeux de données maîtrisés et a permis de mettre en évidence une influence de la méthode employée sur les résultats, et ce quelle que soit l'origine taxonomique des micro-organismes reconstruits (chapitre 3).

Cependant, la modélisation de séquences manque de diversité pour tirer parti des approches consensuelles pourtant prometteuses. Six modélisations, dont trois encore inutilisées pour le binning et une originale, sont réunies dans un même module Python dédié – `fennec`. Ces différentes modélisations sont intégrées en une seule de manière non supervisée et adaptative. Cette adaptabilité a été exploitée au travers d'un processus itératif d'extraction de génomes automatique basé sur un clustering semi-supervisé (chapitre 4).

Cet outil appliqué individuellement ne permet pas la reconstruction de meilleurs génomes sur des données de test et réelles. Cependant, il vient améliorer les résultats d'autres logiciels lorqu'une approche par consensus est envisagée (chapitre 5).

Mots clés bioinformatique, métagénomique, reconstruction de génomes non supervisée, modélisation de séquences, extraction de clusters.

1 | État de l'art

1.1 Étude du monde microbien

Les procaryotes, avec une abondance totale estimée de 4 à 6×10^{30} cellules (Whitman *et al.*, 1998), représentent les formes de vie cellulaire les plus abondantes et les plus diverses sur Terre. On estime par exemple que près de 10^6 espèces distinctes de bactéries et d'archées peuvent être présentes dans un seul gramme de sol (Gasc *et al.*, 2015).

Les micro-organismes sont ubiquitaires et participent à la régulation de tous les cycles biogéochimiques de la planète (Cronan, 2018) mais également à la santé humaine (Jandhyala, 2015; Liang *et al.*, 2018) ou animale (*e. g.* : ruminants (Malmuthuge et Guan, 2017), abeilles (Kwong et Moran, 2016)). Ils représentent donc d'importants leviers pour agir sur la qualité de productions agro-alimentaires (Yáñez-Ruiz *et al.*, 2015), la bioremédiation d'environnements (Terrat, 2010; Defois *et al.*, 2017) ou l'amélioration de la santé humaine (Musso *et al.*, 2010; Qin *et al.*, 2012; Shreiner *et al.*, 2015).

Les micro-organismes cultivables peuvent être caractérisés par l'étude de leur matériel génétique : on parlera alors de **génomique**. Il est en effet possible, à partir du génome d'un micro-organisme pouvant être jusqu'alors inconnu, de l'identifier taxonomiquement, de prédire et d'annoter ses gènes, et donc de déduire ses potentialités métaboliques. Les approches génomiques se sont donc progressivement imposées comme des méthodes incontournables pour l'étude globale d'un organisme. Des outils dédiés comme l'*Integrated Microbial Genomes* (IMG) (Markowitz *et al.*, 2012; Huntemann *et al.*, 2015), Reactome (Croft *et al.*, 2011), MetaCyc (Caspi *et al.*, 2018) ou KEGG (Kanehisa *et al.*, 2017) permettent la prédiction de structures génomiques (*e. g.* : séquences codantes pour des protéines, ARN ribosomaux, ARN de transfert) auxquelles des fonctions biologiques sont attribuées par comparaison à des bases de données de référence (*e. g.* : COG, Pfam). Ces annotations structurales et fonctionnelles aident ainsi à la compréhension du fonctionnement de l'organisme (*e. g.* : hétérotrophie, pathogénicité). De plus, plusieurs génomes peuvent être comparés entre eux, permettant par exemple de reconstruire leur histoire évolutive ou pour identifier des marqueurs d'intérêt biotechnologique et/ou clinique.

1.2 Approches en génomique environnementale

L'étude ciblée d'un micro-organisme par le séquençage de son génome requiert une culture clonale de celui-ci ou tout du moins son enrichissement de façon à en

extraire son matériel génétique. Cependant, on estime que moins de 1 % des espèces bactériennes ont été cultivées de manière axénique (Lloyd *et al.*, 2018). Les microbiologistes sont alors confrontés à un problème similaire à celui des astrophysiciens : la matière noire non observable (Gasc *et al.*, 2015; Ribière *et al.*, 2016). Différentes approches de **génomique environnementale** ont donc été développées pour permettre l'exploration de cette matière noire microbienne par l'étude de l'ensemble des génomes microbiens d'un écosystème. Ainsi, on n'accède plus au génome d'une seule souche microbienne isolée et cultivée en condition axénique mais au matériel génétique de l'intégralité des organismes en présence dans leur milieu.

La génomique environnementale se base sur l'utilisation d'outils moléculaires, notamment le séquençage à haut débit pour assurer l'identification des micro-organismes directement issus de l'environnement. Le matériel génétique est « lu » à l'aide d'une technologie de séquençage (voir Annexe A.1). Les séquences obtenues sont appelées « lectures de séquençage » et représentent entre quelques dizaines et quelques milliers de nucléotides consécutifs selon les technologies. L'appareil de séquençage mesure également la qualité de lecture pour chaque nucléotide (voir Annexe A.1.4). Ces lectures pourront alors faire l'objet de divers traitements, notamment un assemblage (voir sous-section 1.3.1).

La **métagénomique**, visant à caractériser l'ensemble du matériel génétique du microbiome comme une seule entité, est donc une des approches possibles de la génomique environnementale. Cette dernière va donc viser à étudier le microbiote à **l'échelle des génomes ou des gènes** ou à **l'échelle de la population**. Dans le premier cas il s'agit de caractériser les individus (idéalement, les génomes) de la population, soit pour rechercher un gène d'intérêt, soit en vue d'étudier leurs interactions. Dans le second cas, l'objectif est de mettre en lumière la communauté en décrivant sa structure et/ou ses fonctions biologiques. Conformément aux définitions proposées (Marchesi et Ravel, 2015), nous qualifierons de :

- « microbiote » l'ensemble des micro-organismes vivant dans un environnement spécifique ;
- « microbiome » l'ensemble des facteurs biotiques et abiotiques d'un écosystème, comprenant donc son microbiote ainsi que ses conditions environnementales ;
- « métagénomique » l'étude à haut-débit de l'intégralité du matériel génétique, qui forme ainsi le métagénome d'un microbiote ;
- « métabarcoding » l'étude ciblée, qualitative ou quantitative et à haut débit d'un marqueur génétique au sein de l'environnement en vue de décrire et/ou

d'explorer une fonctionnalité biologique du microbiome ;

– « métataxonomique » l'étude qualitative ou quantitative et à haut débit d'un gène permettant la description structurelle du microbiote en ciblant spécifiquement des marqueurs taxonomiques (contrairement au métabarcoding qui peut cibler n'importe quel gène).

Il existe d'autres approches que l'étude d'un microbiote par le prisme de l'ADN comme la métabolomique (analyse à haut débit des métabolites d'un microbiote), la métatranscriptomique (analyse à haut débit des ARN d'un microbiote) ou encore la métaprotéomique (analyse à haut débit des protéines d'un microbiote).

La reconstruction du lien qui existe nécessairement entre la structure d'un microbiote et ses fonctions métaboliques permettrait une compréhension bien plus grande des interactions des micro-organismes entre eux, mais aussi avec leur environnement. Lier la structure et les fonctions d'un microbiote devient alors indispensable pour appréhender de manière globale le fonctionnement d'un écosystème complexe. Or, ce lien structure-fonctions ne peut être résolu qu'en permettant l'exploration à différentes échelles d'un métagénome, allant de l'échelle des génomes individuels à l'échelle de la population globale.

L'accès aux génomes complets des organismes est donc indispensable pour comprendre leur physiologie. Cela est notamment rendu possible par le séquençage des métagénomes qui ne cesse d'être plus accessible et plus performant.

Différentes approches, notamment basées sur la réduction de complexité du métagénome, sont décrites en Annexe A.2. Celles-ci sont cependant limitées dans la vitesse d'obtention des données (besoin de dispositions expérimentales spécifiques) et limitent donc la réutilisation à grande échelle des données (notamment dans le cadre des études croisées).

1.3 Exploration *in situ* sans *a priori* de la diversité microbienne

La répartition des abondances des espèces qui composent un microbiote tend à suivre une loi de puissance très souvent représentée par un diagramme de Whittaker – ou courbe de rang-abondance (Izsák et Pavoine, 2012). On distingue alors quelques espèces majoritaires et de nombreuses espèces minoritaires, ces dernières constituant la biosphère rare (Zhou *et al.*, 2013; Lynch et Neufeld, 2015). Une espèce est considérée

comme rare lorsque son abondance relative est inférieure à 0,1 % ou 0,01 % selon les auteurs (Pedrós-Alió, 2012; Lynch et Neufeld, 2015).

Les techniques de séquençage rendent difficile le séquençage de cette biosphère rare, à moins d'envisager des profondeurs de séquençage (le nombre moyen de fois où chaque nucléotide est séquencé) extrêmement importantes. De plus, le nombre d'espèces présentes dans un environnement peut se limiter à moins d'une dizaine (Tyson *et al.*, 2004; De Filippis *et al.*, 2016) mais peut atteindre plusieurs dizaines de millions (*e. g.* : sols, milieux aquatiques) notamment dans les milieux extrêmes (*e. g.* : lac Vostok, cheminées hydrothermales des abysses) (Rogers *et al.*, 2013; Cowan *et al.*, 2015; Sime-Ngando *et al.*, 2016; Lloyd *et al.*, 2018). Outre la disparité des abondances des espèces, la complexité d'un métagénome réside également dans la variabilité des tailles des génomes, les transferts horizontaux de gènes, la présence de plasmides ou encore la polyploïdie de certains organismes.

Les approches génomiques ciblées limitent par définition le caractère exploratoire et nécessitent la mise en place d'expérimentations spécifiques pour générer les données appropriées (*e. g.* : marqueurs). À l'inverse, les approches métagénomiques peuvent donc nécessiter des profondeurs de séquençages importantes et coûteuses afin d'atteindre la biosphère rare mais présentent les avantages d'être au plus proche des conditions environnementales L'absence d'*a priori* permet ainsi la ré-utilisation de données publiques et donc d'atteindre des profondeurs de séquençage importante par méta-analyse (*e. g.* : Fierer *et al.* (2012); Parks *et al.* (2017)).

1.3.1 Assemblage de lectures de séquençage

Le séquençage d'un matériel génétique est aléatoire et techniquement limité à quelques centaines de nucléotides par lecture alors que les génomes microbiens ont une taille de quelques millions, voire milliards, de nucléotides. L'assemblage sans génome de référence, dit *de novo,* des lectures en séquences contigües – ou contigs – est donc nécessaire pour l'étude exploratoire des génomes à partir de données métagénomiques.

Le principe de l'assemblage *de novo* est le suivant :

1. les chevauchements entre les lectures sont représentés par un graphe
2. le graphe est parcouru pour reconstruire les contigs.

1.3.1.1 Description des algorithmes principaux

L'assemblage *de novo* est principalement basé sur deux algorithmes : Overlap-Layout-Consensus (OLC) et les graphes de De Bruijn (DBG) (Li *et al.*, 2012) qui sont présentés ci-après. Les méthodes d'assemblage basées sur l'alignement des lectures sur des génomes de référence (*e. g.* : Lischer et Shimizu (2017)) ne seront pas abordées.

L'algorithme **OLC** représente le graphe de chevauchement grâce à l'alignement des paires de séquences entre elles. L'identification des chevauchements est donc explicite et permet un assemblage optimal. Cependant, ces aligments sont potentiellement couteux en ressources et donc inapplicables à de trop grands volumes de données. On notera l'existence d'une variante de cet algorithme utilisant un *string graph* (Myers, 2005), obtenu après suppression des liens transitifs dans le graphe de chevauchements, accélérant alors son parcours.

L'algorithme **DBG** quant à lui découpe les lectures en sous-séquences de taille k (aussi appelées k-mers) pour représenter les chevauchements. Avec k suffisament grand (*e. g.* : $k > 31$), chaque k-mer devient probablement unique. Ainsi, la présence d'un même k-mer dans plusieurs lectures indique probablement un chevauchement. De plus, chaque k-mer n'est stocké qu'une seule fois, évitant ainsi une explosion du volume de données à manipuler. Conceptuellement, chaque noeud du graphe de chevauchements représente un k-mer et est lié *via* une arête aux noeuds ayant un chevauchement de taille $k-1$. La construction des contigs prend alors la forme d'une recherche de chemins eulériens dans le graphe de chevauchements (Pevzner *et al.*, 2001). Cependant, cette méthode identifie les chevauchements entre les lectures de manière implicite et peut induire la création de contigs chimériques si la taille de k-mers choisie ne permet pas d'assurer leur unicité.

1.3.1.2 Outils complémentaires et améliorations

Plusieurs outils et stratégies complémentaires ont été développés pour réduire les besoins en ressources de ces méthodes et améliorer leur fiabilité. Une première amélioration consiste à indexer le graphe, notamment avec des structures de données probabilistes comme les filtres de Bloom ou l'index de Ferragina-Manzini (basé sur la transformée de Burrow-Wheeler). Ces structures de données permettent de tester très efficacement en terme de temps et de mémoire consommées la présence d'un k-mer donné dans le graphe au prix d'une marge d'erreur maitrisée.

D'autre part, le fait que le séquençage de l'ADN soit aléatoire ne permet pas de déterminer le brin duquel provient la lecture. Ainsi, on représente les k-mers

par des k-mer canoniques : chaque k-mer et son équivalent complémenté et inversé sont comptabilisés ensemble, permettant ainsi de réduire le nombre d'informations à stocker.

Enfin, la distance en nombre de nucléotides entre les deux lectures peut être connue grâce au séquençage de type *paired-end* (voir Annexe A.1.4). Ainsi, le parcours du graphe de chevauchement peut être guidé par cette information pour confirmer ou infirmer certains chemins. Ces algorithmes sont également fortement impactés par les erreurs de séquençage, bien que diverses stratégies de correction de ces erreurs soient mises en place.

D'autres méthodes hybrides, comme MaSuRCa (Zimin *et al.*, 2013) ou StriDe (Huang et Liao, 2016), tentent d'exploiter les avantages des deux algorithmes précédemment décrits. **MaSuRCa** va créer des « supers lectures » à l'aide d'un graphe de De Bruijn des lectures, puis ces « supers lectures » seules seront assemblées grâce à l'algorithme OLC tel qu'implémenté dans l'assembleur CABOG (Miller *et al.*, 2008). **StriDe** découpe les lectures en fonction des scores de qualité des lectures plutôt que d'utiliser une taille fixe. Ainsi, une portion de séquence considérée comme de faible qualité (avec une forte probabilité de lecture erronée donc) sera découpée en petits k-mers tandis qu'une portion perçue comme fiable sera découpée en plus grands k-mers. À notre connaissance, StriDe n'a pas été utilisé dans le cadre de l'assemblage de métagénomes mais permettrait l'exploitation de données provenant de différentes plateformes de séquençage en même temps.

1.3.1.3 Évaluation des logiciels d'assemblage

De nombreuses évaluations des méthodes d'assemblage *de novo* existent (Salzberg *et al.*, 2011; Bradnam *et al.*, 2013; Magoc *et al.*, 2013; Ghurye *et al.*, 2016; Sohn et Nam, 2016; Vollmers *et al.*, 2017; Forouzan *et al.*, 2018) afin de connaître, voire d'anticiper, les problèmes provoqués par un assemblage. Les logiciels d'assemblage les plus réputés sont IDBA-UD (Peng *et al.*, 2012), MegaHIT (Li *et al.*, 2016), Meta-Velvet (Namiki *et al.*, 2012), Ray Meta (Boisvert *et al.*, 2012) et SOAPdenovo2 (Luo *et al.*, 2012).

1.3.2 Méthodes de regroupement de contigs, ou « binning »

L'assemblage des lectures de séquençage requiert des chevauchements entre les séquences, ce qui ne peut être garanti du fait de la rareté de certains génomes et de l'inégale couverture d'un même génome (Ekblom *et al.*, 2014). Le co-assemblage

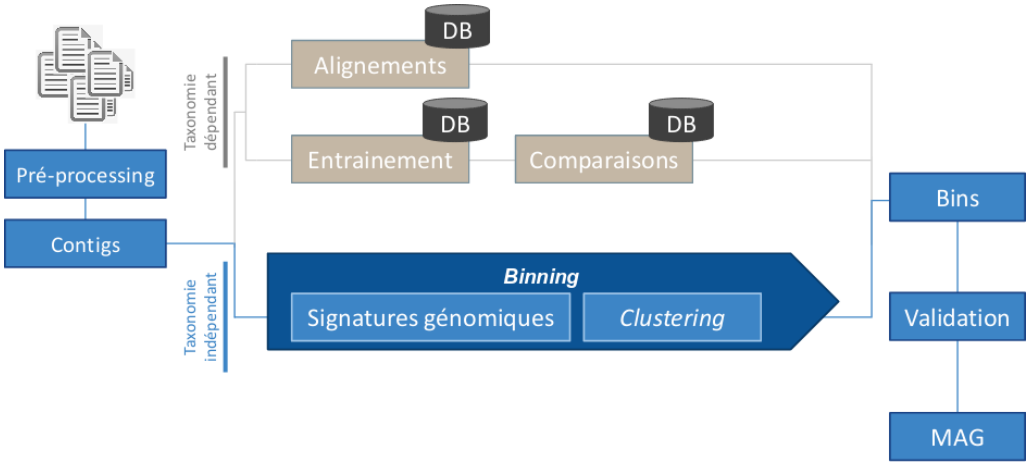


Figure 1.1 | **Workflow du binning de séquences**. En gris, le binning est supervisé, qu'il soit le résultat de l'alignement des contigs à des références ou de la comparaison de représentations des contigs entre eux. En bleu, le binning est non supervisé : il nécessite la représentation des contigs par des signatures génomiques intrinsèques qui serviront pour le clustering. Les bins ainsi produits peuvent alors être validés (plus de détails à la sous-section 1.4.2) puis considérés comme des *Metagenome-Assembled Genome* (MAG).

de multiples jeux de données (van der Walt *et al.*, 2017) permet d'augmenter la profondeur de séquençage mais reste difficilement envisageable du fait des limites des algorithmes et structures de données actuels. De ce fait, la reconstruction de génomes à partir de métagénomes ne peut se reposer sur l'assemblage seul.

Les contigs peuvent néanmoins être regroupés entre eux pour représenter des génomes fragmentés étudiables comme un génome individuel. Ce regroupement – ou « binning » (figure 1.1) – consiste à regrouper des contigs :

- de manière supervisée : les contigs sont comparés à des séquences de référence préalablement annotées ;
- de manière non supervisée : les contigs sont comparés entre eux sans information extérieure.

Le binning supervisé peut prendre la forme d'un alignement de séquence entre un contig inconnu et une base de données de séquences (Altschul *et al.*, 1990; Segata *et al.*, 2012). Les contigs peuvent d'autre part être représentés numériquement pour rendre leur comparaison plus efficace (Wood et Salzberg, 2014; Zielezinski *et al.*, 2017). Par exemple, le dénombrement des k-mers composant un contig peut être utilisé pour le représenter et être comparé à d'autres contigs. Cependant, une séquence qui ne trouve pas d'équivalent dans la base de connaissances restera orpheline. L'utilisation

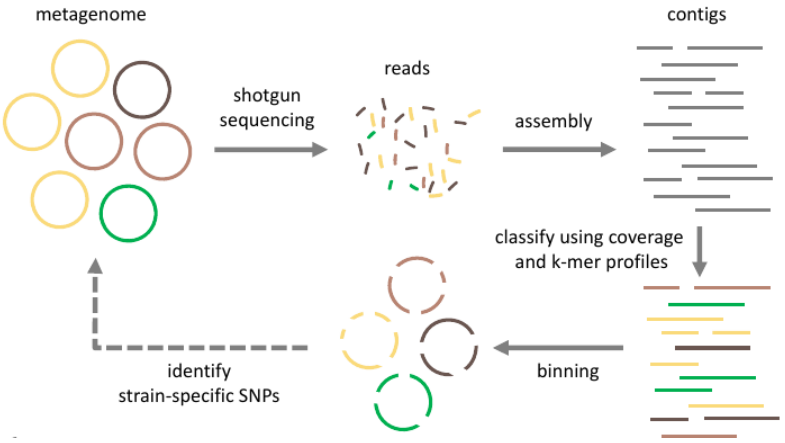


Figure 1.2 | **Vue d'ensemble de la méthode hybride**.

de connaissances *a priori* limite par définition le caractère exploratoire de l'approche métagénomique et ne sera donc pas développée ici.

Le binning non supervisé (ou « *reference-free* ») cherche à regrouper des séquences par des méthodes de partionnement de données, ou « clustering », en fonction de leur composition (Teeling *et al.*, 2004), de leur abondance moyenne au sein du métagénome (Liao *et al.*, 2014) ou d'une approche hybride (Mande *et al.* (2012); Sangwan *et al.* (2016); Sedlar *et al.* (2017), figure 1.2). Le binning non supervisé de séquences métagénomiques nécessite donc trois étapes : (i) la représentation numérique – ou modélisation – des contigs, (ii) l'intégration de ses différentes modélisations en une unique modélisations et (iii) le clustering de ces modélisations intégrées des séquences (contigs ou lectures selon les modélistions).

Les clusters ainsi proposés pourront être validés automatiquement ou manuellement pour devenir des « bins ». Ces bins peuvent aussi bien représenter un génome individuel que représentatif d'un groupe taxonomique de plus haut niveau (*e. g.* : genre, famille). Ce bin peut alors être étudié comme un génome fragmenté comme évoqué précédemment.

1.3.3 Exemples d'applications du binning non supervisé

La technique de séquençage de Sanger, dite de première génération, a permis la reconstruction par binning non supervisé de plusieurs génomes microbiens. Le premier exemple de binning et le plus simple à notre connaissance est celui appliqué au biofilm de surface des eaux drainées d'une mine acide localisée à Iron Mountain en Californie (Tyson *et al.*, 2004). Ce microbiome est soumis à des conditions extrêmes

(pH 0,83, 42°C, 317 mM de fer, 14 mM de zinc, 4 mM de cuivre et 2 mM arsenic), limitant très fortement sa diversité. Après séquençage de 76 millions de nucléotides et assemblage du métagénome, les contigs ont été discriminés en fonction de leur composition en G+C et de leur abondance moyenne. Ainsi, Tyson *et al.* (2004) ont pu reconstruire deux génomes quasiment complets et trois partiels, permettant alors l'exploration de la structure et des fonctions d'un microbiome à l'échelle des génomes individuels directement depuis leur environnement.

La deuxième génération des technologies de séquençage a permis une évolution importante de la reconstruction de génomes microbiens. Par exemple, les eaux saumâtres de la Mer Baltique ont été étudiées par une approche de métagénomique par Hugerth *et al.* (2015). 6,2 milliards de nucléotides ont été séquencés par la technologie Illumina puis assemblés. Les contigs ont ensuite été soumis au logiciel de binning CONCOCT (Alneberg *et al.*, 2014). La recherche de gènes en simple copie (SCG) a permis d'évaluer le taux de contamination des bins et valider 83 génomes assemblés à partir du métagenome (MAG). Ceux-ci ont alors été annotés taxonomiquement et fonctionnellement, permettant la prédiction des voies métaboliques. Ainsi, il a été possible de décrire la dynamique temporelle et la biogéographie de plusieurs espèces bactériennes jusqu'alors inconnues, et d'ouvrir des pistes sur leurs histoires évolutives.

C'est également par une approche de binning que Evans *et al.* (2015) ont pu décrire un phylum archéen méthanogénique et méthanotrophique (Bathyarchaeota), directement depuis l'eau de formation de puits de méthane de houille (Queensland, Australie). 9 milliards de nucléotides ont été obtenus par la technologie Illumina, assemblés puis regroupés à l'aide des logiciels MetaBAT (Kang *et al.*, 2015) et DBB (Parks, 2015). Les MAG correspondant aux Bathyarchaeota ont alors été rassemblés pour obtenir un génome représentatif de ce phylum. Cela a conduit à la découverte de gènes impliqués dans le métabolisme du méthane divergeant des gènes connus (*e. g.* : complexe méthyl-coenzyme M réductase) permettant ainsi de préciser l'apparition de ce métabolisme avant l'ancêtre commun aux Euryarchaeota et aux Bathyarchaeota, tous deux des contributeurs majeurs au cycle du méthane.

La création de consortia internationaux permet de mutualiser les efforts de séquençage pour des études à très grande échelle. Cette mutualisation devient nécessaire afin d'envisager une profondeur de séquençage suffisamment importante pour une étude approfondie de l'écosystème à étudier. Le Projet Microbiote Humain (*Human Microbiome Project*, HMP) (Turnbaugh *et al.*, 2007) est un projet de caractérisation des différents microbiotes associés à l'être humain. Ainsi, les approches

de métagénomique globale ont permis le séquençage de 4900 milliards de bases. Un catalogue de 2000 génomes microbiens issus des microbiotes humains a alors été établi (Nelson *et al.*, 2010).

L'expédition Tara Océans (Sunagawa *et al.*, 2015) permet d'accéder à un total de 7200 milliards de bases, et offre donc une profondeur de séquençage inédite pour les environnements marins.

Les bases de données internationales permettent l'accès à une quantité colossale de données de séquençage. Il devient alors concevable, grâce à l'effort conjoint de séquençage de multiples métagénomes, d'atteindre une profondeur suffisante pour reconstruire des génomes complets. Ce type d'approche ne se concentre plus uniquement sur un milieu donné mais sur des analyses croisées de différents microbiomes (Fierer *et al.*, 2012). Parks *et al.* (2017) ont exploité 1550 métagénomes environnementaux présents dans les bases de données publiques. Ceux-ci ont été assemblés individuellement puis binnés, produisant ainsi plus de 64295 MAG, dont 3438 avec une complétude supérieure à 90 % et moins de 5 % de contamination.

1.3.4 Revue des méthodes de binning

De nombreux outils de binning non supervisé existent. Nous détaillons dans cette section les méthodes de chacun (présentées par ordre alphabétique) et dans le tableau 1.1 (page 34). La compréhension des méthodes de binning non supervisé est nécessaire afin de cerner les avantages et incovénients, notamment en vue de proposer de nouvelles approches.

1.3.4.1 Binning à partir de contigs métagénomiques

Toutes les méthodes décrites dans cette section utilisent une approche hybride, utilisant potentiellement différentes sources d'informations (contigs et/ou lectures).Si elles ne sont pas précisées, la méthode peut alors utiliser indifférement les deux.

ABAWACA (Brown, 2015) découpe les contigs en fragments de 5 kb puis leurs fréquences mono-, di-, tri-nucléotidiques et la couverture moyenne sont calculées. Ensuite, un algorithme de clustering divisif est appliqué à l'ensemble des données. La qualité des clusters est estimée à l'aide de marqueurs de gènes à copie unique.

AbundanceBin (Wu et Ye, 2011) utilise uniquement l'abondance des séquences métagénomiques. Un métagénome est considéré comme un mélange de distributions

de Poisson représentant les génomes (modèle de Lander-Waterman appliqué à la métagénomique). La probabilité qu'une séquence appartienne à un génome donné se fait à l'aide d'un algorithme d'Espérance-Maximisation (E-M). AbundanceBin peut donc travailler sur des séquences plus courtes que les méthodes basées sur la composition mais peut difficilement différencier deux génomes ayant la même abondance dans un même métagénome.

BinSanity (Graham *et al.*, 2017) utilise la couverture de séquence pour alimenter l'algorithme *Affinity-Propagation* (AP). Ainsi, BinSanity n'a pas besoin d'une évaluation préalable du nombre de clusters, mais l'algorithme a une complexité algorithmique temporelle importante, ce qui le rend peu adapté aux grands ensembles de données. Un programme optionnel de raffinement utilise des fréquences de GC% et de tétramères pour regrouper les bins avec une contamination élevée (déterminée par CheckM (Parks *et al.*, 2015)) ou une faible couverture.

BMC3C (Yu *et al.*, 2018) utilise la composition des contigs, la couverture et le biais d'usage du code génétique pour représenter des contigs. Plusieurs algorithmes des k-means avec un nombre variable de cluster sont appliqués en utilisant la distance L1 pour comparer de contigs en fonction de leur représentations. Tous ces résultats de clustering sont utilisés pour construire un graphe de co-occurrence afin de détecter les contigs qui sont souvent regroupés. Ce graphe est ensuite partitionné à l'aide de l'algorithme « *Normalized cuts* » pour représenter les bins finaux.

COCACOLA (Lu *et al.*, 2017) estime le nombre de génomes en utilisant 111 marqueurs de gènes à copie unique (single copy gene, SCG). Les fréquences des tétramères et la couverture moyenne des contigs par échantillon sont utilisées. Ces caractéristiques sont normalisées et juxtaposées de la même manière que CONCOCT dont il s'inspire. Les centres de clusters sont déterminés à l'aide d'un k-means utilisant la norme L1 (ou distance de Manhattan) puis les contigs sont assignées aux différents centres en utilisant une *Non-negative Matrix Factorization* (NMF). Enfin, l'information des lectures *paired-ends* peut être utilisée pour affiner les bins proposés.

CONCOCT (Alneberg *et al.*, 2014) calcule d'abord les fréquences des tétramères et la couverture moyenne des contigs par échantillon. Les fréquences des tétramères et la couverture des contigs sont normalisées indépendamment pour tenir compte de la longueur du contigs. Les deux informations sont intégrées dans un modèle unique par

simple juxtaposition. Les caractéristiques pertinentes sont alors extraites au moyen d'une Analyse en Composantes Principales (ACP). CONCOCT utilise finalement une approche bayésienne variationnelle d'un modèle de mélange gaussien (*Variational Bayesian Gaussian Mixture Model* (VBGMM)) pour regrouper les contigs (Bishop, 2006). L'information des lectures *paired-ends* peut être utilisée pour affiner les bins proposés.

GATTACA (Popic *et al.*, 2018) estime l'abondance des contigs présents dans le métagénomes. Pour ce faire, il utilise un algorithme de dénombrement des k-mers plutôt que de procéder à l'alignement des lectures de séquençage sur les contigs issus d'un co-assemblage de tous les métagénomes. Ces profils d'abondance et les fréquences des tétramères sont normalisés et regroupés à l'instar de CONCOCT (Alneberg *et al.*, 2014). GATTACA est donc conçu pour traiter des dizaines de métagénomes sans avoir besoin d'un co-assemblage ni de l'alignement des lectures.

GroopM (Imelfort *et al.*, 2014) utilise la co-abondance des contigs de plusieurs échantillons. Il requiert un co-assemblage de tous les échantillons et l'alignement des lectures de séquençage aux contigs résultants. Des premiers bins « noyau » sont proposés puis l'utilisateur peut les affiner manuellement (plusieurs visualisations sont proposées à cet effet). GroopM détecte les bins noyaux comme étant une région à haute densité après représentation des données d'abondance. Ce bin « noyau » contient typiquement plusieurs génomes et GroopM utilise les composantes principales d'une ACP appliquée aux fréquences des tétramères sur ce bin « noyau ». Les bins sont ensuite partitionnés à l'aide d'un algorithme de clustering basé sur la transformée de Hough, une technique de vision par ordinateur pour identifier les objets dans les images (GroopM représente les contigs en pixels à ce stade). Ces clusters centraux peuvent être affinés automatiquement à l'aide d'une carte auto adaptative (*Self Organizing Map* (SOM)) sur la représentation des données précédentes (co-abondances et premières composantes principales des profils de fréquence tétramère). Enfin, la distribution en GC de chaque bin est modélisée à l'aide d'une distribution normale et, si cette distribution présente une variance anormalement élevée, le bin est étiqueté comme potentiellement chimérique. étape de pré-assemblage avec un accent particulier sur la scalabilité.

MaxBin (Wu *et al.*, 2016) est également basé sur la composition et l'abondance des contigs et utilise un algorithme d'E-M. Les clusters sont en effet représentés

par un mélange de distributions de Poisson. Les paramètres de ces distributions sont estimés à partir d'une simulation sur 3181 génomes bactériens et archéeens de référence issus de l'Integrated Microbial Genomes (Markowitz *et al.*, 2012). Le nombre de bins est estimé à l'aide de 107 gènes en copie unique. Ces marqueurs sont également utilisés pour évaluer le nombre de génomes au sein de chaque bin après une première étape de clustering. Ces bins sont alors eux-mêmes clusterisés jusqu'à ce qu'il y ait un génome par bin en fonction de ces marqueurs.

MBBC (Wang *et al.*, 2015) estime d'abord le nombre de clusters en recherchant les k-mers partagés parmi les lectures de séquençage (k=16). Ces groupes de k-mers permettent à MBBC d'estimer l'abondance et la couverture des génomes. Le métagénome est alors considéré comme un mélange de distributions de Poisson de paramètres inconnus qui sont estimés à l'aide d'un algorithme d'E-M. Chaque lecture est ensuite affectée à l'une des distributions. Pour chaque cluster, une chaîne de Markov est utilisée pour confirmer l'assignation d'une lecture au cluster.

MetaBAT et **MetaBAT2** (Kang *et al.*, 2015) représentent les contigs selon deux modèles : (i) les distances entre les contigs sont estimées à partir de l'étude des distances inter- et intra-espèces entre la fréquence des tétramères et (ii) la couverture du génome est estimée en utilisant la différence entre un modèle empirique de couverture du génome (construit à partir de 99 génomes isolés) et une distribution normale. Ces deux modèles sont intégrés puis alimentent un algorithme k-médoïdes modifié où les centroïdes sont initialisés en fonction des abondances des contigs. MetaBAT peut fonctionner en utilisant uniquement des informations sur la composition ou des informations sur la composition et l'abondance. De plus, MetaBAT n'accepte pas les contigs d'une taille inférieure à 1,5 kb (2,5 kb est recommandé).

MetaCluster5 (Wang *et al.*, 2012) se déroule en deux étapes. Tout d'abord, les lectures sont séparées en deux groupes : les lectures faiblement abondantes et les lectures très abondantes en fonction des occurrences de 16-mers. Les lectures très abondantes (profondeur de séquençage $>$ 6X) sont regroupées en fonction de leurs 36-mers communs pour construire des contigs virtuels. Le dénombrement des 5-mers de ces contigs virtuels est utilisé pour les clusteriser à l'aide d'un cluster k-means (utilisant une distance de Spearman) avec une initialisation aléatoire des clusters. La deuxième étape met l'accent sur les lectures de faible abondance. Comme pour la première étape, les séquences sont groupées en contigs virtuels mais en utilisant leurs

22-mers partagés, puis la distribution de tétramères de chaque groupe est utilisée pour les clusteriser.

MetaProb (Girotto *et al.*, 2016) regroupe les contigs selon les q-mers partagés ($q = 30$ par défaut). Ces groupes constituent des graines qui seront utilisées par le clustering. Ces graines sont caractérisées en fonction de leur fréquence de tétramères à l'aide d'un sous-ensemble de lectures qui ne se chevauchent pas au sein de la graine pour éviter un dénombrement excessif des tétramères. Ces profils de graines sont comparés entre eux avec une distance euclidienne (norme L2). Le nombre de clusters attendu est déterminé à l'aide de l'algorithme G-means : MetaProb recherche des clusters en utilisant les graines précédentes comme centroïdes et, si un cluster est normalement distribué (selon un test de Kolmogorov-Smirnov), le cluster est validé et ses contigs sont extraites de l'ensemble de données. Le processus est répété jusqu'à ce que l'ensemble de données soit traité. Une fois le nombre de clusters déterminé, l'algorithme des k-means est appliqué. MetaProb accepte les lectures et les contigs en entrée.

MetaWatt (Strous *et al.*, 2012) estime la probabilité qu'une séquence appartienne à un génome donné après avoir décrit une relation empirique entre la moyenne et l'écart-type de la fréquence de ce tétranucléotide. La séquence d'entrée la plus longue est considérée comme le centre d'un premier cluster. Ensuite, la probabilité que la deuxième séquence la plus longue appartienne à l'un des clusters existants est calculée selon le même modèle empirique. La séquence est assignée au cluster ayant la probabilité la plus élevée ou sera utilisée comme le centre d'un nouveau cluster si sa probabilité d'appartenir à un cluster existant est inférieure à un seuil donné. Les séquences suivantes sont traitées de la même manière. Différents outils de validation (affiliation taxonomique, couverture par bin) sont proposés au travers d'une interface graphique pour la curation et l'affinement des bins proposés.

MyCC (Lin et Liao, 2016) prédit les gènes à partir des contigs d'entrée puis recherche 40 marqueurs de gènes à copie unique. Les contigs d'entrée sont caractérisées par leur fréquence en k-mers et, optionnellement, par l'abondance des contigs. Les deux informations (composition et abondance) sont normalisées puis transformées à l'aide du *Centered Log Ratio* (CLR), puis intégrées. Ces données hautement dimensionnelles sont réduites à deux dimensions à l'aide de l'algorithme *Barnes-Hut t-distributed stochastic neighbor embedding* (BH-tSNE), puis les profils de contigs

sont regroupés par AP. Les clusters résultants sont enfin segmentés ou fusionnés à l'aide des marqueurs de gènes à copie unique pour produire des bins.

1.3.4.2 Binning par agrégation d'autres binnings

D'autres outils de reconstruction de génomes sont basés sur des méthodes de binning existantes et parfois intégrés dans des *pipelines*. Ceux-ci procède généralement par agrégation des résultats proposés par différentes méthodes.

Binning_refiner (Song et Thomas, 2017) propose d'améliorer les résultats de binnings existants en combinant les sorties de plusieurs logiciels de binning par comparaison des bins attribués pour chaque séquence d'entrée. Si une séquence est trouvée par tous les logiciels dans le même bin, le bin est alors validé. On parlera de méthode par vote unanime. Actuellement, Binning_refiner ne peut gérer que les sorties MetaBAT, MyCC et CONCOCT.

DAS_Tool (Sieber *et al.*, 2018) identifie des gènes en copie unique pour évaluer des bins communs à plusieurs résultats de binning (figure 1.3). Cette évaluation reflète la complétude et la contamination des bins avec une pénalité spéciale pour les bins contenant plusieurs génomes. Les paramètres de la fonction d'évaluation ont été estimés en maximisant la précision moyenne et le rappel à l'aide d'une communauté microbienne synthétique. Les séquences codantes sont prédites à l'aide de Prodigal (Hyatt *et al.*, 2010) puis comparées aux bases de données de gènes en copie unique. Le bin le mieux noté est alors extrait et ses séquences sont écartées du jeu de données. Les bins restants sont notés à nouveau et l'opération est répétée tant que les notes des bins sont supérieures à zéro ou qu'il ne reste aucun bin. En agrégeant plusieurs résultats de binning, DAS_Tool recherche les bins les plus consensuels en utilisant des critères d'évaluation uniquement biologiques.

MetaWRAP (Uritskiy *et al.*, 2018) est un pipeline complet pour l'extraction du génome des métagénomes qui orchestre le prétraitement de lecture, l'assignation taxonomique des lectures, leur assemblage, le binning des contings, l'agrégation des bins, le ré-assemblage des bins et les annotations taxonomiques et fonctionnelles. L'étape du binning repose sur Binning_refiner (et donc MetaBAT2, MaxBin2 et CONCOCT). Ces résultats sont affinés par agrégation à l'aide de Binning_refiner où toutes les combinaisons des résultats de binning précédents sont traitées. Les

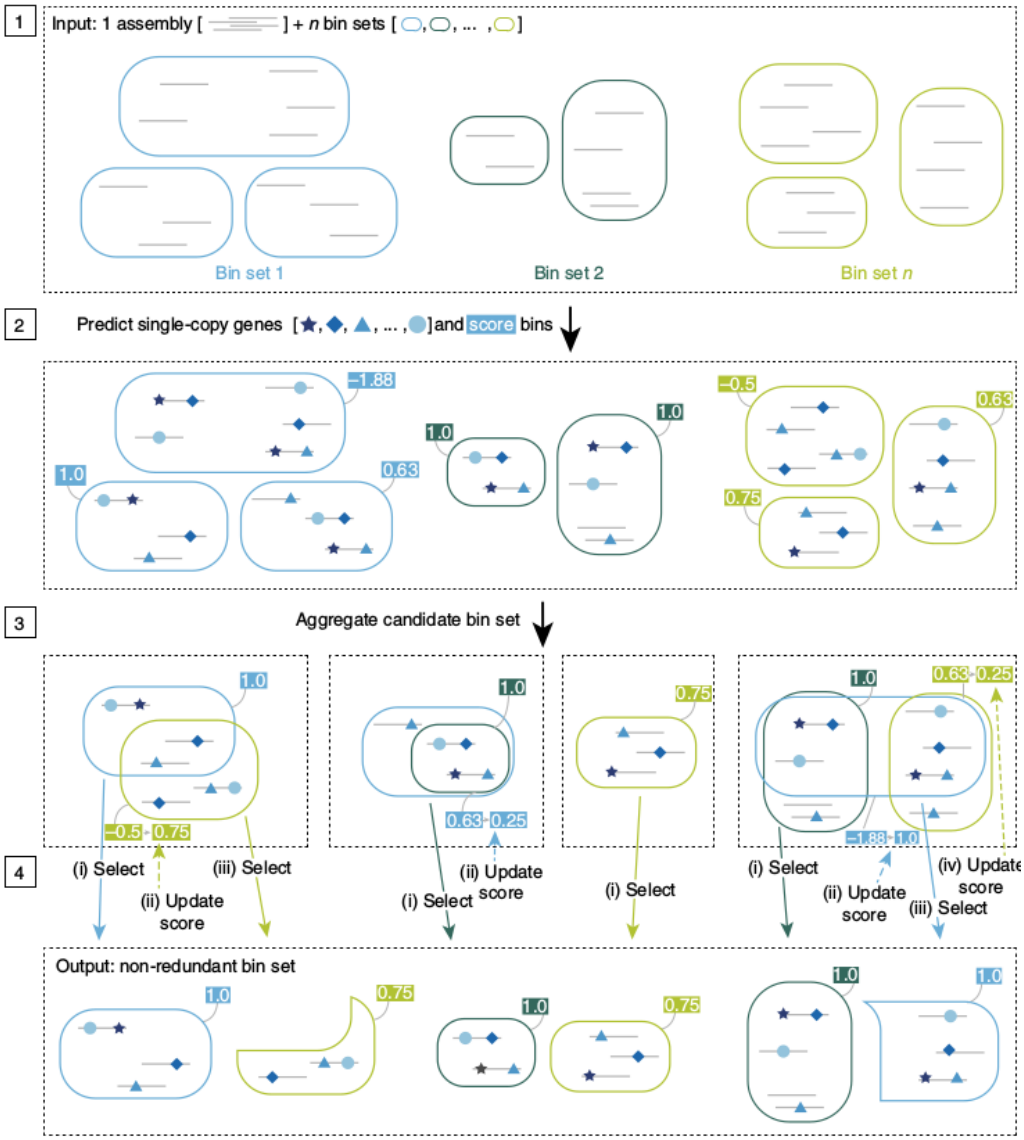


Figure 1.3 | **Vue d'ensemble de la méthode d'agrégation de DAS_Tool** : (1) DAS_Tool accepte des résultats de binning en entrée. (2) Les gènes en copie unique sont prédits pour attribuer un score à chaque bin. (3) Les bins redondants sont fusionnés. (4) Les bins restants sont ré-évalués et extraits si possible. Des bins non redondants sont proposés en sortie à partir de différents résultats de binning. (Figure 1 de Sieber *et al.* (2018))

meilleurs bins hybrides résultants sont extraits après une évaluation par CheckM si la complétude et la contamination respectent les seuils définis par l'utilisateur.

1.3.4.3 Autres méthodes d'intérêt

Sont présentées dans cette section les méthodes qui ne sont pas directement du binning non supervisé mais dont le fonctionnement présente des éléments d'intéret pour le développement de méthodes futures.

DectICO (Ding *et al.*, 2015) est un algorithme de binning supervisé qui repose sur un modèle de séquence peu commun : la corrélation intrinsèque des oligonucléotides (ICO) (Ding *et al.*, 2014). L'ICO « représente la quantification d'une relation intrinsèque entre deux oligonucléotides quelconques » au sein d'une même séquence et a montré de meilleures performances pour la ségrégation des séquences que les méthodes basées sur la composition (*e. g.* : dénombrement des tétramères). Brièvement, les séquences sont découpées en k-mers, eux-mêmes découpés en deux. Chaque k-mer et portion de k-mer est dénombré puis les corrélations entre les dénombrements des deux sous-parties de chaque k-mer sont calculées. Les caractéristiques pertinentes de ces profils sont sélectionnées à l'aide d'une méthode des moindres carrés partiels à noyau puis le classifieur est ajusté à l'aide d'un algorithme récursif basé sur l'algorithme *Support Vector Machine* (SVM).

MGS-canopy (Nielsen *et al.*, 2014) utilise la co-abondance des gènes à travers plusieurs échantillons métagénomiques pour représenter les données (figure 1.4). Les lectures de métagénomique sont assemblées et les gènes sont prédits. Les séquences sont profilées en fonction de l'abondance médiane du gène présent dans la séquence, puis un clustering utilisant l'algorithme Canopy est appliqué. Une séquence est choisie comme centre d'un cluster puis recrute les autres séquences qui lui sont suffisamment proches. Lorsqu'aucune nouvelle séquence ne peut être ajoutée à ce premier cluster, une nouvelle séquence est utilisée pour créer un nouveau cluster et le processus est répété.

VizBin (Laczny *et al.*, 2015) modélise des séquences d'après leur composition en k-mers ($k = 5$ par défaut) puis projette ces données dans un espace bidimensionnel en utilisant l'algorithme BH-tSNE. Cette méthode de réduction de la dimensionnalité permet de représenter les clusters d'un ensemble de données en préservant le voisinage

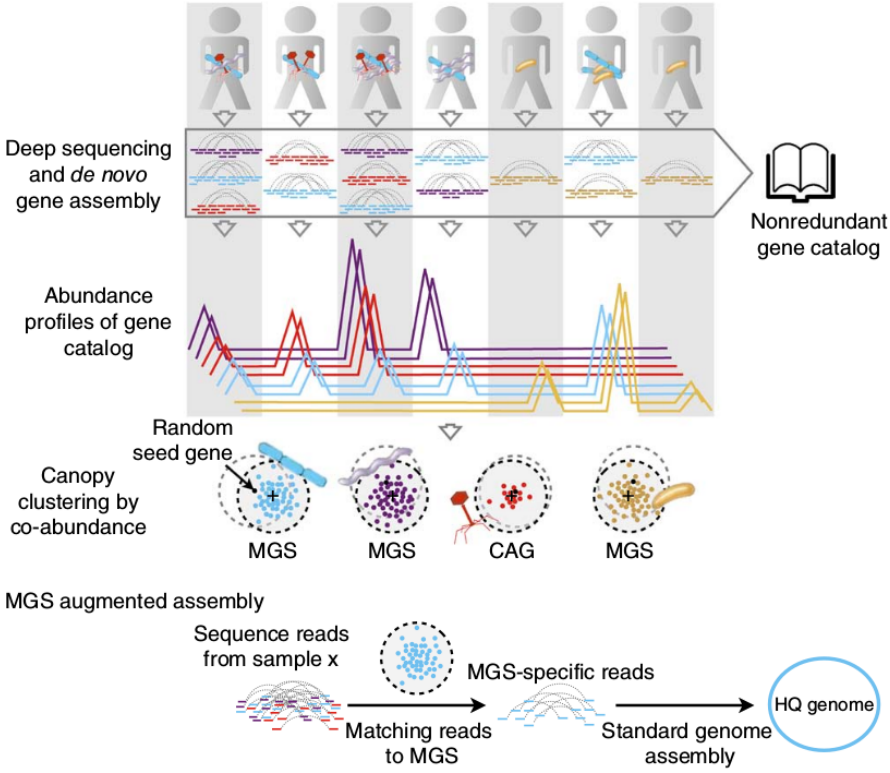


Figure 1.4 | **Vue d'ensemble de l'approche basée sur l'abondance des gènes de MGS-canopy**. Plusieurs métagenomes sont assemblés individuellement puis un catalogue de gènes non redondants est établi. L'abondance de ces gènes permet de caractériser chaque échantillon. Le clustering utilise alors la co-abondance de ces gènes pour construire des groupes de gènes co-abondants (CAG) ou des espèces métagénomiques (MGS) si au moins 700 CAG sont retrouvés. Les séquences de chaque MGS et CAG sont alors extraites puis ré-assemblées. (Figure 1 de Nielsen *et al.* (2014)).

de chaque point et en homogénéisant les densités de grappes. L'utilisateur doit ensuite définir manuellement les clusters à l'aide de cette visualisation des données.

Il est à retenir de cette revue des méthodes que tous ces outils respectent le schéma précédemment décrit : une modélisation numérique des séquences (contigs ou lectures), éventuellement normalisée, puis un clustering.

De plus, les outils de binning se basent sur plusieurs sources d'informations (composition et/ou abondance des séquences) pour modéliser les séquences. Ces différentes informations doivent donc être intégrées pour obtenir une seule modélisation de chaque séquence.

Cependant, l'intégration de ces informations, leur normalisation et le clustering qui est appliqué ensuite est spécifique à chaque méthode. Le détail des modélisations des contigs est donné en section 1.7 et celui des méthodes de clustering est présenté en section 1.8. Des informations complémentaires relatives aux complexités algorithmiques des outils sont données en Annexe A.3.

TABLEAU 1.1 | **Vue d'ensemble des méthodes des logiciels de binning**. Le détail des méthodes de chacun des logiciels est donné en sous-section 1.3.4. La validation des clusters en tant que génomes est considérée comme une étape du binning par certains logiciels, auquel cas ceux-ci reposent sur les propriétés des lectures de séquençage paired-end ou sur des gène-marqueurs simple copie (SC). Les méthodes de normalisation des données ne sont pas présentées dans la mesure où celles-ci sont spécifiques à chaque logiciel et détaillées en sous-section 1.3.4. Le nombre de citations des outils a été évalué à l'aide de Google Scholar le 27 novembre 2018. Les symboles « – » indiquent que le logiciel fonctionne sans cette étape. « n/a » représente les logiciels non publiés ou qui ne ne peuvent pas réaliser l'étape. Les logiciels marqués d'une astéristique ne sont pas des logiciels de binning mais de visualisation ou d'agrégation.

Logiciel	**Année de publication**	**Nombre de citations**	**Données d'entrée**	**Modèles de séquences**	**Réduction de la dimension**	**Nombre de clusters**	**Algorithme de clustering**	**Validation**
ABAWACA	2015	85	Contigs + lectures	Fréquences k-mers + abondances	–	–	Divisif	Marqueurs SC
AbundanceBin	2011	109	Contigs et/ou lectures	Mélange de lois de Poisson	–	–	E-M	–
BinSanity	2017	19	Abondances	Abondances	–	–	AP	–
BMC3C	2018	0	Contigs + lectures	Fréquences k-mers + abondances + code génétique	–	Multiples k-means avec différents k	Ensemble	–
COCACOLA	2017	34	Contigs + lectures	Fréquences k-mers + abondances	–	Multiples k-means avec différents k	NMF	Liens entre *paired-ends*
CONCOCT	2014	327	Contigs + lectures	Fréquences k-mers + abondances	ACP	–	VBGMM	Liens entre *paired-ends*

Logiciel	**Année de publication**	**Nombre de citations**	**Données d'entrée**	**Modèles de séquences**	**Réduction de la dimension**	**Nombre de clusters**	**Algorithme de clustering**	**Validation**
DectICO	2015	5	Contigs	Corrélation intrinsèque des oligonucléotides	kPLS	n/a	n/a	–
GATTACA	2017	1	Contigs	Fréquences k-mers + abondances approximatives	ACP	–	VBGMM	Liens entre *paired-ends*
GroopM	2014	156	Contigs + lectures	Fréquences k-mers + abondances	ACP	–	Transformée de Hough	–
Latent Strain Analysis	n/a	n/a	Lectures	*Hyperplane hashing function*	*Streaming* SVD	–	*custom*	*spiking*
MaxBin	2015	120	Contigs + lectures	Modèle de mélange de lois de Poisson	–	Estimé à partir de 107 marqueurs SC	E-M	107 marqueurs SC
MBBC	2015	16	Contigs	Fréquences 16-mers + mélange de lois de Poisson	–	Multiples d'E-M	E-M	Propriétés de Markov
MetaBAT	2015	283	Contigs (+ lectures)	Modèle de régression pour les fréq. k-mers et abondances	–	–	K-medoids	–
MetaBAT2	2015	–	Contigs (+ lectures)	Modèle de régression pour les fréq. k-mers et abondances	–	–	K-medoids	–

Logiciel	Année de publication	Nombre de citations	Données d'entrée	Modèles de séquences	Réduction de la dimension	Nombre de clusters	Algorithme de clustering	Validation
MetaCluster5	2012	103	Contigs	36-mers partagés + fréquences 5-mers	–	–	K-means distance de Spearman	–
MetaProb	2016	13	Contigs et/ou lectures	30-mers partagés + fréquences k-mers	–	G-means	K-means	–
MetaWatt	2012	121	Contigs	Modèle de régression pour les fréq. k-mers	–	–	Canopy-like	manuelle
MyCC	2016	58	Contigs	Fréquences k-mers (+ abondances)	BH-tSNE	–	AP	Marqueurs SC
Binning_refiner *	2017	8	Contigs + binnings	n/a	n/a	n/a	Agrégation	n/a
DAS_Tool *	2018	10	Contigs + binnings	n/a	n/a	n/a	Agrégation	110 marqueurs SC
MetaWRAP *	2018	3	Binnings	MetaBAT2, MaxBin2, CONCOCT	n/a	n/a	Agrégation	n/a
VizBin *	2015	83	Contigs	Fréquences k-mers	BH-tSNE	n/a	n/a	n/a

1.4 Évaluation de résultats de binning

1.4.1 Évaluation supervisée

Face à la multiplicité des méthodes de binning, la comparaison de celles-ci est devenue un enjeu afin d'évaluer les avantages et les incovénients des outils proposés. (voir en section 1.6). Un jeu de données maitrisé peut servir de point de comparaison : on dispose donc de bins de référence. Ceux-ci représentent le résultat attendu pour cette évaluation, et seront en conséquence considérés comme des classes. Les bins proposés par un outil de binning non supervisé seront considérés comme des clusters.

L'association entre classe et cluster permet l'énumaration des vrais positifs (TP), faux positifs (FP), vrais négatifs (TN) et faux négatifs (FN), et donc la construction d'une matrice de confusion. Cette dernière est alors utilisée pour le calcul de plusieurs métriques descriptives dont l'obtention est détaillée dans cette section.

Pour rappel, une fois les classes et clusters associés :

- les vrais positifs sont les contigs d'une classe qui appartiennent effectivement au cluster qui lui est associé ;
- les vrais négatifs sont les contigs qui ne sont pas de la classe et n'appartiennent pas au cluster associé ;
- les faux positifs sont des contigs de la classe qui n'appartiennent pas au cluster associé ; et
- les faux négatifs sont des contigs qui ne sont pas de la classe mais appartiennent au cluster associé.

La matrice de confusion est utilisée pour calculer divers paramètres tels que la sensibilité (ou rappel), la spécificité, la précision, la fiabilité (*accuracy*), le taux de faux positifs (*false discovrey rate*, FDR), le taux de vrais positifs (*true positive rate*, TPR), le taux de faux positifs (*false positive rate*, FPR), le taux de vrais négatifs (*true negative rate*, TNR) et le score F1 (voir Annexe A.4 pour le détail des formules).

Les métriques de classification binaire (sensibilité, spécificité, précision, fiabilité, score F1) sont calculées pour chaque cluster et moyennées afin de décrire globalement le résultat de cluster. Ces métriques moyennées sont désignées plus tard par "métriques de binning" (par exemple, "précision de binning", "fiabilité de binning"). Certains sont visuellement représentés en figure 1.5.

Les classes comparées à elles-mêmes possèdent logiquement les meilleures métriques possibles : précision de binning, fiabilité du binning, sensibilité du binning,

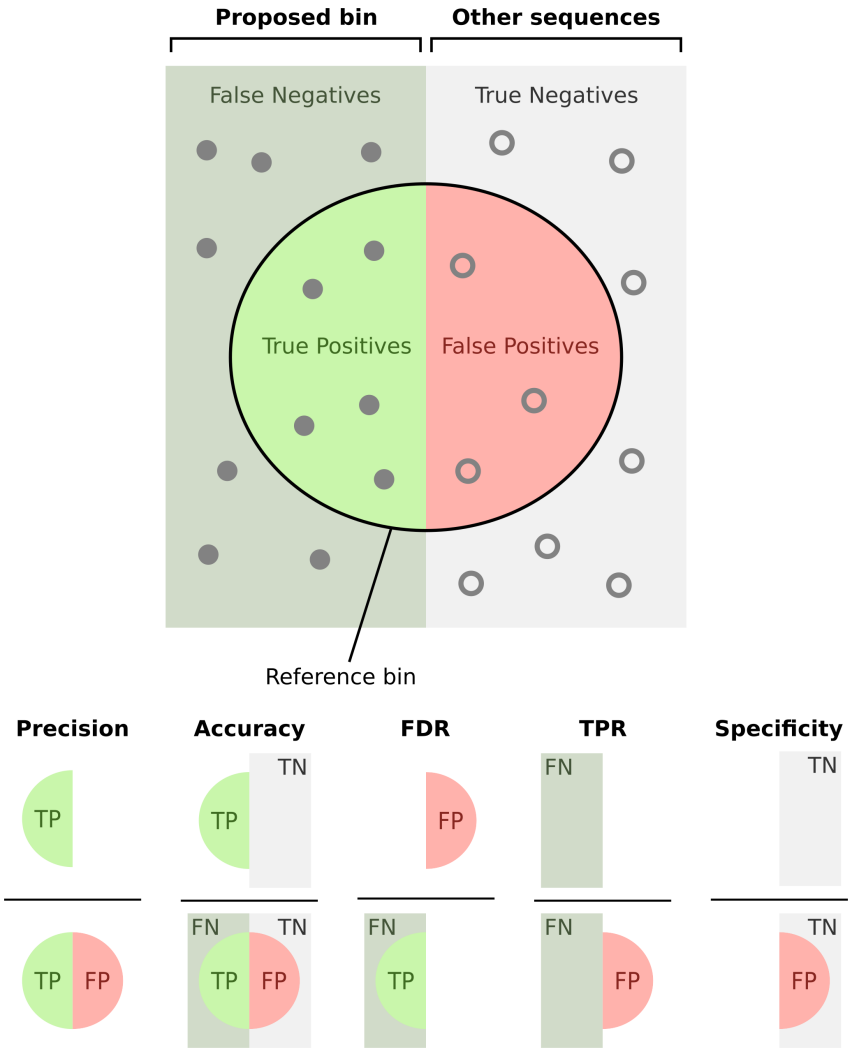


Figure 1.5 | **Représentation schématique des métriques d'évaluation.** Chaque bin représente idéalement un génome. Adapté de Wikimedia.

spécificité du binning, score F1, TPR et TNR sont de 1,0. Les valeurs des FDR et FPR sont de 0,0.

1.4.2 Estimation à partir de marqueurs biologiques

En l'absence de données de référence, un binning peut être évalué par estimation des métriques. Celle-ci repose sur l'utilisation de marqueurs biologiques tels que ceux employés par CheckM (Parks *et al.*, 2015).

CheckM dispose de plusieurs ensembles de marqueurs permettant l'évaluation des taux de complétude et de contamination d'un bin : un ensemble universel, un ensemble spécifique aux domaines (bactérien et archéen) et un ensemble spécifique à

la taxonomie analysée.

Tout d'abord, un gène est identifié comme marqueur s'il est présent en une seule copie (*single copy gene*, SCG) pour 97 % des 3324 génomes de réference et annotés issus de la base de données IMG (Markowitz *et al.*, 2012) représentant 39 classes et 20 phyla. Ces gènes-marqueurs constituent alors un premier ensemble universel. Dans un deuxième temps, des ensembles de gènes-marqueurs spécifiques pour chaque domaine sont construits.

CheckM dispose alors de 104 marqueurs organisés en 58 ensembles pour le domaine des bactéries et de 150 marqueurs organisés en 108 ensembles pour le domaine des archées. Enfin, un arbre de génomes de réferences est pré-établi à partir de 43 gènes-marqueurs hautement conservés extraits de 5656 génomes issus d'IMG. Cet arbre permet alors, à chaque éxecution de CheckM, de déterminer pour chaque groupe taxonomique (soit chaque nœud de l'arbre) les gènes présents en unique copie pour au moins 97 % des génomes appartenant au groupe taxonomique choisi. Après identification des gènes associés entre eux (comme précédemment), ces gènes-marqueurs sont alors organisés en ensembles spécifiques aux différents groupes taxonomiques en présence.

Ces marqueurs sont alors recherchés dans les bins par alignement de séquences. Une fois identifiés dans un bin, ces marqueurs permettent l'estimation du taux de complétude et de contamination. Le **taux de complétude** est alors :

$$\text{completeness}(G) = \frac{\sum_{s \in M} \frac{s \bigcap G_M}{|s|}}{|M|}$$

avec s un ensemble de marqueurs spécifiques à un domaine, M l'ensemble de tous les marqueurs et G_M les marqueurs identifiés dans le bin G. Le **taux de contamination** est calculé comme suit :

$$\text{contamination}(G) = \frac{\sum_{s \in M} \frac{\sum_{g \in s} C_g}{|s|}}{|M|}$$

avec $C_g = \begin{cases} N-1 & \text{si } N > 2 \\ 0 & \text{sinon} \end{cases}$, N étant le nombre de fois où le marqueur a été identifié dans le bin.

Ces taux de complétude et de contanination sont de fait des estimations du TPR et du FDR.

La complétude d'un bin s'évalue de 0 à 100 %. La contamination peut dépasser les 100 % dans le cas où l'on retrouve des marqueurs uniques plus de deux fois dans un même bin.

Une forte contamination (nombreux marqueurs en multicopies) provoquée par la présence de deux génomes proches phylogénetiquement est à distinguer d'une contamination par des séquences phylogénétiquement très éloignées dans la mesure où le premier type de contamination peut mettre en évidence un génome composite représentant un groupe taxonomique alors que le second devrait être écarté ou post-traité. CheckM examine l'AAI (*Average Aminoacid Identity*) entre les marqueurs en multicopies pour estimer l'« hétérogénéité de souche ». Celle-ci est de 0 % si tous les marqueurs sont issus d'un même organisme et de 100 % s'ils sont tous issus d'organismes différents. Il est à noter qu'un bin avec une très forte hétérogénéité représentera alors davantage un groupe taxonomique – on parlera de génome composite – plutôt qu'un génome isolé.

1.4.3 Évaluation intrinsèque

L'utilisation de marqueurs biologiques ne s'adapte pas de fait à la philosophie du travail présenté qui se veut exploratoire et non supervisé. L'évaluation d'un résultat de binning par des métriques ne nécessitant pas de données extérieures devient donc nécessaire.

Les scores *silhouette* (Rousseeuw, 1987) permettent d'évaluer un contig en fonction de sa distance par rapport aux autres contigs de son cluster et par rapport aux autres contigs étrangers à son cluster. Ainsi, deux clusters auront des individus (ici, des contigs) avec des scores *silhouette* proches de $+1$ si les deux clusters sont très nettement séparés les uns des autres. Deux clusters auront des scores proches de 0 s'ils sont confondus et un individu peut avoir un score proche de -1 s'il est très éloigné des autres individus de son cluster et inclus dans un autre cluster.

On peut donc évaluer la qualité d'un cluster sans apport de données extérieures par la distribution des scores *silhouette* des contigs qui le composent. Il est à noter que d'autres métriques d'évaluation interne existent. Seuls les scores *silhouette* sont considérés ici pour leurs performances et leur disponibilité au travers de la bibliothèque scikit-learn (Liu *et al.*, 2010).

1.5 Comparaison de plusieurs binnings

La comparaison de différents résultats de binning est nécessaire notamment pour des approches de binning par agrégation. À l'instar des méthodes de classification et de clustering, ce type d'approche, dite d'ensemble, permettra d'obtenir des résultats particulièrement fiables. Les outils pour comparer des résultats de binning sont présentés dans cette section.

1.5.1 Recherche de marqueurs biologiques

La recherche de gènes-marqueurs permet d'estimer la qualité d'un bin, mais également d'estimer la complémentarité qui peut exister entre deux bins. En effet, des ensembles de marqueurs sont connus et doivent *a priori* être retrouvés complets dans un bin. Si deux bins présentent des caractéristiques communes et une complémentarité dans les marqueurs retrouvés, ceux-ci peuvent alors être fusionnés. C'est exactement l'approche qu'emploie le loigicel DAS_Tool décrit en page 29.

Cependant, comme mentionné précédemment, l'utilisation de données issues de bases de données publiques n'est pas souhaitable dans le cadre du travail présenté.

1.5.2 *Average Nucleotide Identity*

Dans le cadre de la reconstruction non supervisée de génomes, on peut estimer la parenté entre deux bins. Historiquement, la parenté entre deux génomes peut être estimée par l'hybridation ADN-ADN (Wayne *et al.*, 1987). L'identité nucléotidique moyenne (*Average Nucleotide Identity* (ANI)) a été développée comme une quantification approximative *in silico* de cette parenté (Jain *et al.*, 2018).

Pour ce faire, le génome-requête est découpé en fragments de 1020 bases non chevauchants. Ces fragments sont alors alignés sur le génome-référence avec `blastn`On parlera alors d'« hétérogénéité de souche ». Pour ce faire, BLAST (Altschul *et al.*, 1990). Les longueurs des alignements ayant une similarité supérieure à 90 % sont sommées puis divisées par la taille du génome-référence. Ainsi, deux génomes avec un lien de parenté fort auront une ANI de 100 % entre eux alors que la valeur tendra vers 0 % si ces deux génomes n'ont pas de lien de parenté.

Il est à noter que l'ANI entre deux génomes n'est pas réciproque : l'ANI d'un génome A et un génome B peut être différente entre ce génome B et ce génome A. L'ANI ne répond donc pas à la définition d'une distance mathématique. Cette mesure permet néanmoins la comparaison d'une collection de bins entre eux. Il nous

est ainsi permis d'estimer la complémentarité qui peut exister entre les résultats de binning proposés par plusieurs outils pour un même jeu de données en l'absence d'un résultat de référence.

1.5.3 Validation des bins en MAG

Le binning a pour vocation de permettre la reconstruction de MAG depuis des métagénomes en vue de les étudier individuellement. Or, les bins proposés sont de qualité variable suite à l'estimation de leurs taux de complétude et de contamination (voir sous-section 1.4.2).

Ces métriques permettent de définir des niveaux de qualité standardisés des MAG (Bowers *et al.*, 2017). Un bin peut ainsi être classé selon trois manières : haute qualité (*High quality*, HQ), qualité medium (*Medium quality*, MQ) ou basse qualité (*Low quality*, LQ). Ceux ne validant aucun critère (typiquement, contamination $\geq$ 10 %) ne sont alors pas considéré comme des MAG et consitue dans notre cas une quatrième classe (tableau 1.2). La classification dépend donc de la pertinence des estimations faite par CheckM et peut donc souffrir des limites de ce dernier.

Tableau 1.2 | **Critères d'acceptation des bins en MAG** selon Bowers *et al.* (2017)

Qualité	Complétude minimum (%)	Contamination maximum (%)
Haute	90	5
Medium	50	10
Basse	–	10

Il est à noter que ces critères ne rendent pas compte de la possibilité pour les outils de binning de produire des génomes composites. Ces derniers représentent des groupes taxonomiques entiers et sont généralement perçus comme fortement contaminés. Ils ne doivent pas être traités comme des génomes individuels (Shaiber et Eren, 2019) mais restent porteurs d'informations (Schwartz *et al.*, 2015).

1.6 Évaluation des méthodes de binning

Le challenge *Critical Assessment of Metagenome Interpretation* (CAMI) (Sczyrba *et al.*, 2017) propose un cadre pour l'évaluation de diverses analyses métagénomiques.

Assessment of Metagenome BinnERs (AMBER) (Meyer *et al.*, 2018) est la seule initiative à ce jour pour l'évaluation des méthodes de binning. CAMI propose en effet divers jeux de données métagénomiques simulées de composition connue. Cette évaluation est une ressource précieuse pour le choix d'un logiciel de binning.

Brièvement, le jeu de données le plus complexe proposé par le CAMI (dans sa première version) et utilisé par AMBER est simulé à partir de 596 génomes et 478 éléments circulaires (plasmides et virus). Cinq réplicats sont construits en ayant des abondances relatives suivant une distribution log normale et corrélées entre les réplicats pour représenter une série temporelle. Chaque réplicat mime des données Illumina HiSeq avec 15 Gpb, soit 50 millions de lectures 2×150 pb paired-end avec une taille d'insert de 270 bp ($\pm$ 10 %). L'intégralité de ce jeu de données représente 75 Gbp.

Un assemblage de référence est ensuite réalisé à partir de ces lectures simulées avec le logiciel IDBA-UD (Peng *et al.*, 2012). 39140 contigs (taille totale : 2,80 Gbp) ont alors été reconstruits.

Puisque les lectures sont simulées, l'origine de chaque contig est connue et utilisée pour évaluer les résultats de binning. Meyer *et al.* (2018) ont testé 11 logiciels de binning avec une interprétation des résultats ciblée sur leur pertinence biologique (sans se préoccuper des aspects algorithmiques).

Les bins produits sont alors associés aux bins de référence selon le nombre de nucléotides partagés. Quatre métriques sont calculées pour décrire les résultats : la pureté moyenne (*average purity*, ou spécificité), le taux de complétude moyen (*average completeness*, ou sensibilité), la fiabilité (*accuracy*) et l'indice de Rand moyen (*Average Rand Index*, ARI). Il est à noter que les contigs non traités ne sont pas considérés dans le calcul de ces métriques par AMBER, réduisant mécaniquement le nombre de faux positifs et de faux négatifs aux dépens d'une portion de données non traitées. De plus, cette évaluation n'informe pas sur les aspects algorithmiques des méthodes testées (ressources consommées, algorithme utilisé et scalabilité).

Toutefois, les expériences de métagénomique menées, par exemple, dans le sol ou l'eau (Sunagawa *et al.*, 2015) incluent des procaryotes mais également des protistes, des champignons et des virus. L'influence de ces « contaminants » a été peu étudiée quant à la fiabilité des outils de binning. Les jeux de données proposé par le CAMI n'incluent pas pas de de micro-eucaryotes notamment.

De plus, les modélisations des contigs utilisées pour le binning (*e. g.* : composition de tétramères) peuvent être pertinentes pour certains taxons mais ineffectives pour d'autres. Ainsi, les résultats de binning non supervisés nécessitent une évaluation à

un niveau taxonomique plus fin, notamment en présence de contaminants. En effet, certains environnements sont dominés par des phyla spécifiques, rendant ainsi le choix de la méthode de binning crucial.

1.7 Modélisations non supervisées des séquences métagénomiques

Différentes modélisations de contigs utilisées pour le binning non supervisé mais aussi pour la classification ou la visualisation des contigs sont présentées dans cette section.

Cependant, les contigs doivent être prétraités par fragmentation afin d'uniformiser la distribution de leur taille. Un long contig (supposément fiable) sera représenté par une collection de fragments plutôt qu'un seul, et influe donc davantage le binning, en particulier durant la phase de clustering.

Ces contigs fragmentés sont ensuite modélisés. La comparaison de deux contigs devient alors une opération sur deux vecteurs numériques. Celle-ci permet un gain de performance notable par rapport à la comparaison des séquences lorsqu'elles sont représentées par des chaines de caractères ((Wood et Salzberg, 2014)). Cette modélisation numérique des contigs permettra leur traitement par des méthodes de clustering usuelles. L'intégration de ces différentes sources d'information est détaillée en page 48.

1.7.1 Modélisations employées par les logiciels de binning

La composition et l'abondance des contigs sont les deux modèles principalement utilisés par les méthodes de binning existantes (sous-section 1.3.4).

La **composition en k-mers** des contigs est une caractérisation ne nécessitant pas d'apport externe de données et est dans ce cadre un des modèles les plus utilisés pour le binning de contigs (Teeling *et al.*, 2004). Dans la mesure où il n'est pas possible de savoir quel brin d'ADN a été séquencé, il est nécessaire de recourir aux k-mers canoniques : chaque k-mer rencontré est complémenté puis inversé et c'est le plus petit des deux lexicographiquement qui représentera le k-mer rencontré (*e. g.* : « TAC » sera représenté par le k-mer canonique « ATG »). Ainsi, deux contigs ayant une composition similaire peuvent être considérés comme étant originaires d'un même organisme.

Il est également possible de modéliser les contigs en fonction de leurs **abondances relatives** (*e. g.* : Wu et Ye (2011)) dans plusieurs réplicats d'un même métagénome. Ainsi, les relations entre les contigs reposent sur les corrélations entre ces niveaux d'abondances, deux contigs ayant des abondances fortement corrélées au sein des réplicats étant probablement issus d'un même génome.

1.7.2 Modélisations existantes mais non appliquées au binning

D'autres modélisations de séquences n'ont pas encore été appliquées à des problématiques de binning non supervisé.

Les **graines espacées** (« *spaced seeds* ») sont des k-mers pour lesquels certaines positions sont ignorées, ce qui s'avére pertinent lors de la recherche de similarités entre séquences (Keich *et al.*, 2004), pour l'assemblage de lectures de séquençage *de novo* (Birol *et al.*, 2015) ou encore pour la classification (ou binning supervisé) de lectures de séquençage issues d'un métagénome (Břinda *et al.*, 2015). À l'instar des k-mers, les graines espacées livrent une information quant à la composition des séquences.

La **corrélation intrinsèque des oligonucléotides** (Ding *et al.*, 2014) est un modèle de séquences qui représente la quantification des relations intrinsèques qui peuvent exister entre des oligonucléotides, généralement des tri- ou tetra-nucléotides. Ce modèle de données a notamment été utilisé pour classer – de manière supervisée donc – des séquences métagénomiques (Ding *et al.*, 2015).

Le **profil des distances inter-nucléotides** (Xie *et al.*, 2017) est un autre modèle de séquences qui se base sur le nombre de nucléotides présents entre deux nucléotides donnés (*e. g.* : le nombre de nucléotides entre deux adénines ou entre une adénine et une cytosine). Ce modèle a été développé pour la visualisation d'un ensemble de séquences et permet une bonne discrimination des séquences en fonction de leur origine taxonomique, et donc potentiellement utile dans la reconstruction non supervisée de génomes.

On peut également citer la **proportion de nucléotides inclus dans des séquences codantes** (ou « *coding density* »). Elle est définie par la somme des longueurs des séquences codantes prédites divisée par la longueur de la séquence à partir de laquelle les séquences codantes ont été prédites. Les séquences codantes doivent donc être prédites en amont. Ce modèle est notamment utilisé pour la visualisation et la curation des résultats de binning (Imelfort *et al.*, 2014; Eren *et al.*, 2015; Parks *et al.*, 2015) comme étant un critère de validation *a posteriori* d'un

résultat de binning.

Le **traitement du langage naturel** permet de représenter des données textuelles (*e. g.* : mots) sous forme de vecteurs numériques à l'aide d'un réseau de neurones qui a été entraîné de façon non supervisée (Mikolov *et al.*, 2013). Cette représentation numérique des mots permet par exemple de calculer mathématiquement des analogies, comme les célèbres exemples « $king - man + woman = queen$ » ou « $Moscow - Russia + Paris = France$ ». L'arithmétique permise par cette modélisation trouve notamment des applications dans la traduction automatique (Jansen, 2017).

Cette représentation du langage a été transposée à la composition de séquences d'ADN (qui représentent alors des phrases) elles-mêmes composées de k-mers (qui représentent les mots de cette phrase) *via* dna2vec (Ng, 2017). Chaque k-mer est donc représenté par un vecteur numérique.

Dna2Vec repose sur le modèle Word2Vec (Mikolov *et al.*, 2013) pour la représentation numérique dense des mots grâce à un réseau de neurones. En particulier, le modèle du « *skip-gram* » est ici employé (figure 1.6). Ce modèle permet, après un entrainement non supervisé, la prédiction du contexte d'un mot (les n mots entourant le mot-cible).

Dna2vec utilise ce modèle en découpant les contigs en k-mers avec k variant entre deux bornes. Plusieurs découpages de chaque contig sont réalisés afin de couvrir davantage de mots et de contextes. Ce modèle a été pré-entrainé sur le génome humain HG38 et peut être réutilisé pour d'autres applications par « *transfer learning* ». Ainsi, un contig est représenté par un vecteur de mots, chaque mot étant lui-même représenté par un vecteur numérique de taille fixe. Le contig est donc représenté par une matrice en deux dimensions.

La représentation d'une phrase en un seul vecteur est alors possible grâce à sentence2vec (Arora *et al.*, 2016). Cette méthode utilise alors les composantes principales de cette matrice (la méthode de l'ACP est détaillée en page 48). *In fine*, le contig peut être représenté par un vecteur numérique unidimensionnel. Ces approches issues du traitement du langage naturel n'ont pas été utilisées pour le binning non supervisé à notre connaissance.

1.7.3 Manipulation des modélisations des contigs

Les différentes modélisations précédemment présentées permettent de manipuler des vecteurs numériques plutôt que des séquences. Un ensemble de contigs peut donc

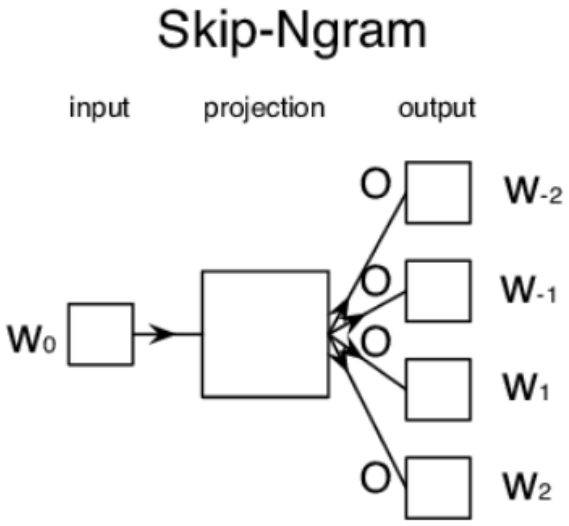


Figure 1.6 | **Vue schématique du module skip-gram utilisé par dna2vec**. Le mot W_n est utilisé pour prédire les mots W_{n-2}, W_{n-1}, W_{n+1} et W_{n+2}.

être représenté par une matrice numérique – représentant un espace de données – à l'aide d'une des modélisations (chaque modélisation produit une nouvelle matrice), chaque ligne étant le vecteur des attributs d'un contig. Le nombre d'attributs définit la dimension de l'espace des données. Cette section et les suivantes présentent les différents outils nécessaires à l'intégration de ces différentes sources d'informations que sont les modélisations en une seule représentation.

La pertinence de ces attributs pour la phase de clustering qui suit n'est pas connue à l'avance. En effet, il se peut que deux attributs soient très fortement corrélés et donc redondants. Or, un trop grand nombre de dimensions peut entraîner des phénomènes indésirables qui ne seraient pas observés dans un espace de dimension moindre. Pour lutter contre ce problème connu sous le terme de « fléau de la dimension », il est nécessaire de réduire le nombre de dimensions de cet espace.

CONCOCT est un des seuls logiciels de binning à utiliser l'ACP pour limiter la quantité d'information à manipuler d'une part (et réduire le fléau de la dimension), mais également pour intégrer les deux modélisations des contigs (composition et abondance des séquences). Ces deux modèles font cependant l'objet de normalisation spécifique.

Il existe de multiples façons de réduire la dimension d'un espace de données, toutes entrant dans deux grandes catégories qui sont (i) la sélection des caractéristiques et (ii) l'extraction des caractéristiques. La **sélection des caractéristiques** consiste à éliminer les attributs les moins discriminants sans changer les valeurs des attributs.

Un exemple simple consiste à filtrer les attributs les moins variants. L'**extraction des caractéristiques** permet quant à elle de transformer l'espace des données en un nouvel espace en modifiant les valeurs des attributs d'origine tout en réduisant le nombre de dimensions. L'espace des données est alors transformé en espace des caractéristiques (ou « *feature space* »).

1.7.3.1 Analyse en Composantes Principales (ACP)

L'ACP est une méthode d'extraction de caractéristiques non supervisée. Cette méthode consiste dans un premier temps à établir la matrice de corrélation linéaire (ou de covariance) entre les paires d'individus (ici, des contigs) – chacun étant caractérisé par ses attributs – puis à chercher les valeurs propres de la matrice de corrélation par une décomposition en valeurs singulières (*Singular Value Decomposition* (SVD)). Les attributs de cette matrice de corrélation sont ensuite triés par ordre décroissant des valeurs propres. Ces dernières permettent de quantifier l'information portée par chaque attribut de cette projection. Leur inertie est alors exprimée comme la valeur propre de cet attribut divisée par la somme des valeurs propres.

L'utilisateur peut ensuite choisir de ne conserver que les n premiers attributs de cette projection – ceux portant le plus d'inertie donc –, ne conservant que les attributs les moins corrélés entre eux et donc les moins redondants. La représentation des données qui en résulte constitue alors un espace des caractéristiques. Cette nouvelle représentation des données permet souvent de drastiquement réduire le nombre d'attributs à manipuler tout en conservant la plupart de l'information portée par les données d'origine. Une récente revue de la méthode et de ses variantes est proposée par Jolliffe et Cadima (2016).

1.7.3.2 Normalisations

La corrélation linéaire utilisée dans l'ACP nécessite souvent une normalisation préalable de l'espace de données pour permettre une comparaison qui fait sens et ainsi estimer une distance entre deux contigs. Ces distances sont nécessaires pour la phase de clustering.

Dans le cas du dénombrement des k-mers, la valeur de chaque attribut est divisée par la somme des valeurs de tous les attributs pour une séquence donnée . On peut alors calculer une distance euclidienne entre deux contigs sans prendre en compte leur taille respective (figure 1.7). Dans le cas d'une modélisation par l'abondance des contigs, cette normalisation se fait en divisant l'abondance de chaque contig

par l'écart-type des abondances. Cependant, aucun consensus sur la méthode de normalisation n'a encore été trouvé (voir sous-section 1.3.4).

La distance cosinus $d_{cos} = 1 - \frac{u \cdot v}{||u|| \ ||v||}$ utilise l'angle formé par les vecteurs u et v définis par l'origine de l'espace et leurs attributs. Ainsi, à la différence des distances de Manhanttan, euclidienne ou corrélation de Pearson, cette distance devient applicable à des données brutes (*e. g.* : dénombrement de tétramères), des données normalisées ou compositionnelles. Ces dernières sont un cas particulier de données normalisées lorsque les attributs sont ramenés à la taille de l'échantillon. La distance cosinus est couramment utilisée pour le traitement du langage naturel. La distance cosinus peut donc être appliquée à un large panel de données issues des modélisations précédemment décrites sans nécessiter d'ajustement spécifique.

1.7.3.3 Astuce du noyau

La comparaison de deux contigs dans un problème d'apprentissage automatique se fait très souvent par le produit scalaire des attributs caractérisant ces deux contigs. Or, ceci impose la recherche de relations linéaires entre contigs et empêche donc la séparation de ces contigs selon des critères non linéaires. Il est cependant possible d'adapter n'importe quel algorithme à des données non linéairement séparables par l'utilisation de l'astuce du noyau, ou « *kernel trick* » (Schölkopf *et al.*, 1998).

L'espace des données est projeté dans un espace H de plus grande dimension dans lequel une séparation linéaire peut être réalisée sur ces données. Le calcul de cette H peut être intractable car de dimension infinie.

L'astuce du noyau consiste à calculer, à partir de l'espace de données initiales, des distances entre les modélisations des contigs dans l'espace H sans nécessiter le calcul explicite de ce dernier. Le calcul de ces distances est donc assuré par un noyau, c'est-à-dire une fonction définie semi-positive : les valeurs propres de l'application de cette fonction sur les données doivent être positives ou nulles. Des distances peuvent également être utilisées en lieu et place des noyaux (Schölkopf, 2000).

L'astuce du noyau trouve des applications dans d'autres méthodes de l'apprentissage automatique puisqu'elle permet l'utilisation d'une méthode linéaire pour résoudre un problème dont les données d'origine ne sont pas linéairement séparables. Dans l'exemple de l'ACP, l'utilisation d'une matrice de corrélation ne permet que de capter les relations linéaires entre les attributs, ce qui peut conduire à l'occultation d'une partie des informations. Pour capter ces relations non linéaires entre les individus, il est possible de recourir à l'astuce du noyau avant de procéder à la

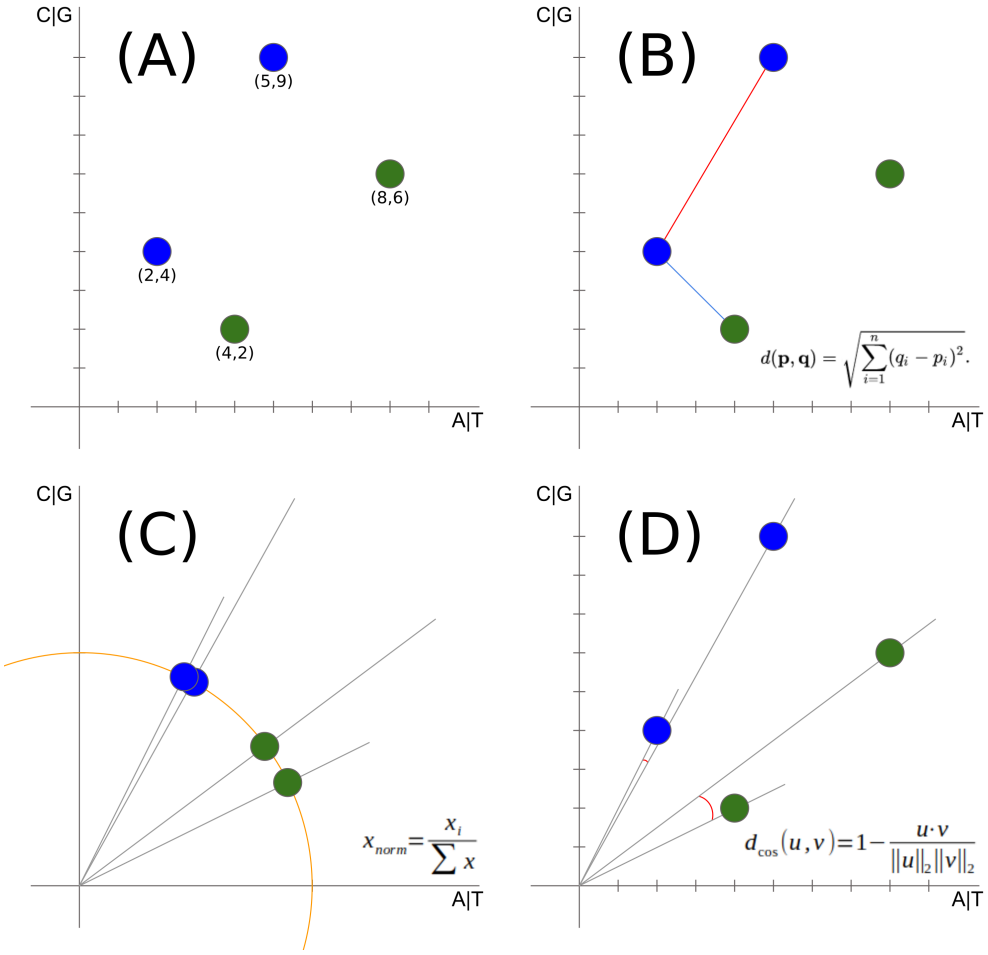


Figure 1.7 | **Illustration des différentes normalisations du dénombrement de k-mers**. **(A)** Quatre contigs sont caractérisées par deux attributs (1-mer canonique : « A ou T » et « G ou C ») qui permettent de les placer dans un espace de données. Les contigs bleues ont une composition très proches mais des tailles différentes, de même pour les contigs vertes. Il est attendu que les contigs bleues soient proches l'une de l'autre en terme de distance, que les contigs vertes le soient également entre elles et que les contigs bleues et vertes soient éloignées. **(B)** Le calcul d'une distance euclidienne entre deux contigs est possible mais ne reflète pas le résultat attendu. La distance entre les deux contigs bleues (en rouge) est supérieure à la distance entre la contig bleue et la contig verte (en bleu). **(C)** Les attributs des contigs peuvent être normalisés par la taille des contigs, ce qui consiste à projeter les données sur un cercle unité (en jaune) : le calcul d'une distance euclidienne entre deux contigs correspond à la comparaison de la composition des contigs. **(D)** Il est également possible de s'affranchir de la normalisation par la taille des contigs en se basant sur les angles formés par les vecteurs représentant les contigs pour calculer une distance respectant le résultat attendu.

recherche des composantes principales. Une ACP utilisant un noyau pour comparer des individus (ici, des contigs modélisés) plutôt que la corrélation linéaire est appelée ACP à noyau (« *kernel PCA* »).

Ainsi, on dispose d'un outil de réduction de dimension basé sur l'ACP et la distance cosinus capable de transformer l'espace des données en un espace de caractéristiques sans recourir à une normalisation spécifique à chaque modélisation. On réduit ainsi le biais implicite dû à l'utilisation des seules relations linéaires sur des données pour lesquelles on ignore les relations.

1.7.3.4 Intégration des modèles

Comme mentionné précédemment, chaque modélisation de contigs produit un nouvel espace de données. Ces derniers peuvent être projetés dans des espaces de caractéristiques afin d'éliminer les données redondantes d'une part, et de limiter le fléau de la dimension. Or, la phase de clustering nécessite une unique représentation des contigs qui prendra la forme de vecteurs numériques.

Par exemple, CONCOCT dénombre les tétramères et mesure l'abondance relative des contigs dans différents échantillons, normalise indépendamment ces deux informations puis les intègre. Cette intégration consiste à produire une unique représentation des contigs à partir de multiples modélisations. Pour ce faire, CONCOCT juxtapose horizontalement les matrices contigs-attributs (on concatène les attributs issus des deux modèles pour chaque séquence) puis utilise l'ACP pour décorréler les attributs issus des deux modélisations. Dans ce cas particulier, l'ACP se base sur la corrélation linéaire de Pearson (ou sur la covariance si les données n'ont pas été centrées et réduites) et les normalisations sont définies spécifiquement pour intégrer ces deux modèles de contigs.

L'utilisation de l'ACP à noyau cosinus sur ces différents espaces de données permet de s'affranchir des normalisations spécifiques. Les espaces de caractéristiques produits sont alors centrés et réduits indépendemment pour les rendre comparables. Enfin et à la manière de CONCOCT, ces espaces de caractéristiques sont juxtaposés horizontalement puis une ACP « classique » est appliquée à cette nouvelle matrice. Les attributs des différents espaces des caractéristiques redondants sont ainsi éliminés.

Ainsi, les multiples modélisations numériques qui ont pu être faites des contigs sont synthétisées en une unique modélisation pour chaque contig. Ces représentations numériques des contigs seront alors le point d'entrée des algorithmes de clutering pour la seconde phase du binning.

1.7.3.5 Visualisation

La réduction du nombre de dimensions est également employée à des fins de visualisation des données. Alors que des données de dimension n (souvent $n >> 3$) se trouvent dans un espace qu'il peut être impossible de représenter, la réduction de dimensions permet la projection des données en un espace uni-, bi- ou tridimensionnel, qui peut alors être visualisé à l'aide d'un nuage de points. L'ACP peut être utilisée dans ce cadre et informera de l'inertie totale portée par les deux axes représentés.

Plus récemment, la *Barnes-Hut t-distributed stochastic neighbor embedding* (BH-tSNE) (Van Der Maaten et Hinton, 2008; Van Der Maaten, 2014) permet de projeter des données de grande dimension dans un espace de visualisation, souvent bidimensionnel, tout en préservant les structures locales. L'utilisation de BH-tSNE pour projeter les données multidimensionnelles sur deux dimensions est non paramétrique, non linéaire et tend à préserver le voisinage des individus tout en homogénéisant la densité des clusters, offrant ainsi des visualisations humainement simple à lire.

Cette méthode de visualisation est particulièrement utilisée pour attester de la pertinence d'une modélisation de données. En effet, une visualisation où des individus censés être retrouvés dans un même voisinage sont effectivement regroupés permettrait d'attester d'une modélisation des données adaptée pour des problématiques de classification ou de clustering.

1.8 Clustering appliqué aux données métagénomiques pour la reconstruction de génomes

Le domaine du clustering dispose d'un large panel de méthodes qui trouvent des applications dans de très nombreux domaines nécessitant le traitement de données (Aggarwal et Reddy, 2013). Nous ne nous focaliserons que sur les applications pour la reconstruction de génomes. Le clustering est en effet, après modélisation des contigs, une approche nécessaire à la reconstruction non supervisée des génomes à partir de données métagénomiques. Le clustering consiste donc à associer des individus (ici, des contigs modélisés) en groupes, ou « *clusters* ». Ces clusters représenteront alors les génomes présents dans le métagénome.

1.8.1 Description des méthodes

Cette section décrit brièvement les familles de méthodes de clustering utilisées pour le binning des contigs en complément de la revue des méthodes de binning (sous-section 1.3.4). Notamment, MetaBAT, MetaCluster, MetaProb utilisent des méthodes par partionnement ; BinSanity et MyCC utilisent une approche hiérarchique ; AbundanceBin, CONCOCT, GATTACA, MaxBin, MBBC utilisent une approche par modèle de mélange. VizBin repose sur une interprétation humaine de la densité des données pour définir des clusters (voir tableau 1.1 en page 34).

1.8.1.1 Clustering par partitionnement

Une des premières approches de clustering est celle de l'algorithme des **k-moyennes** (ou « *k-means* ») (Lloyd, 1982)). Cette méthode est encore largement utilisée et étudiée. Une récente revue est proposée par Jain (2010). La méthode des k-means consiste à sélectionner k individus parmi la population (ici, l'ensemble des contigs) pour en faire des centres de clusters, puis à assigner tous les autres individus au centre le plus proche. Le nombre de cluster k et le choix de la distance sont à la charge exclusive de l'utilisateur.

K-means++ (Arthur et Vassilvitskii, 2007) est une des variantes les plus utilisées qui choisit un premier centre au hasard, puis les suivants de façon à ce qu'ils soient suffisamment éloignés des autres centres. L'opération est alors répétée jusqu'à ce que k centres aient été choisis. On procède ensuite à l'assignation des individus aux centres comme vu précédemment. Le choix des centres influençant le résultat de l'algorithme, il est répété plusieurs fois afin de minimiser la somme des distances entre chaque individu et son centre attribué. La meilleure solution est finalement retenue. Ce type d'approche est particulièrement adapté lorsque les clusters sont de taille homogène, sphériques, et les données linéairement séparables.

1.8.1.2 Clustering hiérarchique

Le clustering hiérarchique considère que chaque individu constitue initialement un cluster puis, à l'aide d'une mesure de distance entre deux individus, fusionne les deux clusters les plus proches et répète l'opération jusqu'à l'obtention du nombre de clusters souhaité. Plusieurs variantes existent selon la mesure la distance entre deux clusters, aussi appelée « *linkage* ». On peut choisir la distance la plus petite entre deux individus de deux clusters (« *single-linkage* ») ou la plus grande distance

entre deux points de deux clusters (« *complete-linkage* »), ou encore la distance moyenne entre toutes les paires de points de deux clusters (« *average-linkage* »), ou l'augmentation de la variance si deux clusters doivent être fusionnés (méthode de Ward). Le choix du type de *linkage* va influer sur la forme des clusters.

Un clustering hiérarchique peut se baser sur un autre algorithme de clustering. L'algorithme G-means (Zhao *et al.*, 2009) procède à un premier k-means, puis recommence sur chacun des clusters proposés de façon à construire une hiérarchie. Comme pour le k-means, ce type d'approche est adapté lorsque les clusters sont de taille homogène, sphériques, et les données linéairement séparables.

1.8.1.3 Clustering par densité

Une autre approche de clustering consiste à identifier des régions de l'espace ayant une densité d'individus jugée suffisamment importante pour y détecter un cluster. C'est par exemple le cas de l'algorithme **DBSCAN** (Ester *et al.*, 1996) (*Density-Based Spatial Clustering of Applications with Noise*) : si au moins $MinPts$ individus se situent dans un rayon ϵ, alors un cluster est formé. On parcourt ensuite le voisinage de ce cluster pour lui incorporer les individus du voisinage tant qu'ils sont à une distance inférieure à ϵ, sinon on cherche un nouveau cluster. De nombreuses extensions de cet algorithme ont été développées, et notamment **HDBSCAN** (Campello *et al.*, 2013) (*Hierarchical DBSCAN*) qui estime automatiquement les paramètres $MinPts$ et ϵ. Contrairement à l'algorithme des k-means, les clusters peuvent être de forme arbitraire, de taille variable et l'algorithme peut permettre une élimination des individus aberrants puisque tous les individus ne sont pas nécessairement assignés à un des clusters.

1.8.1.4 Clustering par modèle de mélange

Il est également possible d'envisager la population comme un ensemble de groupes qui respectent chacun une distribution statistique (ou composante) : on parlera alors de modèle de mélange (« *mixture model* »). Le clustering de cette population consiste donc à décomposer ce mélange pour retrouver les groupes qui le composent.

L'utilisation de composantes gaussiennes reste une des applications les plus fréquentes d'un modèle de mélange, ainsi qualifié de **modèle de mélange gaussien** (*Gaussian Mixture Model*, GMM). L'assignation des individus aux différentes composantes se fait généralement par calcul de la probabilité d'appartenance à une des composantes, composantes dont il nous faut estimer pour chacune les paramètres

(ici, la moyenne et la variance) (Bishop, 2006). Pour ce faire, l'introduction d'une variable latente symbolisant l'appartenance d'un individu à une des composantes permet l'utilisation d'un algorithme d'Espérance-Maximisation (E-M).

On initialise d'abord les composantes, soit au hasard, soit en fonction du résultat d'un k-means par exemple. On calcule ensuite la probabilité de chaque individu d'appartenir à chaque composante (étape E) puis on calcule à nouveau les paramètres des composantes en fonction des individus qui lui ont été assignés (étape M). La fonction-objectif reste celle de l'algorithme E-M, c'est-à-dire qu'on cherche à maximiser la somme des log-probabilités d'appartenance des individus aux composantes. Il s'agit d'un processus d'optimisation : il convient donc de le relancer plusieurs fois pour ne conserver que le meilleur résultat. En effet, l'initialisation du mélange gaussien peut influer le résultat final.

Les paramètres des composantes gaussiennes peuvent être perçus comme des variables aléatoires et non observées, et donc être estimés par inférence bayésienne. Pour ce faire, il est possible de recourir à une inférence variationnelle, la seconde étant une approximation analytique de la solution de l'inférence bayésienne. Pour des explications formelles de l'algorithme d'**inférence variationnelle bayésienne d'un modèle de mélange gaussien** (*Variational Bayesian Gaussian Mixture Model* (VBGMM)), se référer à Bishop (2006), chapitres 10 à 12.

1.8.1.5 Clustering évidentiel

Plusieurs extensions à la théorie probabiliste ont été proposées (Antoine, 2011) afin de mieux représenter les connaissances imparfaites. Une application au problème de clustering est rendue possible grâce à la notion de partition crédale. Cette notion est une application de la théorie des fonctions de croyance (théorie de Dempster-Shafer) et du modèle de croyances transférables (Smets et Kennes, 2008).

Ce modèle considère deux niveaux de raisonnement : le niveau crédale qui représente les connaissances incertaines et imprécises et le niveau pignistique pour la prise de décision à partir du niveau crédale. Une partition crédale issue d'un algorithme de clustering évidentiel (Denoeux et Masson, 2004; Masson et Denœux, 2008) permet de généraliser les notions de partitionnement dur, doux et probabiliste (voir sous-section 1.8.2), mais aussi et surtout, de représenter l'absence de prise de décision et l'exclusion d'un individu du problème. L'algorithme prend alors la forme d'un problème d'optimisation avec une fonction objectif à minimiser.

L'algorithme n'a pas encore fait l'objet d'application pour la reconstruction de

génomes mais des pistes sont explorées en ce sens (Antoine *et al.* (2018), annexe 7.4).

1.8.2 Incertitudes et contraintes

Le partitionnement crédale intègre intrinsèquement l'incertitude et l'imprécision des données, mais la prise en compte de l'incertitude du résultat peut également être ajoutée à des algorithmes existants. Ainsi, il est possible de ne pas attribuer un cluster à chaque individu mais plutôt de proposer une probabilité d'appartenance d'un individu à chacun des clusters. L'utilisateur peut alors choisir d'assigner un individu à un cluster, produisant ainsi un clustering dur, ou à plusieurs clusters, produisant ainsi un clustering doux.

Cette probabilité peut prendre la forme des distances d'un individu aux différents centres des clusters comme dans l'algorithme Fuzzy C-means (FCM), équivalent flou du k-means. La conception de ce flou quant à l'assignation d'un individu à un cluster est également simple à concevoir dans le cas d'un modèle de mélange gaussien : la variable latente discrète indiquant l'appartenance à une des composantes du mélange devient un vecteur de variables continues indiquant la probabilité d'appartenance de cet individu à chacune des composantes.

Les algorithmes de clustering présentés précédemment restent des modèles purement mathématiques, dont quelques paramètres (*e. g.* : nombre de clusters) sont à la charge de l'utilisateur. Selon les domaines d'application, il peut exister de forts *a priori* sur certaines relations que doivent entretenir des individus (Bair, 2013). Ces relations peuvent transparaître grâce à la modélisation des données lorsque celle-ci s'assure d'une faible distance entre des individus censés être similaires mais cela nécessiterait une supervision partielle de la modélisation des données. Cependant, cet *a priori* reste implicite pour l'algorithme de clustering qui peut, selon les données à traiter, ignorer cette relation. Si deux individus normalement liés se trouvent parmi un ensemble d'individus très différents d'eux, alors la relation a de grandes chances d'être respectée. À l'inverse, si ces deux individus se trouvent dans un ensemble d'individus extrêmement proches de l'un ou de l'autre, il est possible que la contrainte ne soit pas respectée.

Pour éviter cette influence de l'ensemble des individus, les relations connues *a priori* peuvent être représentées par des contraintes explicites. Les deux contraintes les plus répandues sont l'obligation pour deux individus d'être dans un même cluster – on parlera de relation « *must-link* » – ou l'interdiction d'être dans un même cluster – on parlera de relation « *cannot-link* ». Cet apport de connaissances *a priori* permet

une supervision partielle du clustering : on parlera alors de clustering semi-supervisé par contraintes.

Dans le cadre du binning, cette semi-supervision peut être apportée par des informations issues de la taxonomie des microorganismes ou par des pré-traitements spécifiques comme la fragementation des contigs. En effet, le contig d'origine de chaque fragment étant connu, celle-ci peut être utilisée comme une contrainte *must-link* sans besoin d'information extérieure.

Il existe un ensemble de méthodes de clustering capables de supporter des contraintes (Basu *et al.*, 2008; Bair, 2013) qui prennent la forme de liens obligatoires entre deux individus (contrainte *must-link*) ou l'obligation d'être dans deux clusters distincts (contrainte *cannot-link*). Ces contraintes peuvent être strictement imposées à l'algorithme ou celui-ci peut avoir la possibilité d'en violer si besoin. Ce comportement peut être souhaitable notamment pour corriger d'éventuelles erreurs dues à un assemblage chimérique. L'algorithme COP-KMEANS (Wagstaff *et al.*, 2001) s'assure du respect d'une contrainte avant l'assignation d'un individu à un des centres, rendant la violation de contrainte impossible. À l'inverse, l'algorithme CECM (Antoine *et al.*, 2012) intègre cette notion de contraintes à un clustering évidentiel, permettant la violation de contraintes.

1.8.3 Consensus clustering

Les méthodes de binning par agrégation de résultats d'autres méthodes de binning présentées en sous-section 1.3.4 repose sur une approche par consensus. DAS_Tool utilise en particulier des informations *a priori* (*i. e.* : gènes-marqueurs) pour la construction de ce consensus, et n'est donc pas *stricto sensu* une approche non supervisée.

Binning_refiner utilise la méthode du vote unanime. Plusieurs solutions sont proposées par plusieurs méthodes au problème du binning. Une solution est validée si toutes les méthodes s'accordent sur le résultat. Ainsi, si MetaBAT2, MaxBin2 et CONCOCT sur lesquels Binning_refiner s'appuie clusterisent tous deux contigs ensemble, cette association est validée. Autrement, ces contigs sont considérés comme non clusterisés.

Une variante de ce système de vote utilise un vote majoritaire. Un résultat est validé si un certain nombre de résultats (typiquement, 50 %) concordent. Cette approche par vote majoritaire n'a pas encore été employée pour la reconstruction de génomes à notre connaissance. Cette voie pourra alors être explorée à mesure que le

nombre de logiciels de binning augmente.

Les méthodes par consensus requièrent une diversité dans les approches sous-jacentes pour devenir performantes (Aggarwal et Reddy, 2013; Heisterkamp, 2015). Ces méthodes peuvent être construites par la variation de certains paramètres (*e. g.* : plusieurs k-means avec k variant) ou de la variation des approches globales. Dans le cas du binning non supervisé, la diversité des méthodes de clustering employées est importante tandis que la diversité des modélisations des contigs reste principalement limitée à deux modèles (voir tableau 1.1).

1.9 Objectifs de la thèse

L'analyse bibliographique des méthodes de reconstruction de génomes par les approches de binning non supervisé montre de nombreuses contibutions et un domaine en plein essor, notamment sur les approches par consensus. Les méthodes de binnning non supervisé peuvent devenir un maillon essentiel des pipelines d'analyse et de méta-analyse de métagénomes. L'objectif principal est donc de proposer une approche originale de binning non supervisé.

En effet, aucun consensus n'est encore établi sur ces approches et peu d'études comparatives sont disponibles et peuvent être considérées comme incomplètes par la communauté scientifique. Or, du fait des modèles de contigs employés, il n'est pas exclu que les logiciels soient fortement dépendants de la diversité analysée. Nous testerons donc, au travers d'une étude comparative, l'hypothèse que la qualité et la fiabilité des résultats du binning dépendent de la diversité en présence et des outils utilisés (chapitre 3).

De ces résultats, nous proposons de dissocier la modélisation des contigs et leur clustering en vue de faciliter le développement de nouvelles méthodes de binning. Nous introduisons trois modélisations de contigs qui n'ont jamais été testés dans le cadre du binning non supervisé : le dénombrement des graines espacées, le profil des distances inter-nucléotides et la proportion de séquences codantes. Un modèle original inspiré du traitement du langage naturel, Contig2Vec, est également proposé (chapitre 4). Nous faisons l'hypothèse que l'augmentation de la diversité des sources d'information par l'intégration de ces nouveaux modèles à ceux existants devrait permettre d'améliorer les résultats du binning.

Ainsi, pour tester cette hypothèse, nous exploiterons les capacités de ces nouveaux modèles couplés à une approche d'extraction de clusters pour reconstruire des génomes microbiens à partir de communautés artificielles définies dans ce travail mais aussi d'un métagénome dont la composition microbienne est inconnue (chapitre 5). Nos résultats seront comparés alors aux logiciels les plus fiables identifiés lors de l'étude comparative.

2 | Matériel et méthodes

Ce chapitre détaille le matériel et les méthodes utilisés pour étudier les résultats de binning, notamment dans le cadre de l'étude comparative du chapitre 3. Les éléments individuels pour la mise en place d'un nouvel algorithme de binning sont également décrits dans ce chapitre. L'implémentation des algorithmes reposant sur ces éléments est décrite dans le chapitre 4 dédié.

2.1 Jeux de données

Plusieurs jeux de données de tests et de validation sont construits pour l'évaluation de méthodes de binning nouvelles ou pré-existantes. Bien que de compositions différentes, tous sont traités de la même manière.

2.1.1 Données simulées pour visualisation : XS

Un premier métagénome est défini à partir de cinq génomes connus et complets, sélectionnés arbitrairement, à savoir *Methylotenera mobilis* (NC_012968.1 ; *Betaproteobacteria*), *Escherichia coli* (NC_000913.3 ; *Enterobacteriaceae*), *Klebsiella pneumoniae* (NC_016845.1 ; *Enterobacteriaceae*), *Enterobacter ludwigii* (NZ_CP017279.1 ; *Enterobacteriaceae*) et *Rhodococcus ruber* (NZ_LRRL01000001.1 ; *Actinobacteria*).

En effet, il est attendu que les 3 entérobactéries soient plus difficiles à reconstruire que les 2 non entérobactéries. Après fragmentation des génomes complets, 2394 fragments de contigs ont été générés (voir section 2.3).

2.1.2 Données simulées pour évaluation : S, M, L et $CAMI1h$

Trois métagénomes, S, M et L, ont été simulés à partir de respectivement 64, 170 et 350 génomes de référence selectionnés au hasard depuis la base de données RefSeq. S est constitué de 64 génomes bactériens. M contient 153 génomes bactériens, 2 archéens, 3 eucaryotes et 11 viraux. L contient 318 génomes bactériens, 3 archéens, 3 eucaryotes et 26 viraux. S et M ne sont composés que de chromosomes tandis que L intègre des séquences plasmidiques. Le détail de la composition des jeux de données (*e. g.* : numéros d'accession, taxonomie, abondance) sont disponibles à l'adresse suivante : `https://keuv-grvl.github.io/thesis_data/chap_02/` (Table S2).

Le critère majeur dans la définition d'un métagénome est l'inégale distribution des abondances des génomes. Cette distribution prend la forme d'un loi de puissance : nous pouvons systématiquement observer quelques espèces dominantes et de très nombreuses espèces de faible abondance (Lynch et Neufeld, 2015).

Ainsi, les abondances relatives des métagénomes S, M et L suivent toutes une loi de puissance qui vont de $1,21 \times 10^{-1}$ à $1,91 \times 10^{-5}$, de $3,48 \times 10^{-2}$ à $2,39 \times 10^{-13}$ et de $1,81 \times 10^{-2}$ à $8,75 \times 10^{-9}$ respectivement (Table S2).

Un total de 54 millions de lectures *paired-ends* pour chaque jeu de données mimant des données Illumina ont été simulés avec le logiciel Grinder (version 0.5.4) (Angly *et al.*, 2012), générant 10 milliards de nucléotides par jeux de données. Chaque

lecture a une longueur de 101 nucléotides avec une distance de 20 nucléotides (+/- 5) de sa lecture appariées. Les erreurs de séquençage suivent le modèle recommandés par Grinder. Les bases corrcetement lues ont un score de qualité Phred de 30 tandis que les bases incorrcetement lues ont un score de 10. Enfin, la couverture effective de chaque génomes varie de 380× à <1×, de 102× à <1× et de 60× à <1× pour les jeux de données S, M and L respectivement.

Ces trois jeux de données ont été assemblés individuellement à l'aide de l'assembleur *de novo* Ray Meta (Boisvert *et al.*, 2012) avec les paramètres par défaut.

Les résultats des binnings appliqués au jeu de données le plus complexe du premier défi CAMI ont été intégrés à l'étude comparative des outils de binning sous le label $CAMI1h$. Brièvement, ce jeu de données est composé de 5 échantillons simulant des données Illumina HiSeq 2×150 pb avec un insert de 270 pb $\pm$ 10% (taille totale : 75Gbp). Ces lectures simulées ont été co-assemblées en une référence (« *gold standard* ») composée de 39140 contigs (taille totale : 2.80 Gpb) avec l'assembleur *de novo* IDBA-UD (Peng *et al.*, 2012).

Les trois jeux de données simulés ici ont pour vocation d'étudier le biais d'une plus grande diversité (notamment par l'introduction de microeucaryotes, absents de CAMI) sur les résultats de binning, notamment en terme de contamination des procaryotes qui les plus représentés.

2.1.3 Données réelles du lac Pavin

En vue d'une application sur des données réelles, deux métagénomes générés à partir d'échantillons du lac Pavin ont été exploités (ANR EUREKA). Ces métagénomes ont été produits à partir d'eau extraite à deux profondeurs du lac Pavin, à savoir, 65 mètres et 80 mètres.

L'assemblage de ces données avec IDBA-UD a permis la reconstruction de 374597 et 91778 contigs pour ces deux profondeurs respectivement. Ces contigs ont été filtrés en fonction de leur taille (minimum de 1 kb) puis prétraités comme les jeux de données de test (voir section 2.3). 168135 fragments et 30430 fragments de séquences ont ainsi été produits respectivement pour les profondeurs de 65m et 80m.

2.2 Environnement de calcul et reproductibilité

Un des buts de l'étude comparative est d'apporter aux utilisateurs des informations quant aux besoins en terme de ressources de calcul. Les logiciels sont exécutés

dans un environnement aux capacités largement supérieures à une machine de bureau afin de leur permettre de fonctionner à leur plein potentiel. Ainsi, les logiciels sont exécutés sur une machine de calcul de type SMP sous CentOS 6.7 dotée de 10 CPU Intel Xeon E7-8870 (2.40GHz ; 160 threads) et de 1 To de mémoire vive. Les données d'entrée des logiciels sont stockées directement en mémoire vive pour éviter d'être pénalisées par les temps d'accès aux disques durs.

L'éxecution des logiciels est surveillée à l'aide du logiciel `GNU time` qui mesure le temps d'éxecution utilisateur (*wall clock time*), les consommations moyennes et maximales de mémoire vive, l'utilisation moyenne des CPU et le nombre d'accès au sytème de fichier. Pour faciliter la comparaison des performances des logiciels, chacun est configuré pour utiliser 8 CPU si possible, afin de se rapprocher d'un environnement de calcul de bureau.

Les logiciels ont été intégrés, si possible, à Bioconda (Grüning *et al.*, 2018) (voir Table S1) et ont été installés dans un environnement virtuel Conda dédié pour permettre la reproductibilité des analyses. Dans le cas contraire, les logiciels ont été installés dans un environnement virtuel dédié selon les instructions fournies.

La sélection des génomes connus, la simulation des lectures de séquençage, le binning des fragments de séquences et la comparaison des résultats ont été automatisés *via* des scripts bash, R et Perl pour permettre la comlète reproductibilité des analyses. Ceux-ci sont librement accessibles à l'adresse suivante : https://gitlab.com/keuv-grvl/cmp-mg-binning.

2.3 Prétraitement des données métagénomiques simulées

Les séquences (chromosomes entiers ou contigs issus d'un assemblage) sont fragmentées selon les recommandations de CONCOCT (Alneberg *et al.*, 2014). Cette fragmentation permet alors de représenter un long contig issu de l'assemblage (supposément fiable) par une collection de plusieurs fragments. Ainsi, ce long contig aura plus d'importance (puisque composé de plusieurs fragments de contigs) lors de la phase de clustering à venir.

Ces fragments sont définis par une fenêtre glissante de 10 kb avançant avec un pas de $6/7^{\text{ième}}$ de la taille de la fenêtre, créant ainsi de chevauchements de 1428 nucléotides. Si le dernier fragment d'une séquence est plus petit que la taille minimum autorisée (fixée ici à 1000 bases, comme la plupart des logiciels de binning utilisant

la composition des séquences), celui-ci est automatiquement accolé à l'avant-dernier fragment.

Suite à la fragmentation des longues séquences, l'origine des fragments est conservée dans une matrice binaire creuse représentant les liens « *must-link* » entre les fragments. Les séquences des fragments et les liens *must-link* sont enregistrés avec les différentes caractérisations des séquences pour assurer la reproductibilité de l'analyse.

Plusieurs méthodes de binning nécessitent la couverture des séquences pour fonctionner. Celle-ci a été calculée par alignement (« *mapping* ») des lectures sur les contigs fragmentés avec le logiciel bowtie2 (version 2.2.8) (Langmead et Salzberg, 2012) en utilisant l'alignement « *end-to-end* » et le jeu de paramètres « *very sensitive* ». Les fichiers BAM résultants ont été triés et indexés avec bamtools (Barnett *et al.*, 2011) pour une utilisation ultérieure.

2.4 Étude comparative des outils de binning existants

Afin de s'approcher du cas de la ré-analyse de données métagénomiques sans *a priori* en vue de reconstruire des génomes, plusieurs critères ont été fixés pour la sélection des outils à tester. Ces critères sont :

— Absence de dispositions expérimentales spécifiques pour permettre la ré-analyse de données publiques ;

— Absence de données de référence pour conserver le caractère non supervisé de l'approche ;

— Modélisation des séquences incluses dans l'outil ;

De plus, les critères suivants ont été considérés :

— Le logiciel doit être dans l'environnement de test ;

— Les résultats doivent être obtenus en moins de 7 jours ;

— Seuls les paramètres par défaut sont utilisés, ou les paramètres recommandés le cas échéant ;

— Le logiciel doit pouvoir être utilisé par des non-spécialistes (*e. g.* : une documentation est founie).

Parmi les méthodes de binning recensées en section sous-section 1.3.4, seuls les outils COCACOLA (Lu *et al.*, 2017), CONCOCT (Alneberg *et al.*, 2014),

DAS_Tool (Sieber *et al.*, 2018), MaxBin (Wu *et al.*, 2016), MetaBAT (Kang *et al.*, 2015), MetaBAT2 (Kang *et al.*, 2015) et MetaProb (Girotto *et al.*, 2016) ont satisfait les critères de sélection pour l'étude comparative (voir Annexe A.5).

Chaque outil selectionné propose une combinaison unique entre la modélisation des données et la stratégie de clustering (tableau 1.1).

2.4.1 Comparaison des résultats de binning de métagénomes simulés

L'évaluation des résultats de binning pour des métagénomes dont la composition est connue est faite selon deux méthodes : par des métriques d'évaluation supervisée et par profilage des résultats des logiciels à différents niveaux taxonomiques.

2.4.1.1 Métriques d'évaluation

La comparaison de résultats de binning en présence d'un résultat attendu revient à une évaluation supervisée puisque les bins de référence sont connus et représentent le résultat attendu.

Pour permettre le calcul des métriques (voir sous-section 1.4.1) et ainsi comparer les résultats, on associe d'abord chaque bin produit – ici considéré comme des cluters – à un bin de référence – ici considéré comme des classes –, en considérant optionnellement les fragments non traités.

Chaque cluster est ainsi associé à la classe avec laquelle il a le plus de fragments de contigs partagés. Tous les fragments issus de la classe X sont considérés comme appartenant au cluster Y si ceux-ci sont associés. Ainsi, un cluster peut contenir plusieurs classes de la même manière qu'un bin peut contenir plusieurs génomes (et donc représenter un génome composite ou un groupe taxonomique).

Cette association entre classes et clusters permet alors l'énumération des vrais positifs, faux positifs, vrais négatifs et faux négatifs, et ainsi la construction d'une matrice de confusion.

Les fragments de contigs non traités influent sur les valeurs de ces métriques et leur simple élimination peut biaiser l'évaluation. Ainsi, nous distinguons le cas particulier où l'on souhaite prendre en compte ces fragments de contigs non traités où ils seront dénombrés comme des faux négatifs.

Les métriques de classification binaire (sensibilité, spécificité, précision, fiabilité et score F1) sont calculées pour chaque bin et moyennées. Ces valeurs sont alors nommées "métriques de binning" (*e. g.* : "précision du binning", "fiabilité du binning").

Puisque nous cherchons à évaluer la reconstruction des génomes, les séquences non traitées (« *unbinned* »), lorsqu'elles sont considérées, le sont comme faussement regroupées (elles sont donc toutes des faux positifs d'un bin fictif).

2.4.1.2 Profilage d'un résultat de binning

Les bins de référence sont utilisés pour énumérer les fragments de contigs qui ont été correctement traités ou non par les différents outils de binning. Ainsi, chaque fragments est labélisé "0" ou "1" si le logiciel de binning considéré a correctement binné ce fragment (Table S3). Chaque fragment est labellisé pour chaque logiciel. On obtient ainsi un profil qualitatif pour chaque logiciel, ce profil étant un vecteur de valeurs binaires.

Un profil artificiel représentant un contrôle négatif, labellisé « *negative* », est ajouté en plus des logiciels pour représenter le cas où le binning serait systématiquement faux.

La matrice des distances entre paires de profils est alors construite à l'aide d'une distance L1 (distance Manhattan). Cette matrice de distance est alors utilisée pour représenter graphiquement les distances entre les outils à l'aide d'une Analyse en Coordonées Principales (PCoA). Les métriques précédemment citées sont de plus mappées comme des variables environnementales avec le paquet R vegan (Oksanen *et al.*, 2017). DAS_Tool étant une méthode aggrégative, ces résultats n'ont pas été inclus dans cette analyse.

L'origine taxonomique de chaque fragment étant connue, le taux de fragments correctement binnés est alors calculé pour chaque taxon à partir (Table S3). Cette analyse taxonomique vise à être plus stricte vis-à-vis des métriques d'évaluation du binnning. En effet, le taux de fragments correctement binnés (nombre de vrais positifs divisé par le nombre de contigs du cluster) est une version exacte (et non estimée) du taux de complétude de CheckM (Parks *et al.*, 2015).

2.4.2 Comparaison des résultats de binning de métagénomes non simulés

En l'absence de résultat de référence, les résultats de binning proposés sont comparés en se reposant sur l'estimation de la parenté entre les paires de bins. Le calcul des valeurs d'ANI est assuré par le logiciel pyANI (version 0.2.4) (Pritchard *et al.*, 2016) qui propose une mesure de l'ANI *via* le programme

`average_nucleotide_identity.py` en utilisant `blastn` de la suite BLAST+ 2.6.0 *via* le paramètre `-m ANIb`.

Deux bins sont considérés comme appartenant à la même espèce si l'ANI entre ces deux bins est supérieur à 95 % (Richter *et al.*, 2008). Ceux-ci sont alors liés par une flèche dans un graphe orienté. En effet, l'ANI entre deux génomes n'est pas nécessairement réciproque.

Une visualisation des graphes d'ANI est construire par le paquet R qgraph (Epskamp *et al.*, 2012).

Les métriques d'évaluation des bins sont ici obtenues grâce au workflow `lineage_wf` proposé par CheckM en utilisant l'arbre taxonomique réduit (option `-reduced_tree`) puis représentées graphiquement grâce au package R lattice (Sarkar, 2008).

2.4.3 Relation entre qualité des bins et logiciels de binning

D'autre part, l'étude de l'influence du choix du logiciel sur la qualité des MAG produits est réalisée sur le jeu de données L uniquement (le plus complexe). Pour ce faire, les matrices de complétude et de contamination de chaque génome pour chaque logiciel sont d'abord estimées par CheckM.

Les bins sont ensuite répartis en quatre classes indiquant leur qualité (voir sous-section 1.5.3). Le dénombrement des MAG par classe de qualité est ensuite utilisé pour tester l'indépendance entre qualité des bins et logiciels par ANOSIM (*via* R). Ce test statistique est effectué pour tous les niveaux taxonomiques.

2.5 Méthode de modélisation intégrative des contigs pour le binning

2.5.1 Implémentation des modèles bruts

2.5.1.1 Abondance des séquences

L'abondance moyenne des séquences est estimée par le logiciel GATTACA (Popic *et al.*, 2018) avec les paramètres par défaut ($k = 31$). Une matrice d'abondance est alors obtenue.

2.5.1.2 Composition en k-mers

La composition des séquences repose sur le dénombrement des k-mers et des graines espacées. Ces k-mers sont représentés par des masques pour offrir une représentation commune aux deux modèles.

Un masque sans position ignorée correspond à un k-mer classique (*e. g.* : le masque « 1111 » représente un tétramère). À l'inverse, le masque « 11011 » indique que le troisième nucléotide sera ignoré.

Du fait de l'utilisation des k-mers canoniques, les masques doivent être palindromiques. Les positions ignorées sont représentées par le caractère « x » afin de les différencier des véritables bases encodées par la nomenclature de l'*International Union of Pure and Applied Chemistry* (IUPAC) pour les nucléotides (principalement A, G, T, C et N). Le choix du masque est laissé à la discrétion de l'utilisateur.

L'extraction des k-mers ne nécessite qu'une seule lecture de chaque séquence, permettant donc une complexité temporelle linéaire fonction du nombre de séquences à caractériser. La parallélisation de l'extraction de ces caractéristiques est faite sur les séquences et devient donc efficace lorsque qu'il y a davantage de séquences que de cœurs de calcul.

2.5.1.3 Profil des distances inter-nucléotides

Ce modèle est le résultat de la concaténation des distances entre les paires des nucléotides et des plus proches nucléotides dissimilaires. Un profil de distances inter-nucléotides est construit en mesurant un certain nombre K de distances entre les paires des nucléotides et des plus proches nucléotides dissimilaires. Ce paramètre K est fixé à 15 par défaut conformément aux expérimentations de Xie *et al.* (2017).

La méthode a été ré-implémentée avec une complexité temporelle linéaire dépendante du nombre de séquences à traiter puisque chaque séquence ne doit être parcourue qu'une seule fois.

2.5.1.4 Densité de séquences codantes

La densité de séquences codantes nécessite une prédiction des séquences codantes. Pour rester dans une approche non supervisée, celles-ci sont prédites *de novo* à l'aide des logiciels Prodigal (Hyatt *et al.*, 2010), FragGeneScan (Rho *et al.*, 2010) et MetaGeneAnnotator (Noguchi *et al.*, 2008). On obtient alors autant d'attributs pour chaque séquence à caractériser qu'il y a de logiciels de prédiction de séquences codantes appliqués.

2.5.1.5 Contig2Vec

La modélisation par dna2vec a été ré-entrainée à partir de 3000 génomes microbiens sélectionnés au hasard dans la banque de données RefSeq (O'Leary *et al.*, 2016), représentant au total 58,4 Gb. La taille des k-mers varie de 4 à 6 bases ; la taille du contexte est de 10 mots avant et après ; 10 lectures de chaque séquence sont réalisées (*dna2vec epochs*) ; le modèle Word2vec sous-jacent est itéré 3 fois ; le vecteur final représentant chaque k-mer a 100 attributs. L'entraînement de ce modèle a duré un temps CPU d'environ 700h et produit un modèle de 3,4 Mo.

Le modèle dna2vec étant contenu dans un unique fichier, la parallélisation de la modélisation des séquences reste limitée par les accès concurrents au système de stockage, bien que la compléxité temporelle reste linéaire en fonction du nombre de séquences.

2.5.2 Intégration des modèles bruts

Chaque modélisation des fragments de contigs produit une matrice avec les contigs en ligne et les attributs en colonne. Chacune est alors traitée à l'aide d'une Analyse en Composantes Principales (ACP) à noyau. Pour ce faire, la distance cosinus est utilisée. Les composantes produites sont alors filtrées en fonction de la variance de manière à conserver 85 % de l'inertie totale. Ainsi, on conserve approximativement autant d'attributs que ce que propose CONCOCT.

L'intégration de ces différents espaces des caractéristiques est alors assurée par une ACP après avoir centré et réduit chacun des espaces et les avoir juxtaposés horizontalement. Le résultat de cette projection est alors filtré pour garder 99 % de l'inertie du jeu de données. Le résultat de l'ACP devient alors la modélisation finale des fragments de contigs.

2.5.3 Traçabilité des données

Les données produites (séquences fragmentées, matrice *must-link* et modèles bruts des séquences) ainsi que les métadonnées associées (nom du fichier d'origine, longueur minimum des séquences, longueur maximum des séquences, longueur des fragments et taille des chevauchements entre fragments) sont stockées dans des conteneurs de fichiers HDF5 (HDF Group, 1997).

Ce format de fichier présente les avantages (i) de pouvoir représenter une quantité virtuellement illimitée de données ; (ii) de permettre de représenter n'importe quel

type de données scientifiques (des matrices et des chaines de caractères dans notre cas) ; (iii) de permettre la présence de métadonnées ; (iv) d'être indépendant de la plateforme ou du language utilisé ; (v) de représenter les données de façon à rendre leur lecture efficace ; (vi) de pouvoir hiérachiser les données en « *datasets* » (ici, des matrices) et « *groups* » (assimilables à des dossiers) ; (vii) d'être morcelable, et donc compatible avec des environnements de calcul distribué comme Apache Spark et (viii) d'être compatible avec l'environnement d'apprentissage automatique de Python.

2.5.4 Visualisation des contigs modélisés

La visualisation des contigs sous forme d'un nuage de points bidimensionnel (« *scatterplot* ») est réalisée grâce à l'algorithme BH-tSNE à partir de la modélisation multidimensionnelle des contigs, et plus particulièrement avec l'implémentation parallèle proposée par Ulyanov (2016).

2.5.5 Contribution de chaque modèle brut à la modélisation intégrée

La contribution de chaque modèle brut de données à la modélisation finale intégrée est évaluée par la recherche de corrélations de Pearson entre les composantes principales de la modélisation intégrée et chacun des attributs de chacun de modèles bruts. Ces contributions sont représentées par la densité des valeurs absolues des coefficients de corrélation. Seules les corrélations ayant une p-value inférieure à 1×10^{-6} sont considérées. Les représentations graphiques de ces densités sont réalisées avec le package R lattice (Sarkar, 2008).

2.6 Clustering

2.6.1 Clustering consensuel appliqué au binning de contigs

Le potentiel du consensus clustering (Ghosh et Acharya, 2011; Bonet *et al.*, 2017) a été étudié en utilisant le profilage des résultats de binning. Un fragment de contigs est alors validé si un certain nombre d'outils l'a consensuellement clusterisé. Autrement, ce fragment est considéré comme non traité.

2.6.2 Prétraitement des données

Le jeu de données de test a été caractérisé selon sept modèles différents : Contig2Vec avec $k = 4$ puis $k = 6$, tous deux en utilisant le modèle fourni par défaut, notés « contig2vec4 » et « contig2vec6 » ; le profil des distances inter-nucléotides avec $K = 15$ noté « ind15 » ; les dénombrements des k-mers en utilisant les masques « 1001001 », « 110011 », « 111 » et « 1111 », notés « kmers1001001 », « kmers110011 », « kmers3 », « kmers4 ». Les graines espacées ont été choisies arbitrairement de façon à ne pas produire un trop grand nombre d'attributs (ici, 64 et 136 respectivement). Tous ces modèles de données produisent alors un total de 836 attributs et sont synthétisé dans le tableau 2.1.

Tableau 2.1 | **Modèles de séquences de `fennec` appliqués par défaut.**

Modèle de séquences	Paramètres	Dénomination interne
MaskedKmer	`mask="110011"`	`kmers110011`
MaskedKmer	`mask="11000100011"`	`kmers11000100011`
MaskedKmer	`mask="1001001"`	`kmers1001001`
MaskedKmer	`mask="111"`	`kmers3`
MaskedKmer	`mask="1111"`	`kmers4`
InterNucleotideDistance	`K=15`	`ind15`
Contig2Vec	`k=4`	`contig2vec4`
Contig2Vec	`k=6`	`contig2vec6`

Puisque le jeu de données XS est uniquement composé de génomes complets et non de lectures de séquençage, l'utilisation de l'abondance moyenne des séquences n'est pas envisageable. De plus, pour évaluer l'adaptabilité de la modélisation des données, le jeu de données XS a été manuellement subdivisé en plusieurs sous-ensembles de séquences correspondants chacun (i) aux cinq génomes du jeu de données ; (ii) aux trois génomes des entérobactéries ; (iii) aux deux génomes qui ne sont pas des entérobactéries ; et (iv) au génome actinobactérien seul.

2.6.3 VBGMM semi-supervisé

L'algorithme de clustering VBGMM a été modifié pour prendre en compte les contraintes *must-link* apportées par la fragmentation des contigs. L'initialisation de l'algorithme était originellement réalisée à partir d'un k-means avec k le nombre

de composantes maximal du modèle de mélange (400 par défaut dans CONCOCT). Cette initialisation a été modifiée de façon à respecter les contraintes *must-link*. Chaque ensemble de fragments originaires d'un même contig est ajouté au hasard à une des composantes initiales de VBGMM. Le reste de l'algorithme reste inchangé, lui permettant ainsi de violer des contraintes lors des étapes d'E-M puisque l'algorithme reste libre de l'assignation des fragments de contigs aux composantes.

L'algorithme a été porté pour être utilisable par Python 3 afin d'être utilisable aux côtés de la boîte à outils puis modifié de façon à pouvoir choisir le nombre d'exécutions parallèles de l'algorithme (originellement fixé à 10). Ce paramètre ne peut cependant être réglé qu'à l'installation du logiciel et non avant chaque exécution. Le clustering par l'algorithme VBGMM prend la forme de l'optimisation d'un objectif : celui-ci sera exécuté parallèlement 32 fois (par défaut) indépendamment de façon à sélectionner le meilleur résultat. Chacune de ses exécutions aura alors pour paramètres par défaut une amélioration minimum de la fonction objectif de 1×10^{-4} (paramètres « epsilon ») avec un nombre maximum de 600 itérations d'E-M (paramètre « maxiter »). Par défaut, l'initialisation des composantes est la même que CONCOCT, à savoir l'application d'un k-means, mais peut être paramétrée pour utiliser les relations *must-link* (paramètre « init_type »).

2.6.4 Post-traitement

Les contraintes *must-link* sont également prises en compte lors d'une étape de post-traitement des résultats de clustering. Cette étape consiste à identifier les clusters de petite taille. Par défaut les clusters doivent contenir au moins 50 fragments de contigs. Les fragments de contig ne dépassent pas 10 kb par défaut.

Un cluster de moins de 50 fragments de contigs serait alors composé au maximum de 500 kb, soit approximativement la taille des plus petits génomes connus. En effet, le plus petit génome bactérien non symbiotique est celui de Mycoplasma genitalium pour une taille de 580 kb (Razin et Hayflick, 2010). Le plus petit symbiote obligatoire bactérien connu est *Nanoarchaeum equitans* avec un génome de 490 kb (Waters *et al.*, 2003). Avec cette contrainte, des génomes très incomplets (moins de 500 kb) sont automatiquement éliminés.

Une fois identifiés, les fragments de contigs composant ces clusters sont alors ré-assignés au cluster avec lequel ils partagent le plus de relations *must-link*. Si ce fragment ne possède pas de relation *must-link* , il est alors considéré comme non clusterisés et assigné à un cluster fictif numéroté `-1`, par opposition aux autres clusters

(voir sous-section 4.2.2). Cette étape du processus utilise le clustering précédent comme information d'entrée mais également les relations *must-link* et propose une modification du clustering en sortie ainsi qu'une visualisation des données sous la forme d'un nuage de points.

2.6.5 Extraction automatique des clusters

Une fois le clustering réalisé et post-traité, le score *silhouette* de chaque fragment de contigs est calculé à l'aide de de la bilbiothèque scikit-learn à partir d'une modélisation des contigs et du résultat de clustering de ceux-ci, ainsi que le score médian et la distribution des scores pour chaque cluster. Un cluster est alors extrait du jeu de données si (i) la valeur du premier quartile de ses scores *silhouette* est supérieure à la médiane globale et si (ii) la valeur du dernier décile est supérieure à 0. On évite ainsi l'extraction des clusters qui pourraient être chevauchants ou pour lesquels la modélisation n'a pas nécessairement produit les attributs les plus discriminants par rapport à l'ensemble de données traité. Les clusters ainsi extraits sont alors enregistrés et écartés du jeu de données à traiter.

3 | Résultat de l'étude comparative

3.1 Ressources informatiques consommées

Le temps, l'utilisation maximale de la mémoire, l'utilisation moyenne des processeurs (CPU) et l'accès au système de fichiers sont mesurés pendant notre étude comparative (tableau 3.1).

Le temps d'exécution varie de 6 minutes pour MetaBAT à 6 heures et 5 minutes en moyenne pour COCACOLA. La plupart des exécutions ont duré moins de 2 heures et 30 minutes. L'utilisation maximale de la mémoire vive (RAM) allait de 270 Mo pour MetaBAT en utilisant l'information sur l'abondance à 26 Go pour MetaProb ; la plupart des outils utilisant moins de 4,5 Go de RAM. L'utilisation moyenne du CPU varie de 142 % pour MaxBin à 5671 % pour MetaProb.

Une utilisation optimale du CPU dans les conditions de l'étude comparative serait de 800 % puisque nous avons mis en place des outils pour utiliser 8 CPU (soit une utilisation optimale du CPU de 800 %). À cet égard, MetaBAT et COCACOLA parallélisent efficacement leur exécution avec une utilisation moyenne du CPU de 690 % à 792 %, respectivement. CONCOCT a également efficacement parallélisé (968 % du CPU est utilisé en moyenne) mais il n'est pas possible de choisir le nombre de threads (fixé à 10) sans recompiler le code source. MetaBAT (avec abondances) et MaxBin exploitent peu la parallélisation avec une utilisation moyenne du CPU allant de 142 % à 381 % alors que d'autres logiciels employant les méthodes (k-means et E-M, respectivement) y parviennent.

Les deux exécutions de MetaBAT2 montrent une parallélisation efficace avec des jeux de données plus importants puisque l'utilisation moyenne du CPU variait de 443 % à 477 % du jeu de données S et M mais a augmenté à 720 % pour le jeu de données L. MetaProb s'adapte à son environnement et utilise de 4199 % à 5671 % CPU en moyenne alors qu'il est l'un des logiciels les plus longs à s'exécuter. La majeure partie des calculs de DAS_Tools est effectuée par Prodigal et BLAST, qui exploitent correctement la parallélisation (utilisation moyenne du CPU : 761 % ; utilisation moyenne de la mémoire : 171 Mo). Bien que DAS_Tool ait pris de 11 à 30 minutes à exécuter, il a besoin des résultats d'autres logiciels de binning. Son temps d'exécution ne reflète donc pas l'ensemble du processus de binning.

On peut remarquer que la plupart des outils peuvent être exécutés avec 8 Go de RAM à partir des contigs fragmentés (96 à 285 Mo en format Fasta) et des mappings lus (3,4 à 3,9 Go en format BAM) pour produire un résultat en une nuit.

Tableau 3.1 | **Vue d'ensemble des performances moyennes des logiciels évalués**. Le temps total, la consommation de mémoire vive, le temps par CPU (temps d'éxecution théorique quand 1 CPU est utilisé) et les métriques de binning moyennes (± écart-type) sont donnés. Les logiciels marqués d'une astérisque intègrent les résultats de AMBER dans le calcul des métriques.

Logiciels	**Temps d'éxecution** (h:m:s)	**Mémoire maximum** (MB)	**Utilisation CPU moyenne** (%)	**Temps par CPU** (h:m:s)	**Séquences traitées** (%)	**Précision moyenne** (%)	**Fiabilité moyenne** (%)
COCACOLA *	06:05:21 (± 04:26:06)	4486 (± 2690)	792 (± 153)	44:13:00 (± 30:00:15)	100 (± 0)	76 (± 4)	95 (± 0)
CONCOCT *	01:25:39 (± 01:13:28)	1640 (± 851)	686 (± 131)	10:46:20 (± 09:22:20)	100 (± 0)	89 (± 5)	99 (± 0)
DAS_Tool *	00:21:11 (± 00:06:25)	171 (± 57)	761 (± 2)	02:41:07 (± 01:11:17)	34 (± 28)	23 (± 21)	35 (± 28)
MaxBin *	02:36:42 (± 01:47:33)	595 (± 135)	247 (± 122)	05:13:56 (± 01:16:37)	97 (± 1)	76 (± 9)	92 (± 3)
MetaBAT	00:01:42 (± 00:01:37)	596 (± 394)	748 (± 16)	00:12:53 (± 00:12:28)	55 (± 5)	42 (± 7)	45 (± 8)
MetaBAT * (avec abondances)	00:06:32 (± 00:02:58)	498 (± 310)	280 (± 65)	00:18:40 (± 00:11:32)	54 (± 5)	43 (± 8)	45 (± 7)
MetaBAT2	00:06:22 (± 00:04:31)	1163 (± 513)	546 (± 151)	00:34:59 (± 00:24:11)	53 (± 5)	84 (± 4)	74 (± 4)
MetaBAT2 * (avec abondances)	00:06:01 (± 00:03:58)	1203 (± 541)	549 (± 147)	00:33:32 (± 00:22:39)	57 (± 6)	44 (± 7)	45 (± 7)
MetaProb *	01:09:32 (± 00:29:06)	16063 (± 9408)	4934 (± 736)	59:10:21 (± 29:42:17)	100 (± 0)	84 (± 5)	97 (± 1)

3.2 Évaluation des résultats de binning excluant les contigs non traitées

Les résultats d'AMBER sont ajoutés à notre étude comparative pour décrire une vue d'jeu de la performance des logiciels de binning. La spécificité du binning varie entre 0,9953 et 1,0 (Tableau S5A), ce qui indique une très bonne capacité à éviter les faux négatifs dans les bins en général. Cependant, la sensibilité du binning varie de 0,0129 à 0,5973, principalement parce que les faux positifs sont plus nombreux que les vrais positifs. Le taux de fausses découvertes varie de 0,0 à 0,0047, montrant très peu de faux positifs parmi les bins. La précision du binning varie de 0,7109 à 0,9991 et de 0,5650 à 1,0, où des valeurs inférieures indiquent une forte proportion de faux positifs par rapport aux vrais positifs.

Comme la précision du binning et la fiabilité du binning ne sont pas corrélées, elles sont résumées en figure 3.1. Les bins de référence (carrés noirs) ont une précision de 1,0 et une fiabilité de 1,0. Le jeu de données $CAMI1h$ affiche une meilleure précision moyenne que les jeux de données S, M et L. Cela pourrait s'expliquer par sa conception (cinq répétitions). CONCOCT et MetaBAT2 (incluant les données d'abondance) affichent les meilleures performances.

Étonnamment, DAS_Tool ne traite que 248 contigs (1,6 %) du jeu de données S. Pour tous les jeux de données, les deux versions de MetaBAT tenant compte ou non de l'abondance traitent 48 % à 67 % des contigs d'entrée tandis que les autres logiciels de binning traitent en moyenne 95 % des contigs d'entrée (tableau 3.1).

Bien qu'imparfaits, les bins reconstruits sont proches des bins attendus pour chaque jeu de données. Cependant, un grand nombre de contigs n'est pas traité par certains outils alors qu'ils pourraient être porteurs d'informations biologiques nécessaires à l'étude de ces génomes.

3.3 Évaluation des résultats de binning incluant tous les contigs

Si on s'attend à ce qu'un logiciel de binning regroupe tous les contigs d'entrée, on choisira de considérer les contigs non traités comme de faux positifs ou faux négatifs. Ainsi, la sensibilité, la précision et l'exactitude globales du binning sont dégradées. En effet, la sensibilité du binning varie de 0,0003 à 0,5973, la précision de 0,0153 à 0,9220 et la précision de 0,0599 à 0,9961 (figure 3.2, Table S5B).

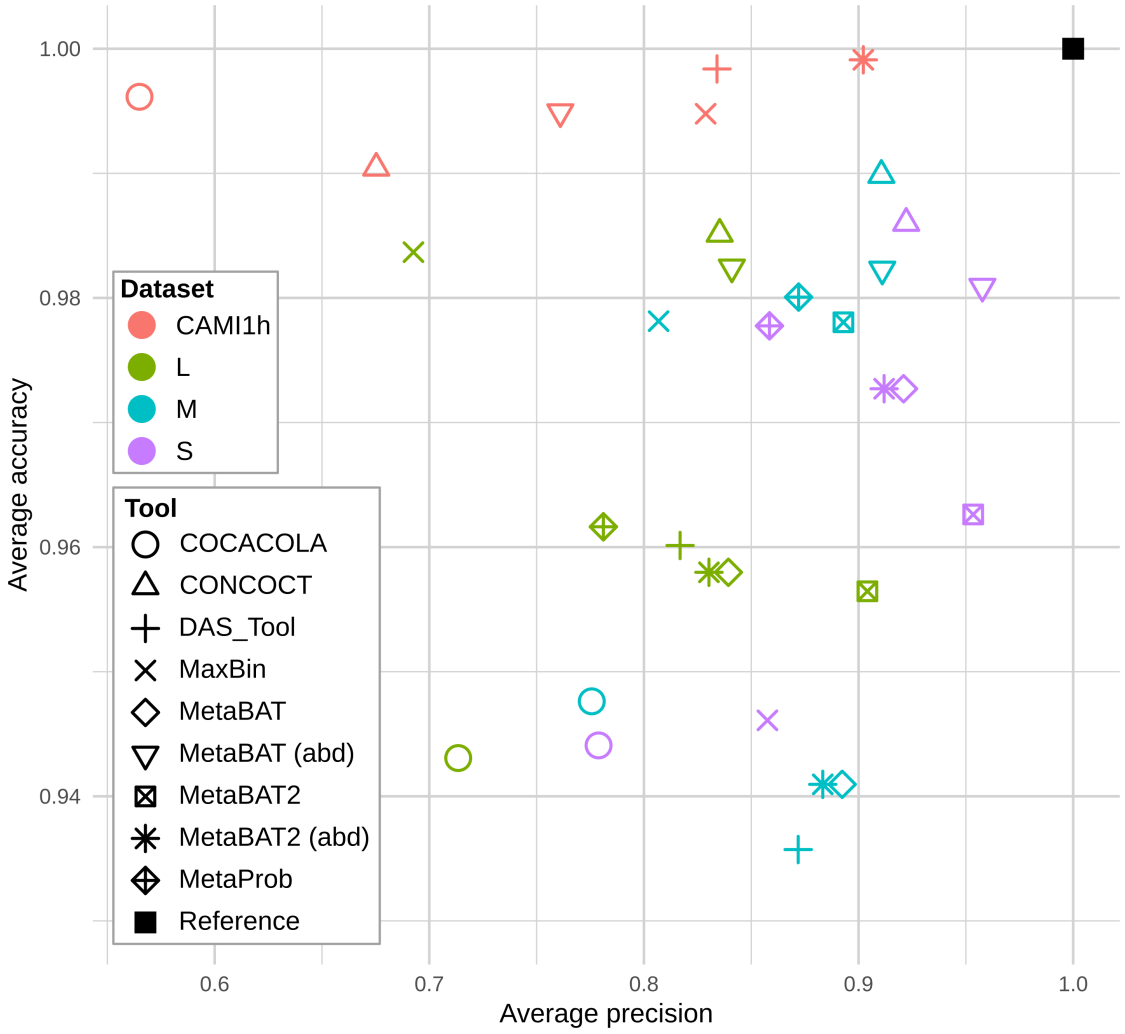


Figure 3.1 | **Précision et fiabilité du binning sans tenir compte des contigs non traités**. Les résultats DAS_Tool du jeu de données S ne sont pas représentés ici (précision moyenne : 1,0, fiabilité moyenne : 0.74). Les logiciels marqués en italique sont exécutés sur le jeu de données $CAMI1h$ uniquement.

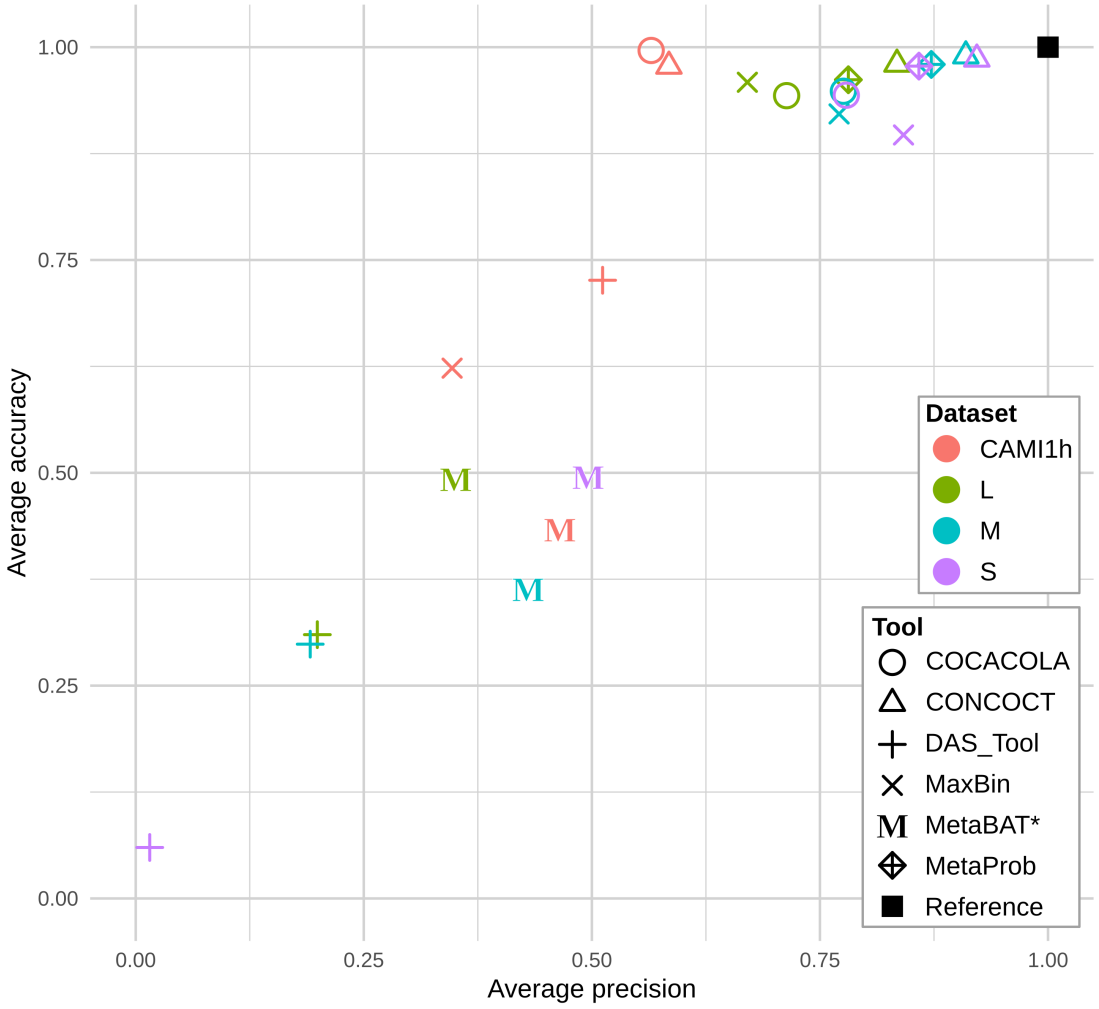


Figure 3.2 | **Précision et fiabilité moyennes lorsque les résultats du binning considèrent les contigs non traités comme faux**. Les résultats des deux versions de MetaBAT, considérant ou non l'abondance des contigs, sont regroupés sous le symbole « `M` » car quasiment identiques. Les logiciels marqués en italique ne sont exécutés que sur le jeu de données *CAMI1h*.

Tableau 3.2 | **Dénombrement des bins pour chaque logiciel**. Les bins sont distribués en quatre classes selon les critères de Bowers *et al.* (2017) : haute qualité (HQ), qualité médium (MQ), qualité basse (LQ) ou "Écarté" quand les bins ne satisfont aucun des critères.

	Écarté	**LQ**	**MQ**	**HQ**
COCACOLA	116	101	6	0
CONCOCT	110	69	29	15
DAS_Tool	114	98	10	1
MaxBin2	152	47	22	2
MetaBAT	107	111	3	2
MetaBAT avec abondance	96	110	14	3
MetaBAT2	93	114	16	0
MetaBAT2 avec abondance	107	111	3	2
MetaProb	144	64	13	2

Les résultats des deux versions de MetaBAT, considérant ou non l'abondance des contigs, sont pénalisés pour tous les jeux de données en raison de la faible proportion de contigs traités même si les bins proposés étaient fiables en moyenne. Si l'on considère les contigs non traités pour tous les jeux de données, la sensibilité de MetaBAT varie de 0,0007 à 0,0199 (0,0831 à 0,5132 sans contigs non traités) et sa précision varie de 0,3468 à 0,5030 (0,7611 à 0,9577 sans les contigs non traités) alors que sa précision varie de 0,3557 à 0,4987 (0,9410 à 0,9991 sans les contigs non traités). En tant que méthode d'agrégation, DAS_Tool peut être pénalisé si l'une de ses entrées propose un résultat très différent des autres entrées. Cela pourrait expliquer ses très faibles scores. CONCOCT et BinSanity sont les méthodes les plus performantes si l'on veut explorer le maximum de contigs.

3.4 Évaluation des logiciels de binning sur la base de la complétude et de la contamination

Nous notons que peu de bins de haute qualité sont proposés pour les jeux de données de test (tableau 3.2).

La relation estimée par ANOSIM entre les logiciels et la qualité des bins est très

Tableau 3.3 | **Dénombrement des bins par type de qualité par logiciel et par domaine.**. Le jeu de données L est composé de trois domaines : bactérien (B), eucaryote (E) et viral (V). Les bins sont distribués en quatre classes selon les critères de Bowers *et al.* (2017) (voir tableau 1.2) : haute qualité (HQ), qualité médium (MQ), qualité basse (LQ) ou "Écarté" quand les bins ne satisfont aucun des critères.

	Écarté			**LQ**			**MQ**			**HQ**		
	B	**E**	**V**	**B**	**E**	**V**	**B**	**E**	**V**	**B**	**E**	**V**
COCACOLA	108	3	5	85	11	5	6	0	0	0	0	0
CONCOCT	103	6	1	52	8	9	29	0	0	15	0	0
DAS_Tool	101	10	3	87	4	7	10	0	0	1	0	0
MaxBin2	145	2	5	30	12	5	22	0	0	2	0	0
MetaBAT	99	8	0	95	6	10	3	0	0	2	0	0
MetaBAT avec abondance	87	9	0	95	5	10	14	0	0	3	0	0
MetaBAT2	84	9	0	99	5	10	16	0	0	0	0	0
MetaBAT2 avec abondance	99	8	0	95	6	10	3	0	0	2	0	0
MetaProb	140	2	2	44	12	8	13	0	0	2	0	0

significative ($p - value = 2,2e - 16$) et nous a permis de rejeter l'hypothèse nulle d'indépendance entre la qualité des bins et les logiciels. Le logiciel a donc un effet sur la qualité des bins produits.

Plus précisément, pour les bactéries, eucaryotes et virus (tableau 3.3), les bins de haute qualité sont obtenus uniquement à partir de génomes bactériens alors que les génomes eucaryotes et viraux étaient plutôt considérés comme bins de mauvaise qualité. Les bins eucaryotes de faible qualité sont principalement calculés par COCACOLA, MaxBin2 et MetaProb, tandis que les bins représentant les virus de faible qualité sont proposés par MetaBAT, CONCOCT et MetaProb, les autres logiciels ayant un nombre similaire de bins de faible qualité et de bins non considérés.

3.5 Efficacité des méthodes de binning à différents niveaux taxonomiques

Les tests ANOSIM sont significatifs (p-value $< 0,05$) pour les rangs taxonomiques allant de la classe à la sous-espèce pour la complétude et la contamination

Tableau 3.4 | **Relation entre les taux de complétude et de contamination groupés par rangs taxonomiques et les logiciels de binning qui les ont produits.**Une significativité (p-value) inférieure à 0,05 nous permet de rejeter l'hypothèse nulle stipulant que la similarité entre les groupes est supérieure ou égale à la similarité interne au groupe. Une forte discrimination est observée pour une statistique R entre 1,0 et 0,5 tandis que des chevauchements sont observés pour une statistique R entre 0,5 et 0,1. L'influence du critère de regroupement (ici, le niveau taxonomique) n'est pas négligeable pour une statistique R inférieure à 0,1.

Rang taxonomique	**Complétude**		**Contamination**	
	Statistique R	**p-value**	**Statistique R**	**p-value**
Domaine	0.041	0.083	0.204	0.001
Phylum	0.000	0.469	0.046	0.079
Classe	0.098	0.001	0.093	0.001
Ordre	0.179	0.001	0.160	0.001
Famille	0.255	0.001	0.223	0.001
Genre	0.424	0.001	0.384	0.001
Espèce	0.538	0.001	0.448	0.001
Sous-espèce	0.538	0.001	0.448	0.001

(tableau 3.4). Cependant, les statistiques vont de 0,098 à 0,538 et de 0,093 à 0,448 respectivement pour la complétude et la contamination, montrant une discrimination moyenne entre taxons. Ainsi, la complétude et la contamination ont tendance à dépendre du logiciel lorsqu'on se concentre sur les niveaux taxonomiques les plus fins.

La discussion de ces résultats est regroupée avec les autres discussions en chapitre 6 (page 131).

Il est à noter que l'influence du choix du logiciel pourrait être limitée par l'utilisation de méthodes consensuelles. Ces dernières nécessitent de la diversité dans les approches pour devenir pertinentes. Le large panel d'algorithmes de clustering disponibles tend à aumgenter cette diversité. Cependant, elle reste particulièrement limitée pour les modélisations des contigs. Ainsi, pour outrepasser ces limites et tirer profit des méthodes par consensus, nous proposons de développer un nouvel outil dédié à la modélisation des contigs.

4 | Algorithme proposé

L'algorithme proposé pour répondre aux problématiques de cette thèse est détaillé dans ce chapitre. Il se divise en deux grandes parties qui reprennent la logique du binning non supervisé : modélisation des données puis clustering. La modélisation est non supervisée, intégrative, adaptative et ré-utilisable puisqu'implémentée dans un module indépendant, dénommé `fennec`. Le clustering est un processus itératif pour exploiter les capacités d'adaptation de la modélisation.

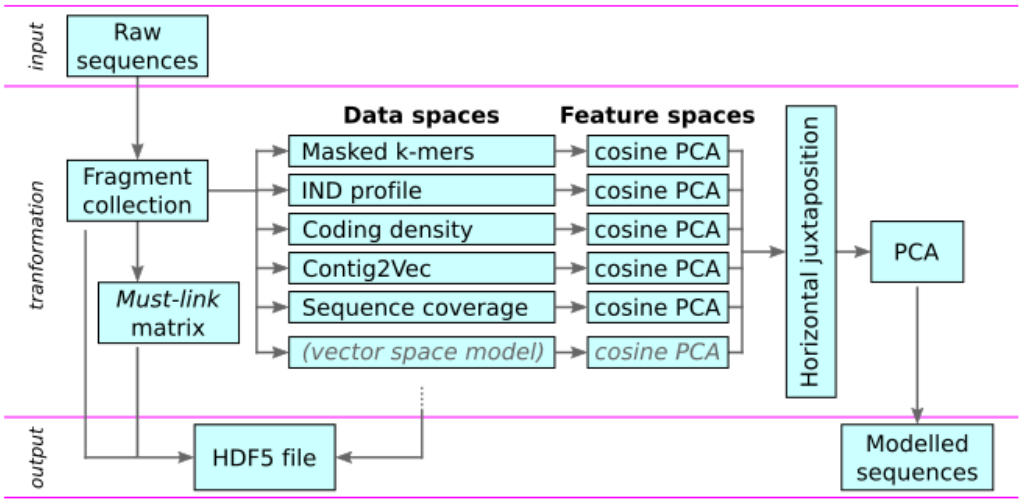


Figure 4.1 | **Worklow général de `fennec` pour la modélisation non supervisée de séquences métagénomiques**.

4.1 Modélisation non supervisée, intégrative et adaptative des contigs

4.1.1 Workflow général

La figure 4.1 synthétise les modélisations et traitements des données qui ont été implémentés au sein de `fennec`. L'utilisateur fournit en entrée les séquences non traitées au format FASTA, quelle que soit leur origine (*e. g.* : génomes complets, contigs, lectures). Celles-ci sont filtrées puis fragmentées avant d'être caractérisées par les modèles choisis par l'utilisateur.

Les modèles « k-mers », « profil des distances inter-nucléotides » et « Contig2Vec » peuvent être appliqués plusieurs fois sur les données avec des paramètres différents (*e. g.* : masques des k-mers).

4.1.2 Démonstration de l'utilisation

`fennec` est implémentée en Python 3.6 et se base sur les bibliothèques *numpy*, *scipy*, *pandas* et *scikit-bio* pour la représentation et la manipulation des données.

L'interface de programmation hérite de *scikit-learn* avec ses méthodes `fit()`, `predict()` et `transform()` permettant ainsi d'exploiter les capacités des pipelines de scikit-learn pour de futures utilisations.

L'intégralité du code est disponible à l'adresse suivante :

https://github.com/keuv-grvl/fennec/. Un exemple de code Python du traitement illustré par la figure 4.1 est présenté ci-après.

1. Chargement et prétraitement des séquences

```
#!/usr/bin/env python3
import fennec
fastafile = "/path/to/file.fasta"
h5file = "/path/to/file.h5file"
seqdb = fennec.DNASequenceBank(
    min_length=1000, chunk_size=10000, overlap=0, verbose=2
)
seqdb.read_fasta(fastafile)
seqdb.to_hdf(h5file)
```

2. Définition des modèles

```
models_to_apply = {
    "kmers4":       fennec.MaskedKmerModel(mask="1111"),
    "kmers110011":  fennec.MaskedKmerModel(mask="110011"),
    "ind15":        fennec.InterNucleotideDistanceModel(K=15),
    "contig2vec4":  fennec.Contig2VecModel(k=4),
    "contig2vec6":  fennec.Contig2VecModel(k=6)
}
```

3. Application des modèles et sauvegarde

```
for model in models_to_apply:
    print(f" - applying {model}")
    X = models_to_apply[model].fit_transform(seqdb)
    print(f"{model} loaded (shape={X.shape})")
    X.to_hdf(h5file, model)
```

4. Chargement des modèles désirés depuis le fichier HDF5

```
wanted_models = ['kmers4', 'contig2vec4']
raw_models, id_list, mustlink_mat =
    fennec._utils.load_models(h5file, wanted_models)
```

où `raw_models` est un dictionnaire contenant les modélisations brutes des données, `id_list` contient la liste des identifiants et `mustlink_mat` est une matrice creuse contenant les liens *must-link* .

5. Affichage du nombre d'attributs de chaque modèle brut

```
print([(i, d.shape[1]) for i, d in raw_models.items()])
```

6. Intégration des différents modèles selon les paramètres de l'utilisateur

```
kpca_params = {
    "inertia": 0.85,  # inertie de l'ACP à noyau à conserver
    "n_jobs":  8,     # nombre de jobs parallèles
    "verbose": 3,     # niveau de verbosité
    "t":       0.75   # proportion de données à utiliser
    # pour l'entraînement
}
D, pca_components, pca_explvar_ratio, n_comp =
    fennec._utils.merge_models(
    raw_models,
    id_list,
    kpca_params
)
```

où `D` est une matrice contenant la modélisation intégrée des séquences, `pca_components` contient les composantes des ACP, `pca_explvar_ratio` contient l'inertie de chacune des composantes et `n_comp` le nombre de composantes finales.

7. Contribution des modèles bruts à la première composante de la modélisation intégrée

```
pcacomp_to_model(D[0], raw_models, 0, outfile="file.csv")
```

8. Application d'un k-means sur les données modélisées et intégrées

```
from sklearn.cluster import Kmeans
km = Kmeans(n_clusters=5)
km.fit(D)
D_clusters = km.predict(D)
```

où `D_clusters` est alors un vecteur contenant le résultat du clustering des séquences modélisées par `fennec`.

Cet exemple montre que l'utilisateur peut définir les modèles de séquences de son choix (Extrait 2) qui seront appliqués aux données de son choix (Extrait 1). Ces modélisations peuvent peuvent alors être enregistrées dans un fi-

chier HDF5 (Extrait 3). Ces différentes étapes ont été automatisées dans le script `fennec_sequence_characterization.py`. Ces modélisations brutes peuvent alors être réutilisées *a posteriori* (Extrait 4) pour être intégrées (Extrait 6).

Ces représentations finales des données métagénomiques peuvent alors être manipulées à l'aide d'algorithmes de clustering existants dans la bibliothèque scikit-learn. Ici, un k-means est appliqué (Extrait 8) uniquement à des fins de démonstration et ne correspond pas à l'algorithme final.

4.1.3 Visualisation de l'adaptabilité

La modélisation finale des séquences des cinq génomes du jeu de données XS a produit 10 composantes pour 2394 fragments de contigs (ou « individus dans le cadre du clustering », celle des séquences des entérobactéries 84 composantes pour 1534 individus et celle des séquences des génomes non entérobactériens 8 composantes pour 860 individus. Enfin, la modélisation des séquences du génome actinobactérien a généré 96 attributs pour 605 individus.

Ces différentes modélisations ont alors été projetées en deux dimensions pour permettre leur visualisation sous forme de nuage de points (figure 4.2).

Il est attendu que les trois génomes des *Enterobacteriaceae* soient difficilement différentiables les uns des autres. La représentation du jeu de données comprenant les cinq génomes permet de visualiser certains clusters facilement identifiables qui se forment spontanément, dont un incluant toutes les séquences actinobactériennes, et deux autres correspondant tous les deux au génome de *Methylotenera mobilis*.

Les trois génomes entérobactériens restent difficilement dissociables mais la visualisation montre qu'ils ne sont pas non plus complètement intriqués les uns dans les autres. Les trois génomes entérobactériens sont représentés par un nuage de points dans le second panel. Ceux-ci ne sont malheureusement toujours pas dissociables les uns des autres sans connaissance *a priori* , mais la représentation qui est produite laisse supposer qu'une séparation est possible puisque les clusters attendus, bien que contigus, ne sont pas clairement chevauchants.

Le troisième panel représente l'intégration des modèles pour les génomes non entérobactériens uniquement. Leur dissociation, mise en évidence suite au traitement du jeu de données complet, se fait encore plus nettement. Enfin, la modélisation des séquences de l'actinobactérie donne un nuage de points homogène en accord avec un seul génome.

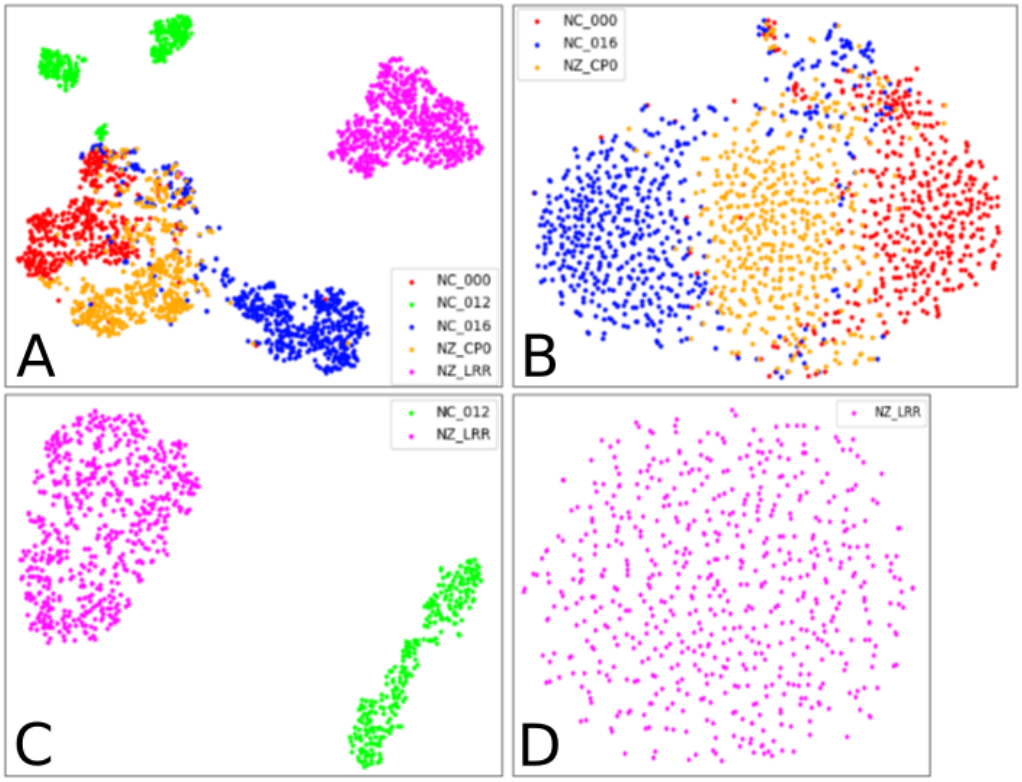


Figure 4.2 | **Visualisation des modélisations des séquences** : (**A**) du jeu de données complet contenant 5 génomes ; (**B**) des séquences entérobactériennes ; (**C**) des séquences non entérobactériennes et (**D**) des séquences actinobactériennes seulement.

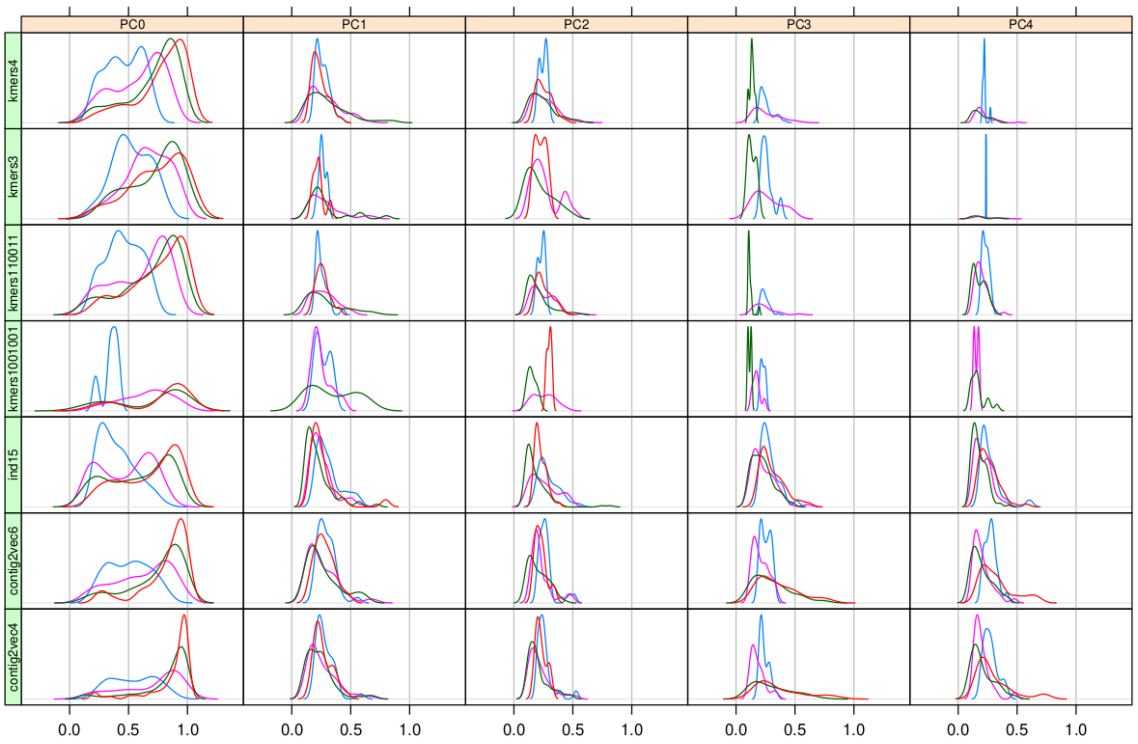


Figure 4.3 | **Contribution des modèles de données bruts** aux cinq premières composantes des quatre jeux de données. Les contributions sont représentées par la distribution des corrélations entre caractéristiques et attributs pour $p < 1 \times 10^{-6}$. Les courbes vertes représentent les cinq génomes ; les courbes magenta représentent les entérobactéries ; les courbes rouges représentent les génomes non entérobactériens ; les courbes bleues représentent l'actinobactérie.

4.1.4 Contribution des différents modèles de données

Les contributions de chaque modèle brut au modèle intégré sont représentées par la densité des corrélations entre les attributs des modèles bruts et les composantes du modèle final (figure 4.3).

La composante PC0 étant celle qui a la plus grande valeur propre lors de l'intégration des modèles, on observe une forte contribution générale de tous les modèles à cette composante. Cependant, on peut dès à présent observer que chaque modèle ne contribue pas de la même façon en fonction du jeu de données à modéliser. En effet, pour la composante PC0, le modèle « kmers4 » est une source majeure d'information pour tous les jeux de donnés (pic de densité des corrélations aux alentours de 0,9) à l'exception du sous-ensemble de données ne concernant que l'actinobactérie (un premier pic de densité de corrélation vers 0,6 et un second vers 0,4). Il en est de même pour la composante PC1 où l'on observe que le modèle

« kmers1001001 » ne contribue pas de la même façon à cette composante pour le jeu de données comprenant les cinq génomes et pour les autres (actinobactérie seule et entérobactéries seules). On remarque également que ce modèle ne contribue pas du tout à cette composante pour le jeu de données comprenant les génomes non entérobactériens.

On observe que l'intégration des modèles bruts se fait en fonction du jeu de données sans aucune supervision afin d'utiliser les attributs ayant les valeurs propres les plus fortes lors de l'intégration des différents modèles, et qui sont donc les plus discriminants pour le jeu de données traité.

4.2 Extraction itérative de clusters

Nous proposons dans cette section une utilisation de la modélisation non supervisée des séquences métagénomiques au sein d'un processus itératif d'extraction de génomes. Ce processus itératif est composé de trois outils originaux (voir section 2.6, à savoir :

1. la modélisation adaptative des séquences proposée par `fennec` ;
2. une modification de l'algorithme VBGMM de CONCOCT pour prendre en compte des contraintes *must-link* ;
3. une procédure d'extraction automatique des clusters.

La structure de ce processus d'extraction de clusters est détaillée dans un premier temps, puis il sera évalué sur les mêmes jeux de données que l'étude comparative du chapitre 3 pour permettre leur évaluation objective, puis sur des jeux de données réelles du chapitre 5.

4.2.1 Présentation générale du processus itératif d'extraction de clusters

Les trois étapes précédemment évoquées (modélisation, clustering puis extraction) ont été orchestrées au sein d'un processus itératif (figure 4.4). Ces étapes ainsi que les différentes conditions d'arrêt de ce processus sont détaillées dans les sous-parties suivantes. Brièvement, la première itération de cet algorithme utilise l'intégralité du jeu de données à traiter. Les modélisations brutes des séquences préalablement déterminées sont intégrées comme décrit précédemment.

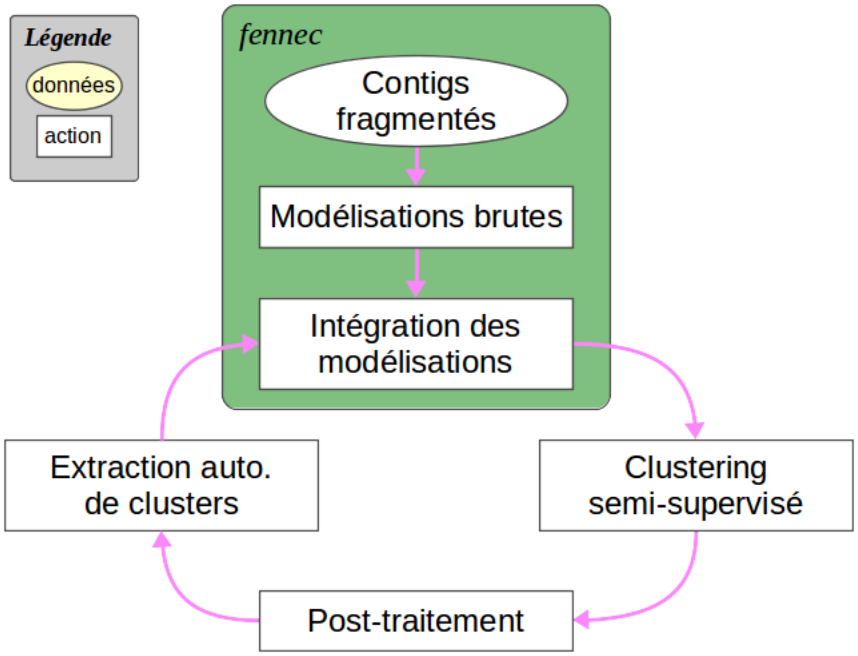


Figure 4.4 | **Vue d'ensemble du processus itératif d'extraction de clusters basé sur `fennec`.** Les contigs fragmentés de séquences sont modélisés une fois. Ces modélisations sont ensuite intégrées, clusterisées et post-traitées. Les clusters les mieux isolés sont alors extraits automatiquement. Ce processus est répété plusieurs fois.

La modélisation intégrée des séquences qui en résulte est alors utilisée pour le clustering. Ce dernier, ainsi que le post-traitement qui en découle, exploite les relations *must-link* qui ont été calculées à partir de la fragmentation des contigs. Ensuite, les clusters les plus isolés sont extraits automatiquement du jeu de données par une analyse des scores *silhouette*. Une nouvelle itération du processus commence alors avec les séquences non extraites à la fin de la première itération.

On réalise alors une nouvelle intégration des modélisations brutes de données pour le sous-ensemble de données restant.

L'intégration des modélisations des données étant adaptative et les données ayant changé, celle-ci peut donc produire des attributs différents et ainsi conduire à un clustering différent, permettant l'extraction de nouveaux clusters indétectables.

4.2.2 Conditions d'arrêt de la boucle et étiquettage des clusters

Le processus étant itératif, des conditions d'arrêt sont nécessaires pour déterminer si l'extraction de clusters n'est plus pertinente (tableau 4.1). Un nombre d'itérations maximum est fixé par défaut à 10 afin de garantir l'arrêt du processus. Ce seuil est suffisant pour traiter les jeux de données simulés. Si des séquences n'ont pas encore été traitées après 10 itérations, elles seront labellisées « maxiter_<identifiant> », avec un identifiant permettant de les considérer néanmoins comme un cluster à part entière. Il est possible de relancer le processus sur les séquences non clusterisées uniquement.

Un nombre maximum de clusters est également fixé afin de provoquer l'arrêt des itérations. Cette valeur est également utilisée comme étant le nombre de composantes de VBGMM et sera décrémentée à chaque fois qu'un cluster sera extrait, permettant de réduire progressivement les temps de calcul (la complexité de VBGMM dépend aussi du nombre de composantes). Elle est fixée par défaut à 400, soit le nombre de composantes par défaut dans l'implémentation originale de VBGMM par CONCOCT. Une fois le nombre de clusters maximum extraits, les séquences restantes sont labellisées « lastcluster_<identifiant> ».

Le nombre de séquences restantes doit ensuite être supérieur au nombre de séquences nécessaires pour constituer un cluster (par défaut, 50), sans quoi le processus itératif s'arrête et labellise les séquences restantes comme « notenough_<identifiant> ».

Après une intégration des modélisations brutes d'un jeu de données, la mo-

Tableau 4.1 | **Critères d'arrêt, valeurs par défaut et préfixes associés** du processus itératif d'extraction de clusters.

Cause d'arrêt	Valeur par défaut	Préfixe
Extraction normale	–	–
Nombre maximum d'itérations atteint	10	`maxiter_`
Nombre maximum de clusters à extraire atteint	400	`lastcluster_`
Nombre de séquences restantes insuffisant	50	`notenough_`
Nombre maximum d'attributs atteint	$min(250, nb_{attributs}/3)$	`unbinned_`
Aucun cluster extrait	–	`badbinned_`

délisation intégrée pourra être jugée non pertinente si elle est constituée de très nombreux attributs, traduisant ainsi une moins bonne capacité de discrimination pour l'algorithme de clustering. De plus, un nombre important d'attributs va fortement impacter le temps nécessaire au clustering (voir Annexe A.3). Cette limite est fixée empiriquement comme étant le minimum entre le tiers du nombre d'attributs initiaux (279 sur nos données d'exemple) et 250. Une intégration des données qui produirait plus d'attributs que voulu arrêterait le processus itératif et labelliserait les séquences restantes comme « unbinned_<identifiant> ».

Finalement, si aucun cluster ne peut être extrait à l'aide des scores *silhouette* , le résultat du clustering est retenu tel quel avec le préfixe « badbinned_<identifiants> » puis le processus itératif est arrêté. Toutes les séquences sont ainsi assignées à un cluster final, qui peut être un cluster correctement extrait (non labellisé) et donc simplement numéroté, ou bien un cluster ayant rencontré une des difficultés précédemment citées, et labellisé en conséquence. Ces labels permettent alors d'exploiter un clustering partiel mais également d'avoir un premier regard quant à la fiabilité des clusters.

4.2.3 Récapitulatif détaillé

La figure 4.5 synthétise les différentes étapes du processus itératif d'extraction de clusters basé sur `fennec` précédemment décrit ainsi que les différents tests d'arrêt et les données utilisées et produites par chacune des étapes. L'utilisateur apporte des

contigs issus de l'assemblage des lectures de séquençage d'un métagénome. Ceux-ci sont fragmentés et modélisés selon des modèles définis par l'utilisateur.

Les modélisations brutes de ces fragments de contigs sont alors représentées numériquement et contenues dans des espaces vectoriels, à raison d'un par modèle. Si un des tests visant à arrêter le processus itératif est passé (nombre d'itérations maximum atteint, assez de séquences à traiter, nombre de clusters à extraire atteint), le processus est stoppé et les fragments de contigs qui n'ont pas encore été assignés à un bin sont labellisés en conséquence. Autrement, les modèles bruts sont intégrés par `fennec` et la contribution des modèles de données brutes à la modélisation intégrée est évaluée. Chaque fragment de contig est alors représenté par un unique vecteur numérique. La pertinence de cette représentation est évaluée en fonction du nombre d'attributs produits. Cette concentration d'information permet ainsi de lutter contre le fléau de la dimension et de mettre en exergue les attributs discriminants du jeu de données. De plus, limiter le nombre d'attributs permet de réduire le temps de calcul lors du clustering.

Le clustering semi-supervisé des modélisations des fragments de contigs utilise alors les relations *must-link* déduites de la fragmentation des contigs pour initialiser le clustering, permettant ainsi d'orienter ce dernier. Le clustering peut produire des clusters anormalement petits, qui sont alors redistribués parmi les autres clusters en utilisant à nouveau les relations *must-link*.

Une projection bidimensionnelle des données est réalisée après clustering et après post-traitement du clustering pour produire des nuages de points (*scatter plot*) à l'aide de la méthode BH-tSNE à partir des modélisations des fragments de contigs.

Une fois post-traités, les clusters sont alors caractérisés selon les scores *silhouette* des individus qui les composent, permettant ainsi d'estimer à quel point un cluster est isolé du reste du jeu de données. Si tel est le cas, celui-ci est extrait automatiquement et donc considéré comme un bin final. Les distributions des scores *silhouette* pour chaque cluster sont représentées sous forme de boîtes à moustache (ou *boxplot*).

Si aucun cluster ne peut être extrait à cette étape, le résultat du clustering post-traité est conservé tel quel et labellisé en conséquence, puis le processus arrêté. Les fragments de contigs qui avaient été assignés à des clusters qui n'ont pas été extraits lors de cette étape constituent alors un nouveau jeu de données. Les modélisations brutes de ce sous-ensemble du jeu de données initial seront alors réutilisées pour une nouvelle itération du processus.

Le processus itératif d'extraction de clusters prend la forme d'un programme Python 3.6 utilisant la bibliothèque `fennec` et associé à un environnement vir-

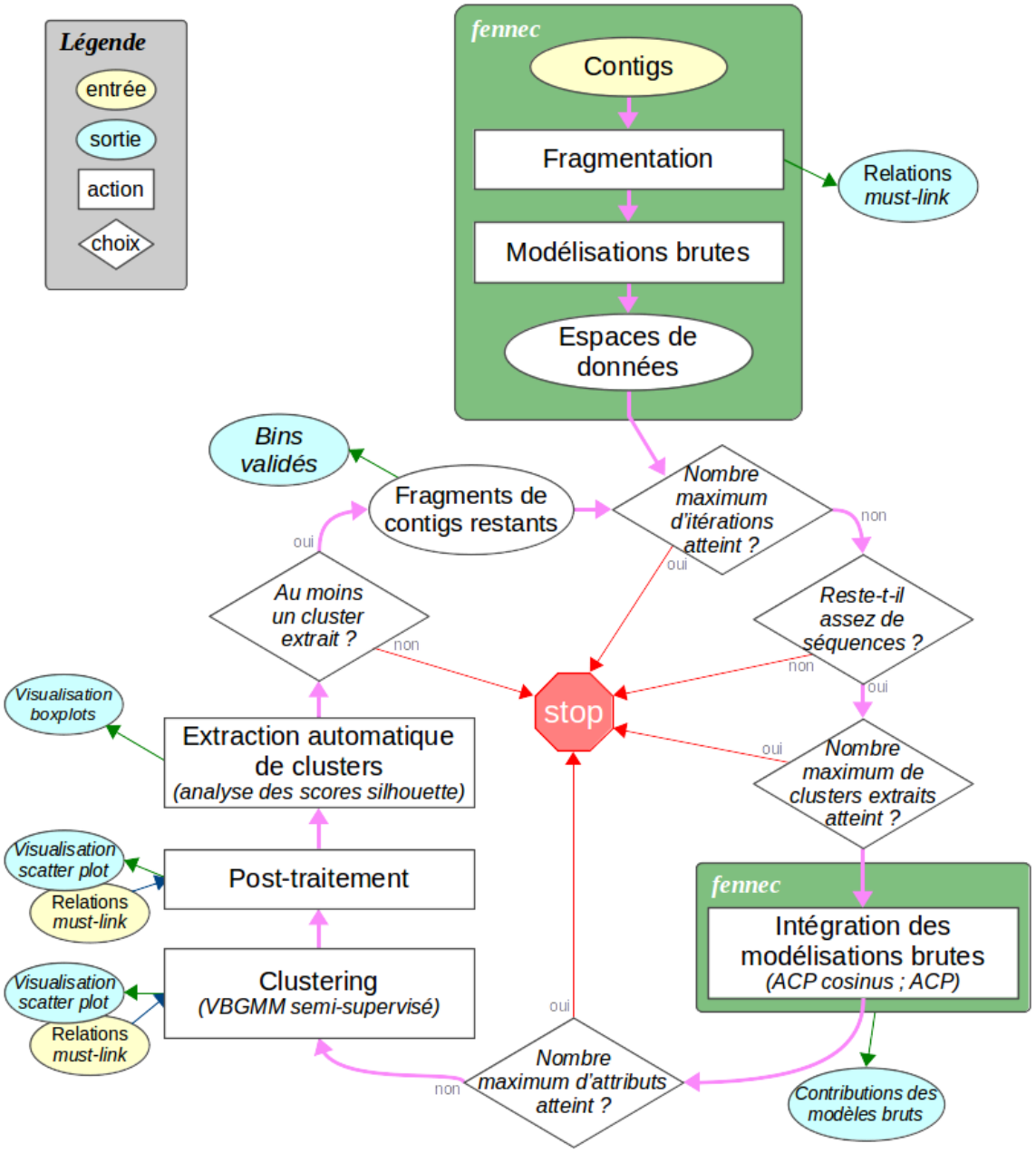


Figure 4.5 | **Vue schématique détaillée du processus itératif**. Les données d'entrée et de sortie sont indiquées par les ellipses jaunes et bleues respectivement. Les traitements réalisés par le processus sont les actions, symbolisées par les rectangles blancs. Les losanges symbolisent les vérifications qui sont effectuées au cours du processus qui peuvent conduire à l'arrêt de celui-ci. Les étapes assurées par `fennec` sont incluses dans les boîtes vertes.

tuel Conda (Grüning *et al.*, 2018) permettant la portabilité et la reproductibilité des traitements. Ces différentes étapes ont été automatisées dans le script `fennec_cluster_extraction_pipeline.py`.

5 | Application du processus itératif d'extraction de clusters

Le processus itératif d'extraction de clusters a été appliqué à de multiples jeux de données, simulés ou réels. Toutes les données ont fait l'objet du même prétraitement et ont été traitées selon les mêmes modalités que l'étude comparative. L'évaluation des résultats obtenus a été réalisée à partir des résultats de référence lorsque ceux-ci sont connus (chapitre 3), sinon par comparaison à d'autres résultats de binning.

5.1 Données simulées

5.1.1 Présentation des bins produits pour le jeu de données S

Le processus itératif étant une succession de plusieurs méthodes produisant de nombreuses données, un exemple de résultat obtenu au cours de deux itérations est présenté.

Pour la première itération de l'algorithme sur le jeu de données S, la modélisation des données a produit 20 attributs. La projection des fragments de séquences ainsi représentés en un nuage de points permet d'identifier visuellement des regroupements d'individus (figure 5.1). La plupart des clusters individualisés sont composés de séquences originellement issues d'un même génome.

La visualisation des distributions des scores *silhouette* obtenus après analyse du résultat du clustering corrobore la visualisation bidimensionnelle précédente (figure 5.1 et figure 5.2). On distingue en effet des scores *silhouette* médians supérieurs à 0,5 qui pourraient correspondre aux clusters les plus périphériques sur la projection bidimensionnelle du jeu de données. À l'inverse, les clusters ayant des scores *silhouette* moindres présentent des zones de chevauchement avec d'autres clusters et correspondent à ceux principalement situés au centre du nuage du points de la projection bidimensionnelle du jeu de données.

Conformément aux critères décrits précédemment (voir sous-section 4.2.2), les clusters 1, 3, 5, 10, 11, 12, 13, 15, 16, 17, 19 et 22 représentant à eux tous 7157 fragments de séquences – soit 45,19 % du jeu de données initial – ont été extraits suite à cette première itération. Les 8681 fragments de séquences restants subissent une seconde itération du processus, c'est-à-dire une nouvelle modélisation des séquences puis une nouvelle extraction des clusters.

Les modélisations brutes des 8681 fragments restants sont intégrées comme décrit précédemment et produisent alors 54 attributs. Ces attributs sont alors utilisés pour caractériser les fragments de séquences et utilisés par l'algorithme de clustering VBGMM modifié pour prendre en compte les relations *must-link* entre les données à traiter. De la même façon, une visualisation des données sous forme de nuage de points est obtenue ainsi que la distribution des scores *silhouette* des fragments de séquences (figure 5.3).

A la fin de cette deuxième itération, les clusters 3, 5, 6 et 9 seront extraits, représentant 2078 fragments de séquences, soit 23,94 % des données à traiter à la deuxième itération. Les 6603 fragments de séquences restants à l'issue de cette

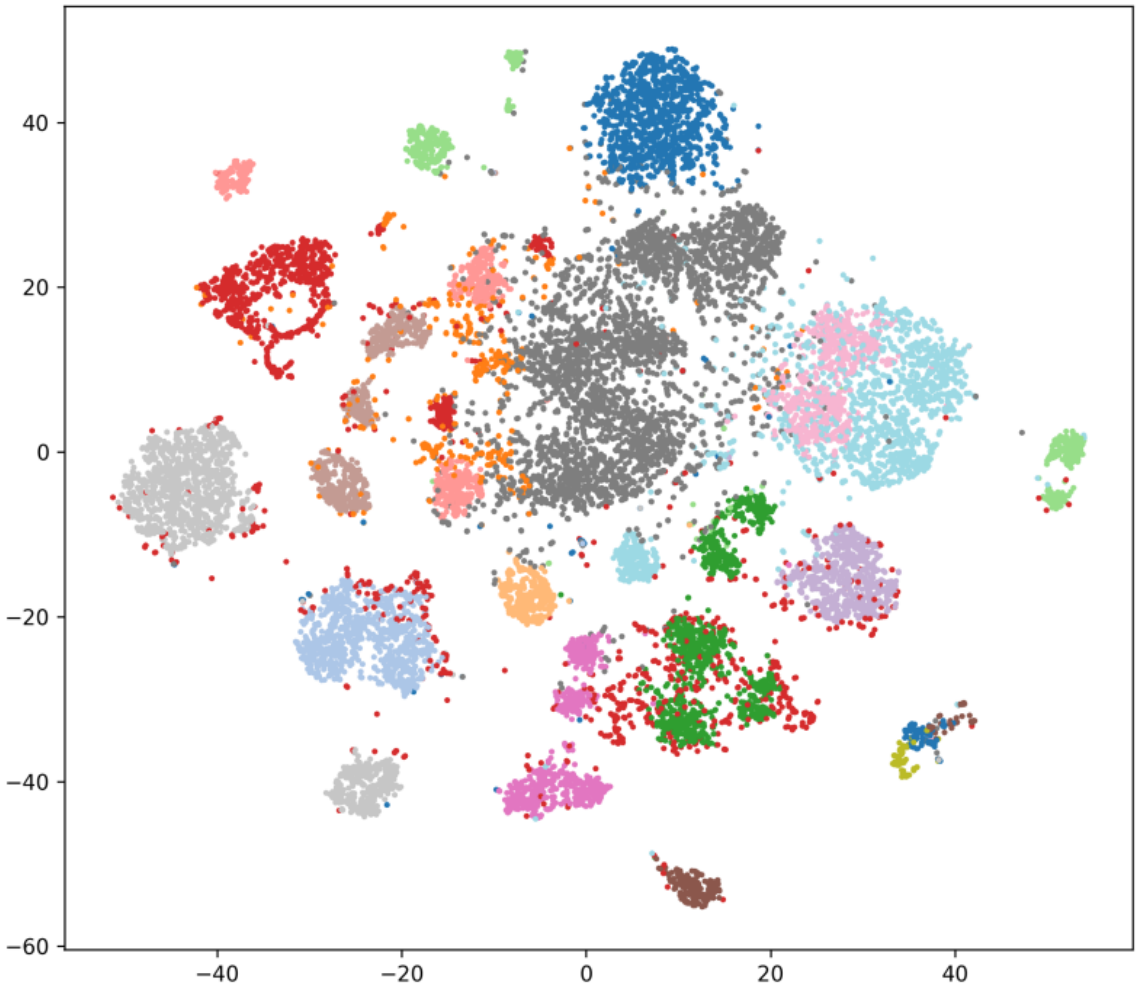


Figure 5.1 | **Projection des séquences du jeu de données** S avec la méthode BH-tSNE après modélisation intégrative et adaptative du jeu de données complet. Les séquences représentées par les points ont été colorées en fonction du résultat du clustering à l'exception des points rouges qui correspondent à des séquences non assignées à un bin.

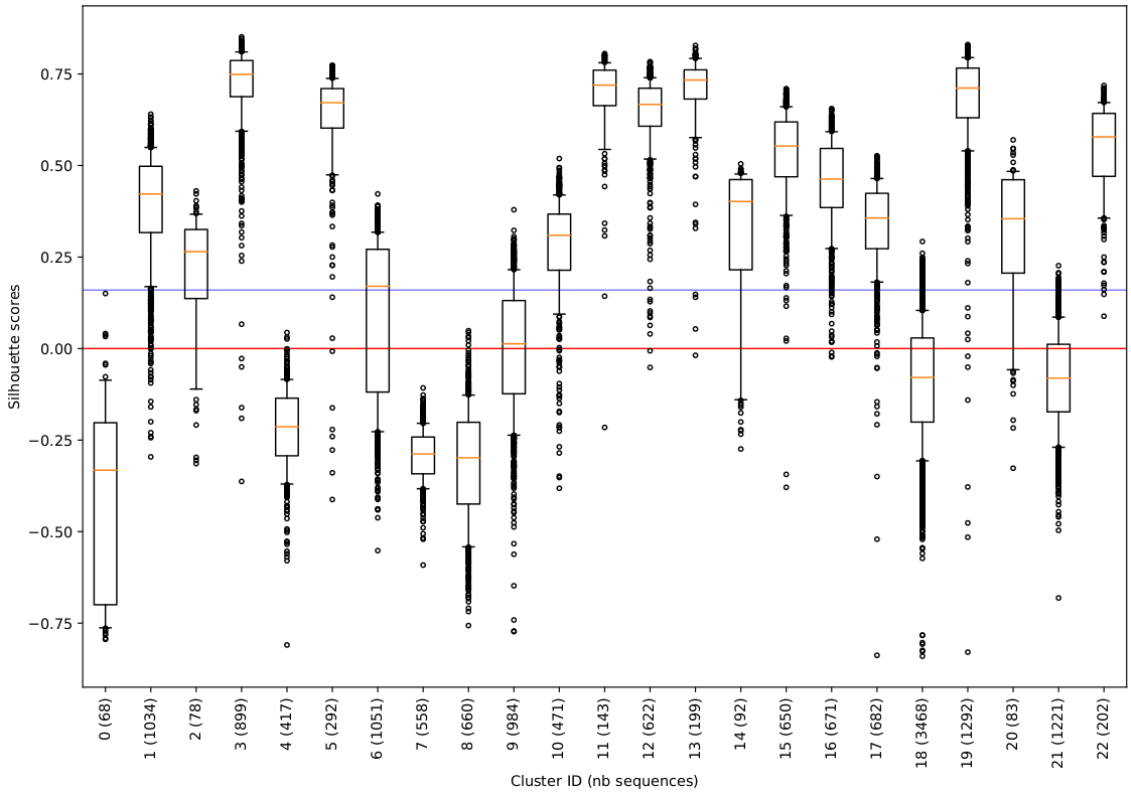


Figure 5.2 | **Distributions des scores *silhouette* de chaque cluster à la fin de la première itération pour le jeu de données** S. Le nombre de séquences que contient chaque cluster est indiqué entre parenthèses. Si une boîte à moustaches a sa médiane (ligne orange) supérieure à la médiane globale (ligne bleue) ainsi que la moustache inférieure (premier décile) supérieure à 0 (ligne rouge), alors le cluster est extrait (*e. g.* : cluster 3). Les séquences des clusters non extraits subiront alors une nouvelle itération du processus.

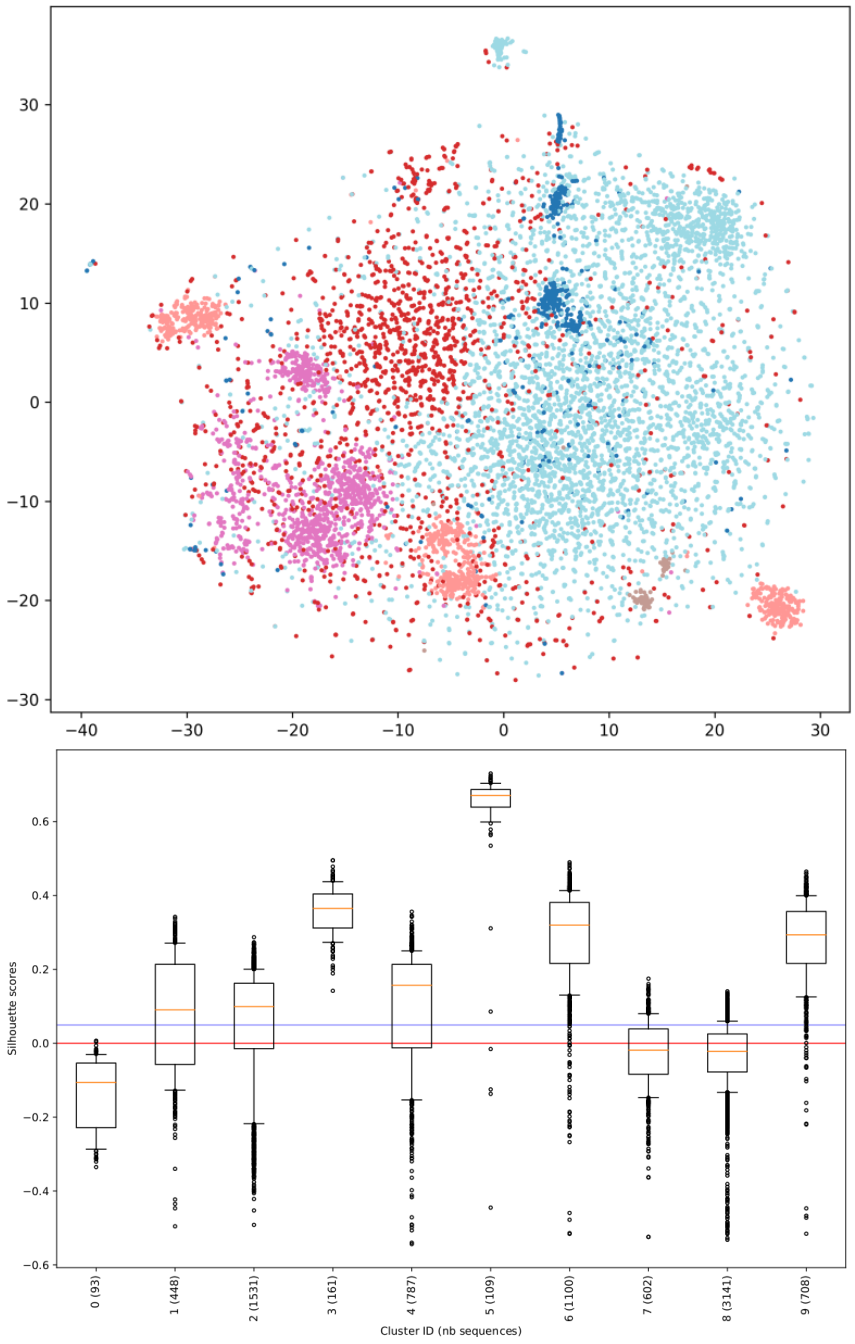


Figure 5.3 | **Visualisation des résultats de la deuxième itération du traitement du jeu de données** S. Le panel supérieur permet de visualiser les séquences après modélisation intégrative puis projection avec la méthode BH-tSNE. Les séquences (points) ont été colorées en fonction du résultat du clustering. Le panel inférieur représente les distributions des scores *silhouette* par cluster.

deuxième itération seront alors analysés de la même façon par de nouvelles itérations du processus et ce jusqu'à rencontrer un des critères d'arrêt du processus. Pour chaque itération, les visualisations des séquences, des scores *silhouette* et la clusterisation sont effectuées.

5.1.2 Résultats finaux du processus itératif

Le processus itératif s'est arrêté après respectivement 4, 8 et 7 itérations pour les jeux de données S, M et L suite à l'échec d'extraction d'au moins un cluster : les fragments de contigs restants sont alors labellisés « badbinned ». Une fois le processus d'extraction de clusters terminé, un fichier `final_clustering.csv` présente le résultat du binning au format CSV où chaque fragment de contig est assigné à un cluster selon les critères cités précédemment. Les conditions d'extraction des clusters sont alors identifiables : soit l'extraction s'est déroulée normalement, auquel cas le bin est identifié par un nombre ; soit un des critères d'arrêt précédemment décrits est rencontré, auquel cas le bin est labellisé tel qu'indiqué dans le tableau 4.1 (page 97).

Ainsi, 22, 38 et 51 bins ont été proposés par le processus itératif, dont 17, 34 et 42 qui ont été extraits automatiquement pour les jeux de données S, M et L respectivement, les autres bins étant les clusters résultant du clustering mais qui n'ont pas validé les critères d'extractabilité (tableau 5.1). Les clusters extraits totalisent 9554, 30665 et 32797 fragments de contigs, soit 60 %, 82 % et 57 % des jeux de données S, M et L. Les bins qui n'ont pas été extraits automatiquement mais néanmoins proposés par l'étape de clustering représentent 6283, 6854 et 24436 fragments de contigs répartis en 5, 4 et 9 bins composés de 51 à 3257, 20 à 3647 et 189 à 5227 fragments de contigs pour les jeux de données S, M et L respectivement.

5.1.3 Performances de calcul

Le processus itératif d'extraction de clusters a été testé sur une machine disposant de 160 CPU (10 processeurs Intel Xeon CPU E7-8870 cadencés à 2,40GHz) et de 1 To de RAM. Le clustering a été paramétré pour être exécuté 32 fois en parallèle et l'intégration des données par `fennec` repose sur *numpy* qui s'appuie lui-même sur OpenBLAS qui est limité à 64 threads dans notre environnement de test.

Une parallélisation efficace du processus devrait au mieux atteindre 6400 % lors de l'intégration des modèles bruts et 3200 % lors du clustering et donc se traduire par une utilisation des CPU d'au moins 3200 % pour l'intégralité du processus. Cependant, l'étape de sélection des liens *must-link* pour un sous-ensemble de données

Tableau 5.1 | **Caractéristiques des résultats du processus itératif d'extraction de clusters** appliqué aux données de test.

	S	**M**	**L**
Nombre de contigs	15837	37519	57233
Nombre d'itérations	4	8	7
Cause de l'arrêt des itérations	Plus de cluster extractible	Plus de cluster extractible	Plus de cluster extractible
Nombre de bins attendu	22	38	51
Nombre de bins obtenu	17	34	42
Nombre de contigs extraits	9554	30665	32797
Proportion de séquences extraites	60 %	82 %	57 %
Taille du plus petit bin obtenu	109 contigs	118 contigs	53 contigs
Taille du plus grand bin obtenu	1292 contigs	3311 contigs	3637 contigs
Nombre de bins non extraits	5	4	9
Taille du plus petit bin non extrait	51 contigs	20 contigs	189 contigs
Taille du plus grand bin non extrait	3257 contigs	3647 contigs	5227 contigs

à partir de la matrice originale est une étape non parallélisable et relativement longue. Ainsi l'utilisation des CPU varie de 1490 % à 1609 %, dénotant une scalabilité importante mais perfectible (tableau 5.2). Les temps de calcul observés sur les jeux de données de tests, allant de 1h03 à 13h08, sont compatibles avec les critères de sélection des logiciels de l'étude comparative (voir chapitre 3).

5.1.4 Contributions des différents modèles

La modélisation intégrative des séquences proposée par le logiciel `fennec` est donc utilisée pour représenter les fragments de séquences au début de chaque itération. L'intégration des modèles de données brutes pour les trois jeux de données S, M et L lors de la première itération est dépendante du jeu de données (figure 5.4).

De plus, l'extraction progressive des clusters modifie le jeu de données d'une itération à l'autre. Les modélisations de données brutes à chacune des itérations a

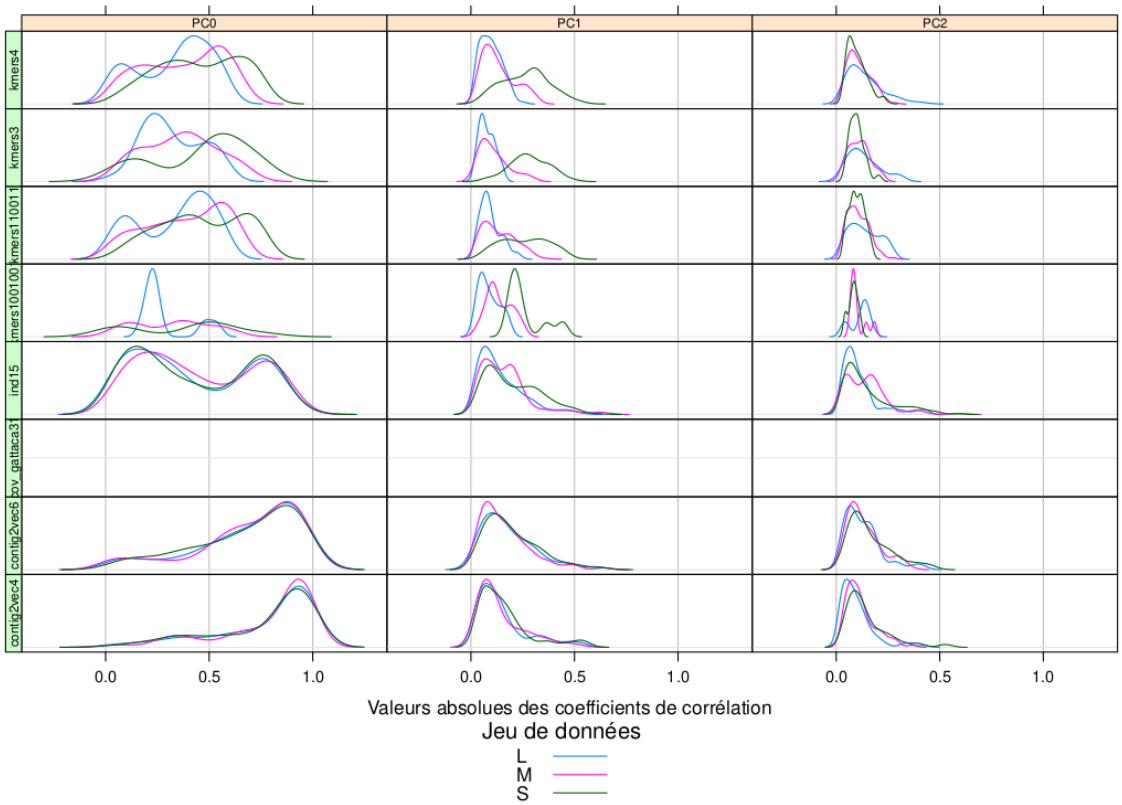


Figure 5.4 | **Contributions des différents modèles de données aux 3 premières composantes** de la modélisation intégrative et adaptative des séquences pour les jeux de données S, M et L et pour la première itération seulement. Chaque courbe représente la distribution des valeurs absolues des corrélations entre les attributs d'une modélisation brute des séquences et une des composantes de la modélisation intégrative et adaptative de ces mêmes séquences. On peut ainsi estimer la contribution des modèles bruts au modèle final. On visualise ici l'aspect adaptatif de la modélisation par `fennec` lorsque les courbes de distribution ne sont pas superposées. Par exemple, les attributs proposés par le modèle « Contig2Vec4 » contribuent énormément à la composante PC0 (pic dans la distribution proche de 1) et ce quelle que soit le jeu de données. À l'inverse, le modèle « kmers4 » contribue davantage à la PC0 des données de S que de L. La couverture des séquences n'est dans ce cas pas un modèle qui contribue à la modélisation des séquences.

Tableau 5.2 | **Temps et ressources informatiques consommées par le processus itératif d'extraction de clusters** appliqué aux jeux de données de test.

Jeu de données	Nombre de fragments de séquences	Temps horloge (h:m:s)	Mémoire maximum (Mo)	Utilisation CPU (%)
S	15837	01:03:47	28022	1609
M	37519	04:01:59	153401	1490
L	57233	13:08:32	390088	1546

donc conduit, à partir des mêmes modélisations brutes, à une modélisation intégrée de ces données différente (figure 5.5).

Les différences entre les contributions des attributs apportés par les différents modèles de séquences montrent que la modélisation intégrative des séquences est capable de s'adapter aux données à traiter en mettant en exergue des attributs différents en fonction d'un ensemble ou sous-ensemble de données (figure 5.6). De plus, certains modèles de séquences semblent contribuer de la même façon à différents jeux de données (e.g : « Contig2Vec4 » et « Contig2Vec6 ») laissant ainsi supposer que ces deux modèles apportent, pour les jeux de données traités ici, des informations redondantes.

5.1.5 Intégration des résultats à l'étude comparative

La réutilisation des données de l'étude comparative (chapitre 3) permet la comparaison des résultats obtenus par le processus itératif d'extraction de clusters proposé avec ceux obtenus lors de l'évaluation des méthodes de binning existantes (figure 5.7 et tableau 5.3).

Pour rappel, CONCOCT donne les meilleurs résultats sur les jeux de données S, M et L lorsque l'objectif affiché est de traiter toutes les séquences du jeu de données d'entrée, et donc de n'en laisser aucune non regroupée. La précision et la fiabilité des résultats du binning (qualifiées de binning precision et binning accuracy dans le chapitre 3) atteignent en moyenne 0,8887 (± 0,3626) et 0,9856 (± 0,0031) respectivement sur l'ensemble des jeux de données. Sur ces aspects, MetaProb obtient également des scores élevés, avec une précision moyenne de 0,8372 (± 0,0374) et une fiabilité moyenne de 0,9732 (± 0,0076). À l'inverse, l'importante part des séquences non traitées par les différentes versions de MetaBAT ne lui permet d'atteindre au

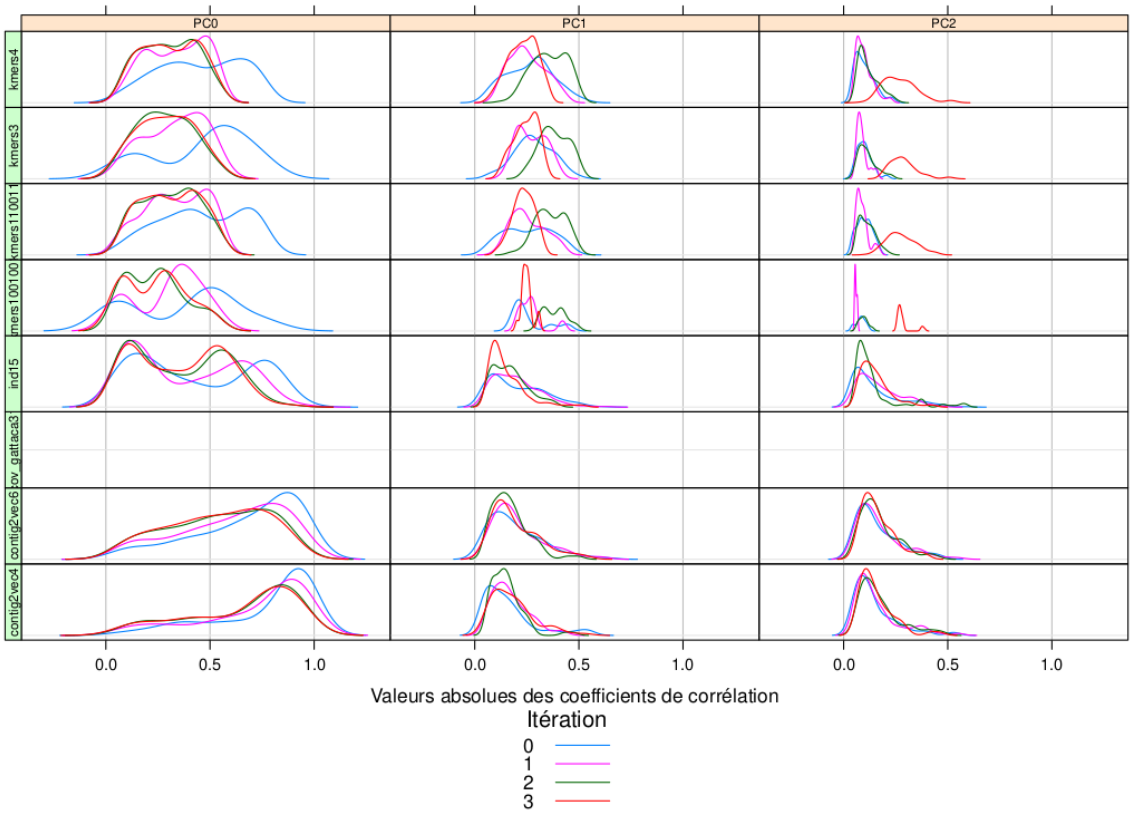


Figure 5.5 | **Contributions des différents modèles de données aux 4 premières itérations** pour les 3 premières composantes et pour le jeu de données S uniquement. On visualise que pour un même jeu de données d'origine duquel certaines séquences sont extraites d'une itération à l'autre, les contributions des différents modèles de données brutes varient. Par exemple, les attributs apportés par le modèle « kmers110011 » contribuent très fortement à la composante PC0 mais tendent à diminuer lorsque l'on progresse dans les itérations. À l'inverse, la contribution de ce modèle à la PC1 est la plus forte pour l'itération 2. Ces variations dans les contributions d'une itération à l'autre montre la capacité d'adaptation de la modélisation.

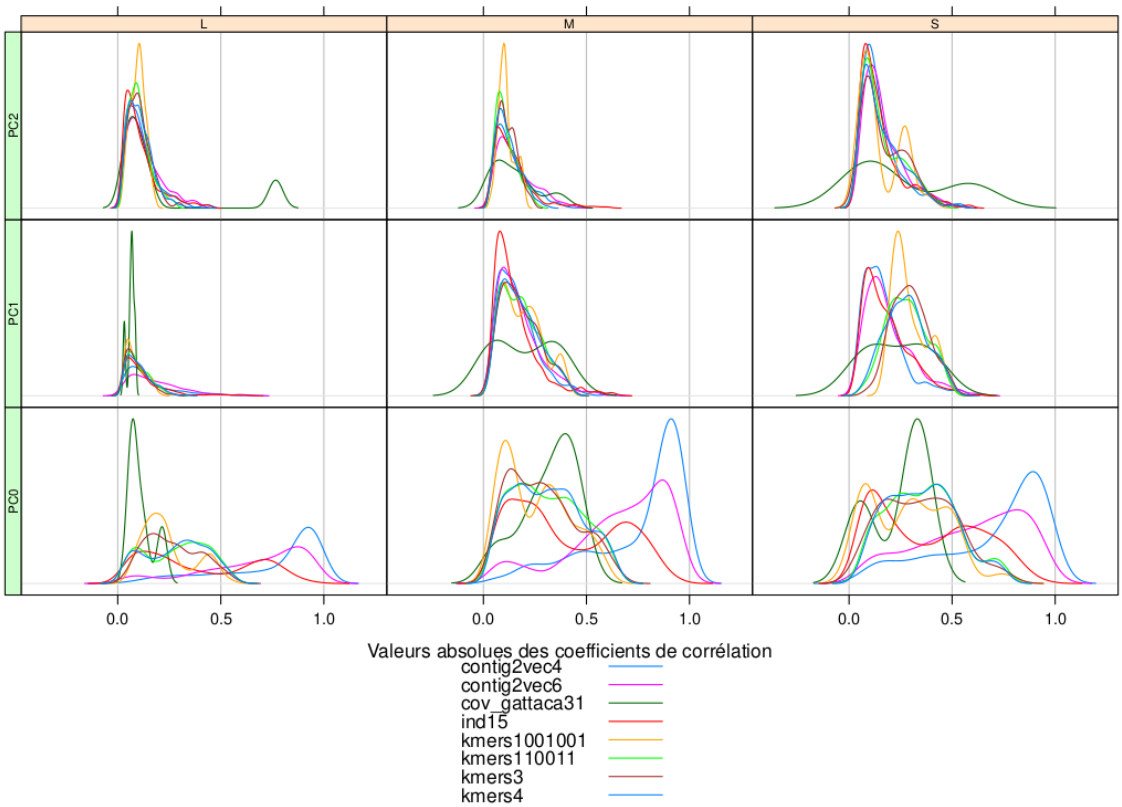


Figure 5.6 | **Contributions des différents modèles de données aux 3 premières composantes** pour les jeux de données S, M et L pour toutes les itérations réunies. On visualise que, pour chaque jeu de données, les attributs qui contribuent le plus aux mêmes composantes varient. Par exemple, les attributs du modèles « Contig2Vec4 » contribuent majoritairement à toutes les composantes PC0 mais dans des proportions différentes alors que la couverture moyenne des séquences contribuent peu à la composante PC0 mais plus fortement à la PC2, en particulier pour les jeux de données L et S et dans une bien moindre mesure pour le jeu de données M. Ces variations dans les contributions des différents modèles entre les différents jeu de données montrent la capacité d'adaptation de la modélisation aux séquences traitées.

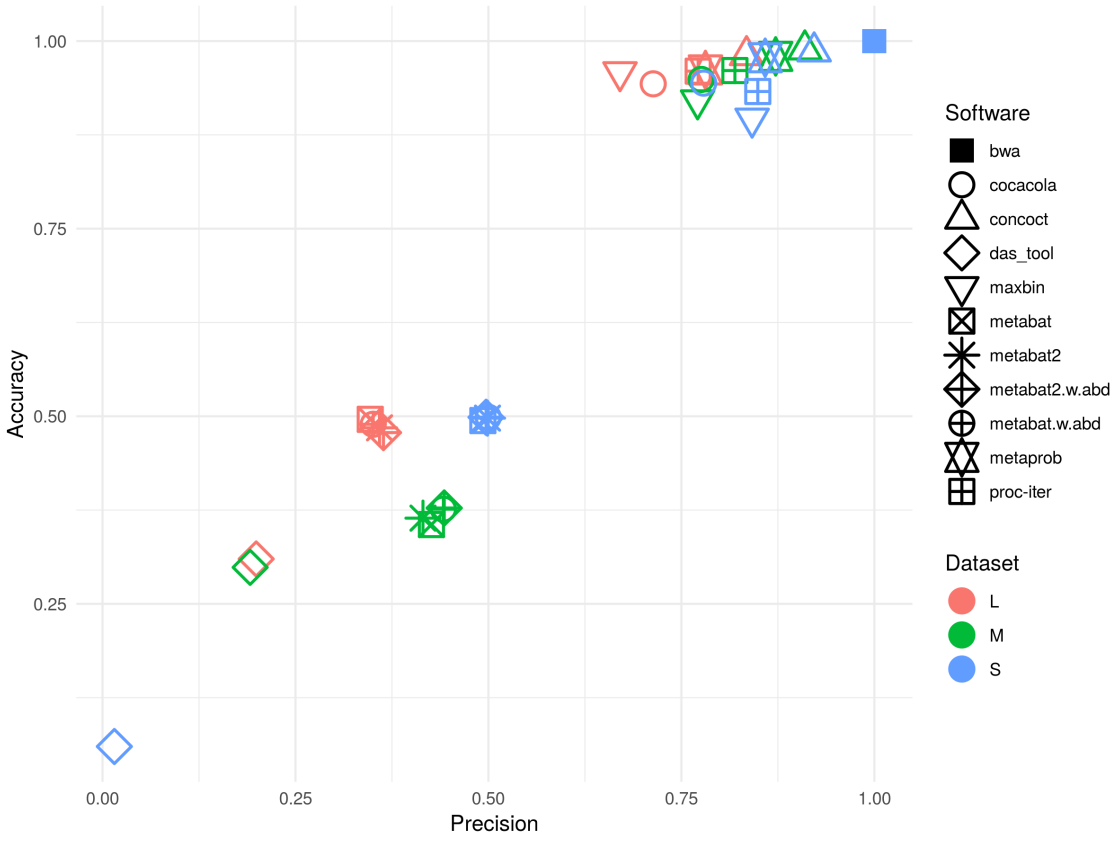


Figure 5.7 | **Précision et fiabilité moyennes des logiciels de binning** dans le cas où les données non traitées sont considérées comme regroupées incorrectement. Les références ont nécessairement une précision et une fiabilité moyennes de 1,0 pour les trois jeux de données S, M et L.

mieux qu'une précision de 0,4307 ($\pm$ 0,053).

Le processus d'extraction de clusters basé sur la modélisation `fennec` atteint une précision moyenne de 0,8133 ($\pm$ 0,0278) et une fiabilité moyenne de 0,9515 ($\pm$ 0,0123). Bien qu'il n'atteigne pas les performances de CONCOCT sur les données simulées, le processus d'extraction basé sur `fennec` obtient des résultats aux performances similaires à celles de MetaProb et supérieures ou égales aux autres méthodes de binning testées (DAS_Tool, MetaBAT, MaxBin et COCACOLA) sur les jeux de données de test.

TABLEAU 5.3 | **Nombres de vrais positifs, faux positifs, faux négatifs et vrais négatifs produits par chaque logiciel pour chaque jeu de données**. Les précision, fiabilité, FDR (*false discovery rate*), sensibilité, spécificité et rappel moyens sont ensuite calculés. Les taux de complétude et de contamination définis par CheckM correspondent respectivement à la sensibilité et au FDR.

		Vrais positifs	Faux positifs	Faux négatifs	Vrais négatifs	Précision	Fiabilité	FDR	Sensibilité	Spécificité	Rappel
Référence	S	15837	0	0	776013	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000
	M	37519	0	0	4914989	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000
	L	57233	0	0	12705726	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000
COCACOLA	S	12335	3502	40859	735154	0,7789	0,9440	0,0658	0,2319	0,9953	0,2319
	M	29103	8416	244696	4670293	0,7757	0,9489	0,0307	0,1063	0,9982	0,1063
	L	40834	16399	710728	11994998	0,7135	0,9430	0,0218	0,0543	0,9986	0,0543
CONCOCT	S	14601	1236	9844	766169	0,9220	0,9860	0,0506	0,5973	0,9984	0,5973
	M	34138	3381	52736	4862253	0,9099	0,9887	0,0389	0,3930	0,9993	0,3930
	L	47755	9478	232409	12473317	0,8344	0,9810	0,0338	0,1705	0,9992	0,1705
DAS_Tool	S	243	15594	728825	47188	0,0153	0,0599	0,0214	0,0003	0,7516	0,0003
	M	7176	30343	3443194	1471795	0,1913	0,2986	0,0088	0,0021	0,9798	0,0021
	L	11394	45839	8762721	3943005	0,1991	0,3098	0,0052	0,0013	0,9885	0,0013
MaxBin	S	13328	2509	79147	696866	0,8416	0,8969	0,0271	0,1441	0,9964	0,1441
	M	28926	8593	381667	4533322	0,7710	0,9212	0,0209	0,0704	0,9981	0,0704
	L	38371	18862	507976	12197750	0,6704	0,9587	0,0345	0,0702	0,9985	0,0702
MetaBAT	S	7804	8033	392435	383578	0,4928	0,4943	0,0201	0,0195	0,9795	0,0195
	M	15988	21531	3169567	1745422	0,4261	0,3557	0,0068	0,0050	0,9878	0,0050
	L	19849	37384	6396237	6309489	0,3468	0,4959	0,0058	0,0031	0,9941	0,0031
MetaBAT avec abondance	S	7914	7923	389058	386955	0,4997	0,4987	0,0200	0,0199	0,9799	0,0199
	M	16570	20949	3068352	1846637	0,4416	0,3762	0,0068	0,0054	0,9888	0,0054
	L	20070	37163	6480703	6225023	0,3507	0,4893	0,0057	0,0031	0,9941	0,0031
MetaBAT2	S	7899	7938	389951	386062	0,4988	0,4975	0,0200	0,0199	0,9799	0,0199
	M	15577	21942	3126685	1788304	0,4152	0,3642	0,0070	0,0050	0,9879	0,0050
	L	20540	36693	6541355	6164371	0,3589	0,4846	0,0056	0,0031	0,9941	0,0031
MetaBAT2 avec abondance	S	7870	7967	389017	386996	0,4969	0,4987	0,0201	0,0198	0,9798	0,0198
	M	16609	20910	3060838	1854151	0,4427	0,3777	0,0068	0,0054	0,9888	0,0054
	L	20831	36402	6622755	6082971	0,3640	0,4782	0,0055	0,0031	0,9941	0,0031
MetaProb	S	13596	2241	15362	760651	0,8585	0,9778	0,0774	0,4695	0,9971	0,4695
	M	32719	4800	96337	4818652	0,8721	0,9796	0,0372	0,2535	0,9990	0,2535
	L	44705	12528	478056	12227670	0,7811	0,9616	0,0240	0,0855	0,9990	0,0855
Processus Itératif	S	13445	2392	50836	725177	0,8490	0,9328	0,0372	0,2092	0,9967	0,2092
	M	30740	6779	187724	4727265	0,8193	0,9607	0,0310	0,1407	0,9986	0,1407
	L	44166	13067	498549	12207177	0,7717	0,9599	0,0241	0,0814	0,9989	0,0814

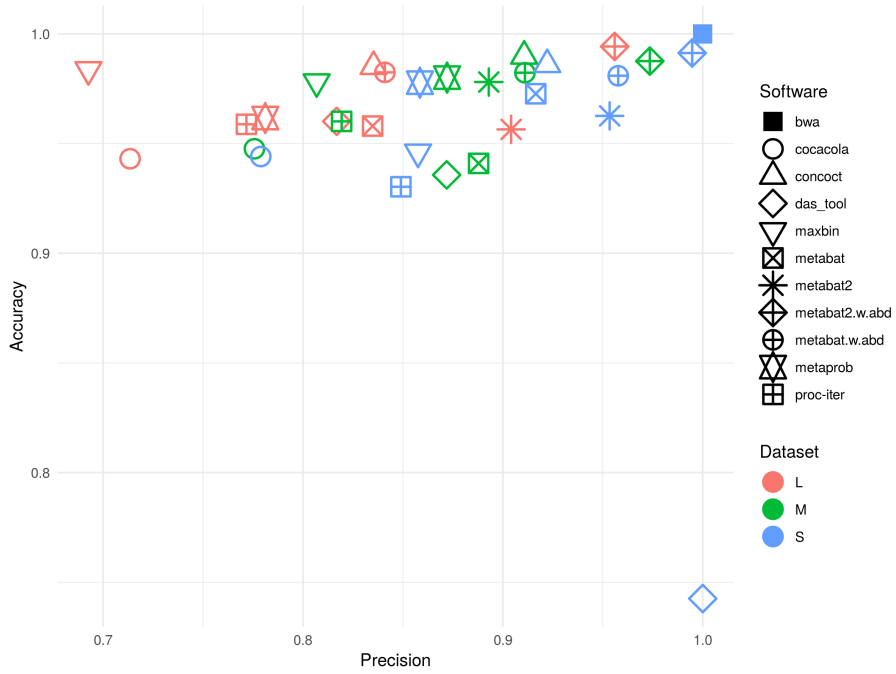


Figure 5.8 | **Précision et fiabilité moyennes des logiciels de binning** dans le cas où les données non traitées ne sont pas prises en compte dans le calcul des métriques.

Comme au chapitre 3, les fragments de séquences non traités peuvent être complètement écartées des données, y compris dans le calcul des métriques pour caractériser la fiabilité des résultats (figure 5.8). Dans ce cas, MetaBAT2 (avec les abondances) devient alors le logiciel proposant les meilleurs résultats malgré la faible proportion de données traitées (voir section 3.2). Le processus itératif d'extraction de génomes a pu traiter davantage de données (60 %, 82 % et 57 % pour les jeux de données S, M et L respectivement) mais celles-ci ne semblent pas constituer des bins aussi fiables que les quelques uns proposés par les deux logiciels cités.

Ainsi, de la même manière qu'au chapitre 3, ceci place MetaBAT2 comme outil à privilégier pour une analyse rapide et la reconstruction de quelques génomes tandis que des outils comme CONCOCT et le processus itératif d'extraction de génomes pour des analyses exploratoires. Ce dernier propose néanmoins des résultats individuellement moins bons que ses deux concurrents.

5.1.6 Diversité des différents résultats de binning

Les logiciels ont été profilés puis comparés par visualisation après projection par PCoA (figure 5.9). Le résultat de référence et le contrôle négatif sont diamétralement

opposés et les outils de binning distribués entre ceux-ci. CONCOCT est le plus proche de la référence pour tous les jeux de données à l'expection du $CAMI1h$, comme précédemment observé. On peut constater que : (i) les outils de binning ne sont pas regroupés mais plutôt équidistants du résultat de référence et (ii) les résultats dépendent de la communauté artificielle.

La stratégie d'agrégation de DAS_Tool est observée ici car elle est quasiment équidistante à tous les autres logiciels de binning, en particulier pour le jeu de données $CAMI1h$.

Le plan expérimental de $CAMI1h$ (5 répétitions) et la complexité du métagénome (*e. g.* : présence d'eucaryotes et de virus), même si elle est limitée par rapport à un métagénome réel, peuvent expliquer les différences observées, confirmant la nécessité de divers ensembles de données de test pour une évaluation non supervisée du regroupement.

Un tel comportement peut s'expliquer par le fait que chaque logiciel propose une solution différente au même problème de binning, même avec des performances de binning similaires. En cela, l'approche du processus itératif d'extraction de clusters propose bien une approche nouvelle.

5.1.7 Complémentarité des différents résultats de binning

La complémentarité de ces résultats a ensuite été étudiée (tableau 5.4). Un bin consensuel virtuel est considéré comme valide lorsqu'au moins n logiciels produisent le résultat escompté d'après la référence. Si au moins 50 % des outils testés ($n = 4$ pour S, M et L et $n = 5$ pour $CAMI1h$) fournissent le résultat escompté, on pourrait s'attendre à 76,74 %, 72,50 %, 55,78 % et 58,92 % des ensembles de données S, M, L et $CAMI1h$ respectivement.

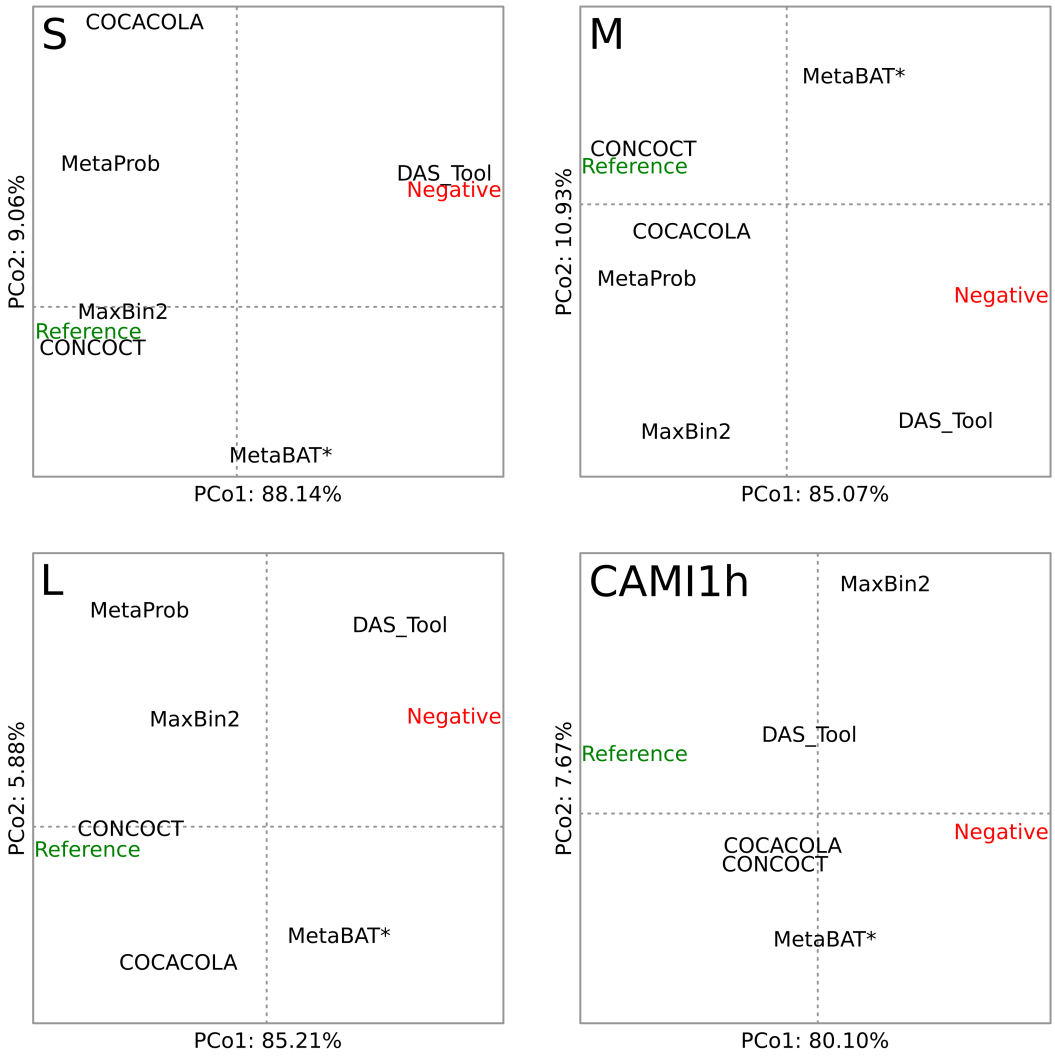


Figure 5.9 | **Projection des distances entre les résultats de binning**. Les résultats du binning sont profilés sur leur capacité à regrouper correctement chaque séquence par rapport aux bins de référence. La distance de Manhattan est calculée entre chaque paire de profils et la matrice de distances résultante est visualisée au moyen d'une analyse en composantes principales (PCoA). Un témoin négatif est ajouté pour signaler les pires résultats possibles. Les deux premiers axes cumulent au moins 87,77 % d'inertie dénotant une visualisation bidimensionnelle représentative. Les variations multiples d'un même logiciel sont marquées d'un astérisque (*e. g.* : "MetaBAT*") si leurs résultats sont suffisamment proches pour ne pas pouvoir être distingués sur la représentation visuelle.

Tableau 5.4 | **Intégration des résultats du processus itératif à l'approche consensuelle par vote majoritaire**. (A) Résultats hypothétiques d'un consensus par choix de la meilleure solution possible parmi tous les résultats disponibles. (B) Résultats hypothétiques d'un consensus par vote majoritaire. (C) Nombre de bins qui sont : (i) confirmés par le processus itératif par rapport aux autres méthodes ; (ii) pour lesquels aucune méthode de binning n'a jamais proposé le bon résultat ; (iii) pour lesquels seul le processus itératif a proposé un résultat faux ; (iv) pour lesquels seul le processus itératif a proposé le bon résultat.

		S (15837 contigs)	**M** (37519 contigs)	**L** (57233 contigs)	**Moyenne** (± écart-type)
(A)	Consensus du « meilleur » sans le processus itératif	97,56 %	97,07 %	95,46 %	96,69 % (± 1,10)
(A)	Consensus du « meilleur » avec le processus itératif	98,55 %	97,49 %	96,43 %	97,49 % (± 1,06)
(A)	Gain approche consensus du « meilleur »	+0,99 %	+0,42 %	+0,97 %	0,79 % (± 0,33)
(B)	Consensus « top 50 % » sans le processus itératif	76,74 %	72,49 %	55,79 %	68,34 % (± 11,07)
(B)	Consensus « top 50 % » avec le processus itératif	85,75 %	83,94 %	69,08 %	79,58 % (± 9,15)
(B)	Gain approche consensus du «top 50 %»	+9,01 %	+11,44 %	+13,28 %	11,24 % (± 2,14)
(C)	Confirmation par le processus itératif	5336 (33,7 %)	10116 (29,7%)	12528 (21,9%)	28,4 % (± 6,0)
(C)	Tous faux (processus itératif inclus)	230 (1,5 %)	940 (2,5 %)	2046 (3,6 %)	2,5 % (± 1,1)
(C)	Uniquement le processus itératif faux	74 (0,3 %)	319 (0,9 %)	403 (0,7 %)	0,6 % (± 0,3)
(C)	Uniquement le processus itératif vrai	157 (1,0 %)	157 (0,4 %)	557 (1,0 %)	0,8 % (± 0,3)

D'un point de vue optimiste, 97,56 %, 97,07 %, 95,46 %, 95,46 % et 83,04 % des séquences avaient été binnés comme prévu par au moins un des outils pour S, M, L et $CAMI1h$, respectivement. Comparativement, les meilleurs résultats individuels pour les ensembles de données S, M, L et $CAMI1h$ ont atteint 92,20 % (CONCOCT), 90,99 % (CONCOCT), 83,44 % (CONCOCT) et 75,40 % (BinSanity). Cette vision optimiste pourrait grandement améliorer le binning correct des séquences de 5,36, 6,08, 12,02 et 7,64 points, respectivement, pour les ensembles de données S, M, L et $CAMI1h$ comparativement au meilleur outil seul. Un défi reste à relever dans la sélection du bon résultat parmi de multiples possibilités.

Une vision pessimiste exige que tous les outils trouvent le bon résultat, ce qui mène à des taux binning corrects de 34,16 %, 27,81 %, 22,20 % et 16,25 % pour les ensembles de données S, M, L et $CAMI1h$, respectivement. Toutefois, même si ces outils traitaient une partie limitée des données, on pourrait s'attendre à ce que les bins proposés soient très fiables.

Il est à noter que le processus itératif confirme en moyenne 28,4 % (± 6,0) des résultats proposés par toutes les autres méthodes testées. 2,5 % (± 1,1) des séquences des jeux de données n'ont jamais été correctement assignées à un bin par les outils existants. Le processus itératif propose un binning faux alors que toutes les autres méthodes trouvent le résultat attendu pour 0,6 % (± 0,3) des séquences mais il est le seul à en proposer un correct pour 0,8 % (± 0,3) des séquences.

5.2 Données réelles

5.2.1 Résultats du binning

Le binning des données métagénomiques du lac Pavin a été effectué avec les paramètres par défaut que ce soit pour l'extraction de clusters par le processus itératif basé sur `fennec` ou avec les outils MetaBAT2 ou CONCOCT.

Le jeu de données « Pavin 80m » n'est constitué que de 30430 fragments de contigs, ce qui a permis à tous les logiciels de proposer des résultats dans un temps considéré comme raisonnable – ici, moins de 30 jours. En revanche, CONCOCT n'a pas pu proposer de résultats dans ce délai pour le jeu de données « Pavin 65m » qui est composé de 168135 séquences. Ces temps de calcul sont explicables par les complexités algorithmiques des méthodes utilisées (voir chapitre 1).

Comme observé lors de l'étude comparative des logiciels de binning (chapitre 3), CONCOCT tend à assigner la plupart des séquences à un bin (99,84 %) tandis que

MetaBAT2 délaissera une grande partie des séquences (seuls 17,41 % et 23,57 % sont assignés pour les jeux de données 80m et 65m respectivement). Le processus itératif a permis l'assignation à un bin de 35,17 % et 35,16 % des séquences pour ces deux jeux de données respectifs. Ainsi, 8, 16 et 79 bins ont été construits à partir du métagénome « Pavin 80m » par le processus itératif, MetaBAT2 et CONCOCT respectivement. Le métagénome « Pavin 65m » a quant à lui permis la construction de 38 bins et 117 bins par le processus itératif et MetaBAT2 (tableau 5.5).

Il est à noter que le processus itératif est limité à 10 itérations maximum par défaut. Ainsi, les séquences non traitées au bout de ses 10 itérations sont labellisées « maxiter ». Il est toujours possible de relancer le processus uniquement sur ces données.

Tableau 5.5 | **Nombre de bins et proportion des séquences utilisées.** « * » : pas de résultats en moins de 30 jours.

	Pavin 65m	**Pavin 80m**
MetaBAT	117 bins (23,57 %)	16 bins (17,41 %)
CONCOCT	*	79 bins (99,84 %)
Processus itératif	38 bins (35,17 %)	8 bins (35,16 %)

5.2.2 Performances de calcul

La consommation en ressources informatiques a déjà été mesurée pour MetaBAT2 et CONCOCT, seules celles utilisées par le processus itératif ont été mesurées ici (tableau 5.6). Comme pour les jeux de données simulés vus précédemment, le processus itératif exploite l'architecture multi-processeurs tout en étant perfectible.

Tableau 5.6 | **Performances informatiques du processus itératif d'extraction de clusters** appliqué aux données du lac Pavin.

Jeu de données	**Nombre de fragments**	**Temps horloge (h:m:s)**	**Mémoire maximum (Mo)**	**Utilisation CPU (%)**
Pavin 80m	30430	58:33:40	120 169	1738
Pavin 65m	168135	487:46:42	3376519	1725

Les temps de calcul vont de 58 heures et 33 minutes à 487 heures et 46 minutes (soit environ 20 jours) pour les métagénomes du lac Pavin correspondant aux profondeurs de 80m et 65m respectivement. Ces temps de calcul restent importants mais nécessaires pour conserver une approche entièrement non supervisée. La quantité de mémoire utilisée, bien qu'importante, dépend en très grande partie de l'implémentation de scikit-learn qui s'adapte à l'environnement dans lequel il est exécuté. Les quantités de RAM indiquées peuvent donc ne pas être pertinentes pour d'autres exécutions dans d'autres environnements de calcul.

De par son approche itérative sur des sous-ensemble de données, le processus d'extraction de génomes permet l'utilisation de l'algorithme VBGMM sur des jeux de données que CONCOCT ne peut pas traiter avec ses paramètres par défaut.

5.2.3 Comparaison des résultats

En l'absence de données de référence, CheckM permet d'évaluer les bins en proposant le taux de complétude, de contamination et d'hétérogénéité de chacun d'entre eux puis de les classer en MAG selon les critères de Bowers *et al.* (2017).

MetaBAT2 ou CONCOCT proposent un nombre de MAG systématiquement plus important que le processus itératif basé sur `fennec` (tableau 5.7). En effet, le processus itératif propose des bins ayant une complétude médiane plus importante que ceux proposés par MetaBAT2 et CONCOCT mais la contamination des bins dépasse les critères d'acceptation des bins en MAG définis (figure 5.10).

Du fait de la distribution des abondances relatives des génomes dans les jeux de données de test, de nombreux bins de référence de quelques milliers de nucléotides viennent impacter négativement les taux de complétude et de contamination. Bien qu'il s'agisse dans les faits de génomes existants et identifiés dans nos jeux de données de tests, les métriques proposées par CheckM et les critères d'acceptation de Bowers *et al.* (2017) des bins en MAG ne permettent pas de reconstruire ces génomes. Ceci n'est pas observé pour les bins proposés par les outils dans la mesure où un nombre minimum de séquences par bins est fixé par ces outils.

La contamination des bins proposés par CONCOCT sur le jeu de données « Pavin 80m » atteint au maximum 165 % tandis que le processus itératif basé sur `fennec` atteint une contamination maximum pour les jeux de données simulés S, M, L et réels « Pavin 80m » et « Pavin 65m » de 117 %, 763 %, 547 %, 208 % et 649 % respectivement.

Une contamination provoquée par des organismes proches tend à se confirmer sur

Tableau 5.7 | **Nombre de bins, proportion de fragments de séquences traités et nombre de MAG**. Les critères d'acceptation des bins en tant que MAG sont détaillés au tableau 1.2. Les proportions de fragments de séquences traités pour le processus itératif correspondent aux proportion de séquences extraites (tableau 5.1).

Jeu de données	Logiciel	Traité (%)	HQ	MQ	LQ	Autre	Total
S	CONCOCT	99,97	0	9	18	18	45
	MetaBAT2	52,59	0	13	3	13	29
	Proc. itératif	60,33	0	5	2	10	17
M	CONCOCT	99,92	0	26	28	26	80
	MetaBAT2	51,55	0	26	9	23	58
	Proc. itératif	81,73	0	8	10	16	34
L	CONCOCT	99,90	0	23	50	50	123
	MetaBAT2	44,11	0	39	16	31	86
	Proc. itératif	57,30	0	5	10	27	42
Pavin 80m	CONCOCT	99,84	0	1	64	14	79
	MetaBAT2	17,41	0	4	8	4	16
	Proc. itératif	35,16	0	0	3	5	8
Pavin 65m	CONCOCT	–	–	–	–	–	–
	MetaBAT2	23,57	0	32	53	32	117
	Proc. itératif	34,19	0	2	13	23	38

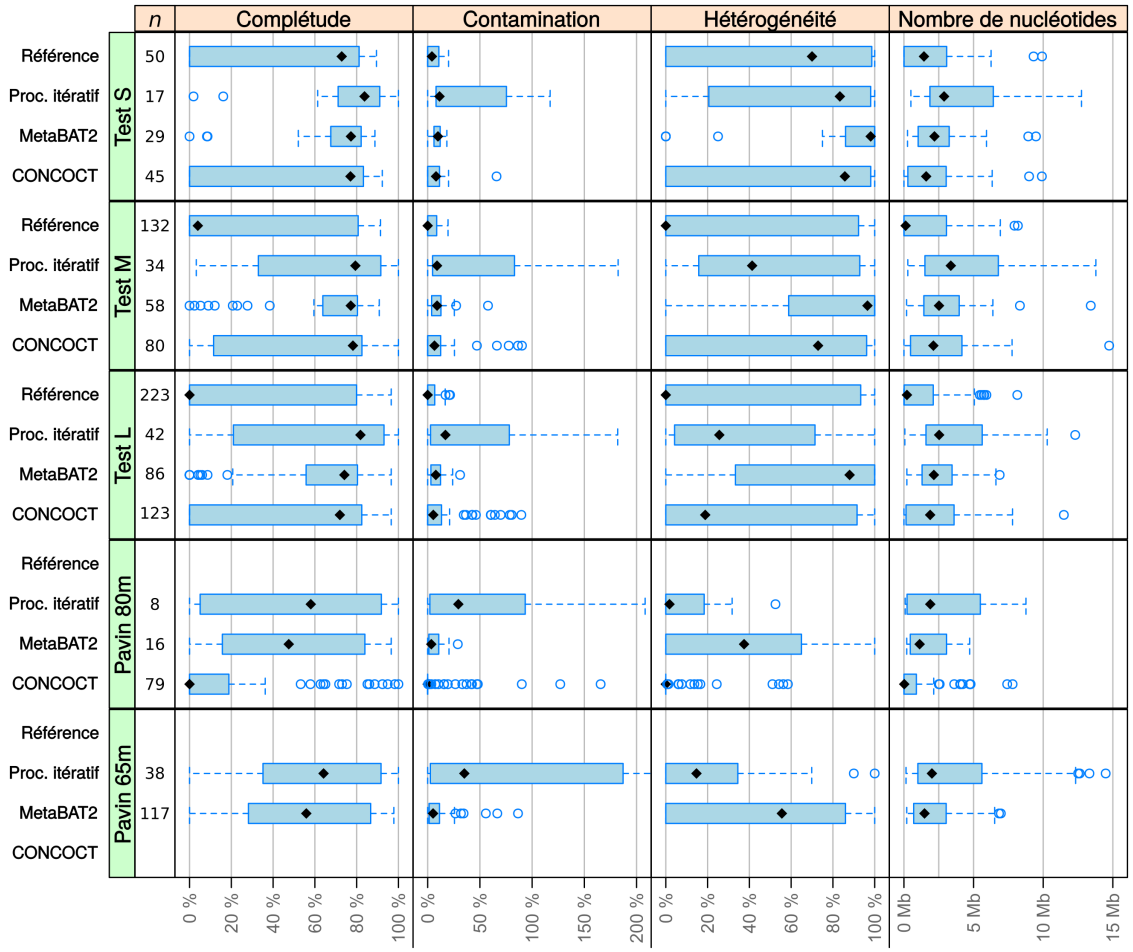


Figure 5.10 | **Comparaison des métriques d'évaluation des résultats de binning** et des tailles des bins en nombre de nucléotides par outil et avec les bins de référence. Le diamant noir symbolise la médiane de chaque distribution.

les données non simulées du lac Pavin avec une forte contamination mais une meilleure complétude et une bien moindre hétérogénéité de souche. Les bins reconstruits tendent à représenter des génomes composites plutôt que des MAG.

Cette contamination ne s'accompagne en revanche pas d'une augmentation de l'hétérogénéité de souches, ce qui peut contredire l'hypothèse de la reconstruction de génomes composites. Enfin, le processus itératif tend à proposer des bins ayant un plus grand nombre de nucléotides que les autres méthodes, même si moins de séquences sont utilisées (voir tableau 5.5).

5.2.4 Diversité et complémentarité des résultats

La diversité des résultats de binning proposés est estimée en se reposant sur l'estimation de la parenté entre les paires de bins grâce à l'ANI et représentée sous forme de graphes. Une version interactive est disponible à l'adresse suivante :

`https://keuv-grvl.github.io/thesis_data/chap_04/ani_graph`.

On observe que certains bins proposés par le processus itératif incluent plusieurs bins provenant d'autres logiciels pour le jeu de données « Pavin 80m » : le bin 1 (en vert) proposé par le processus itératif possède une ANI supérieure à 95 % avec 7 bins : les bins 23, 26, 27, 42, 43 et 77 proposés par CONCOCT et le bin 4 proposé par MetaBAT2 (figure 5.11). D'autre part, les bins 23, 26, 42, 43, 54, 77 et 78 proposés par CONCOCT ont une ANI supérieure à 95 % avec le bin 1 proposé par le processus itératif, montrant ainsi la non-réciprocité des ANI entre deux bins. On observe également 32 bins uniques, soit 41 % des bins proposés par CONCOCT pour le jeu de données « Pavin 80m » alors que tous les bins proposés par MetaBAT2 et le processus itératif ont une ANI supérieure ou égale à 95 % avec un autre bin parmi les 103 proposés par ces trois logiciels.

Cette propension du processus itératif à proposer des bins représentant des génomes composites est également observée pour le jeu de données « Pavin 65m » : le bin 4 proposé par le processus itératif possède une ANI supérieure à 95 % avec aucun autre bin. En revanche, les bins 5, 18, 38, 46, 49, 59, 61, 65, 68, 77 et 115 proposés par MetaBAT2 et le bin 33 proposé par le processus itératif ont une ANI supérieure à 95 % avec le bin 4 du processus itératif (figure 5.12). On observe également 9 bins proposés par le processus itératif et 27 bins proposés par MetaBAT2 qui n'ont pas d'ANI $\leq$ 95 % avec un autre bin parmi les 155 proposés par ces deux logiciels, soit 24 % et 23 % des bins proposés respectivement.

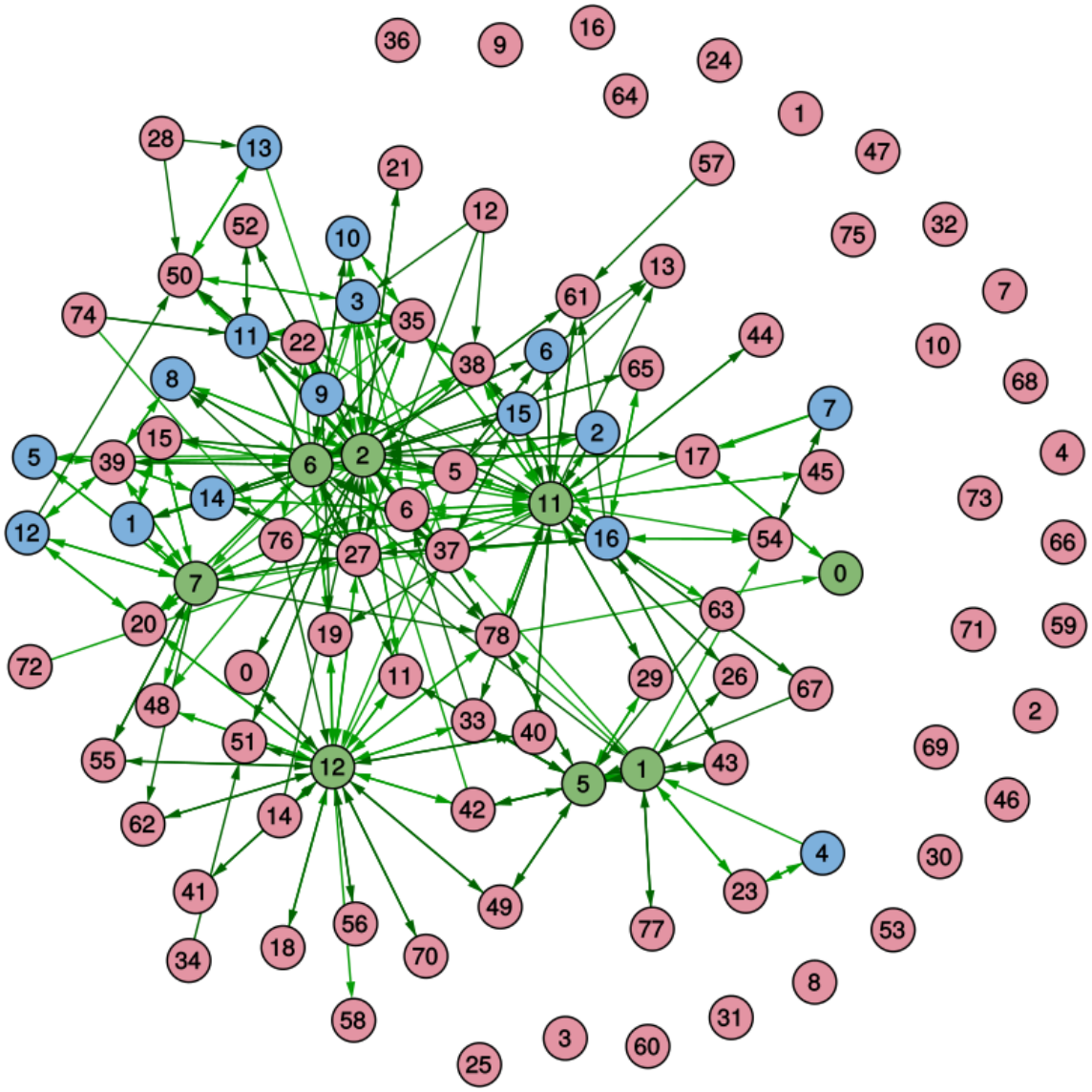


Figure 5.11 | **Graphe des ANI pour les bins reconstruits à partir des données « Pavin 80m »**. En rose, les bins proposés par CONCOCT ; en vert par le processus itératif ; et en bleu ceux proposés par MetaBAT2. Une flèche orientée d'un bin A vers un bin B indique que le bin A possède une ANI supérieure ou égale à 95 % avec le bin B (la réciproque n'est pas nécessairement vraie). Ainsi, le bin A est potentiellement inclus dans le bin B.

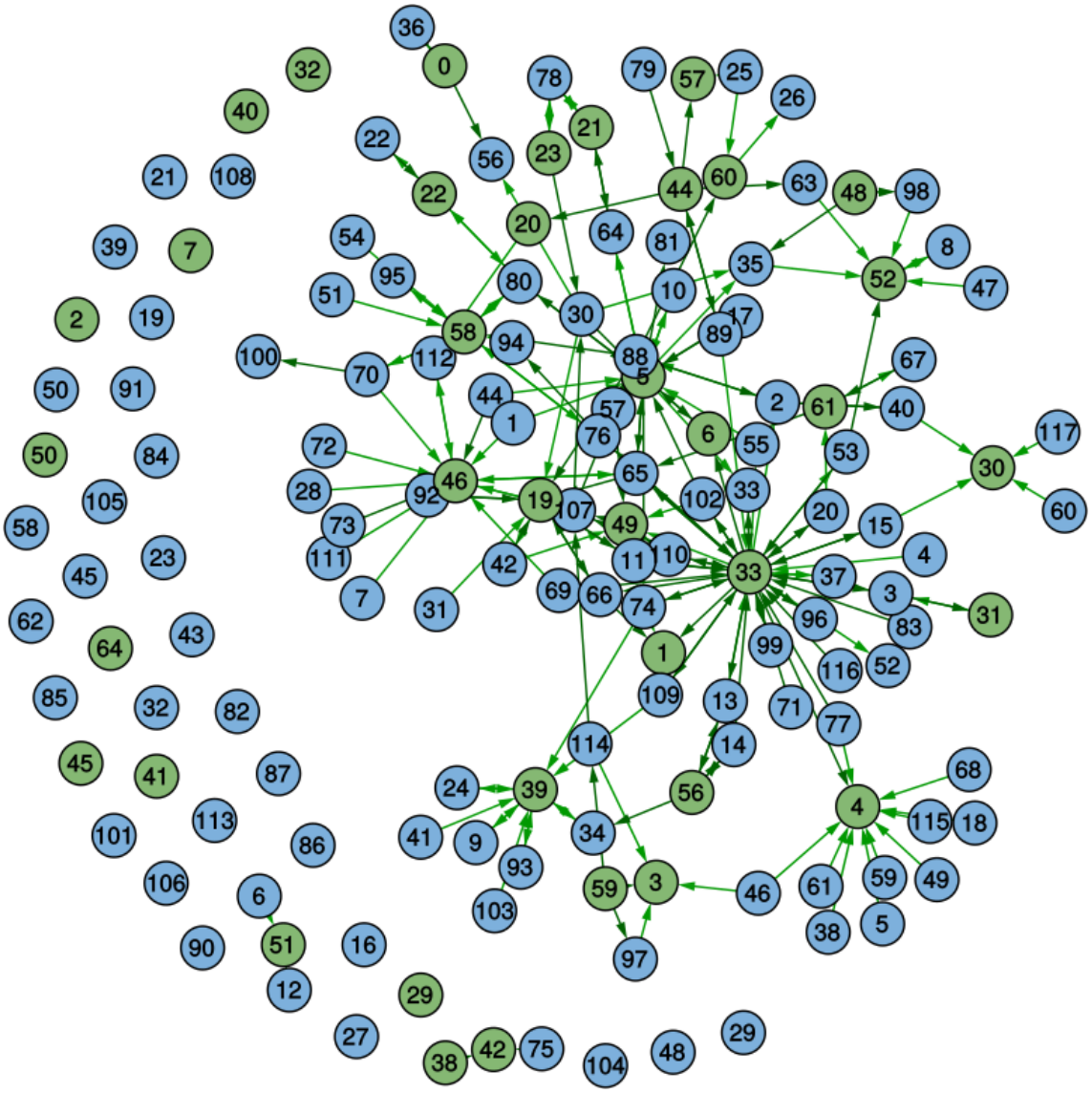
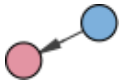
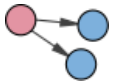
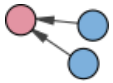
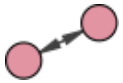


Figure 5.12 | **Graphe des ANI pour les bins reconstruits à partir des données « Pavin 65m »**. En vert, les bins proposés par le processus itératif ; et en bleu ceux proposés par MetaBAT2. Une flèche orientée d'un bin A vers un bin B indique que le bin A possède une ANI supérieure ou égale à 95 % avec le bin B (la réciproque n'est pas nécessairement vraie). Ainsi, le bin A est potentiellement inclus dans le bin B.

Tableau 5.8 | **Grille de lecture des graphes des ANI.**

Motif	Symbole	Exemples	Interprétations
Un bin n'est connecté à aucun autre		– Bin 36 proposé par CONCOCT (figure 5.11)	– Aucun autre bin n'inclut celui-ci et celui-ci n'inclut aucun autre bin. Ce bin n'a pas été reconstruit par les autres logiciels.
Un bin inclut un autre bin proposé par d'autres logiciels		– Bin 60 proposé par MetaBAT2 et bin 30 proposé par le processus itératif (figure 5.12)	– Le bin bleu a une ANI supérieure à 95 % avec le bin rose. Le bin bleu appartient donc à la même espèce que le bin rose, sans que cela soit réciproque, notamment si le bin bleu représente un groupe taxonomique.
Un bin est distribué entre plusieurs bins proposés par d'autres logiciels		– Bin 1 proposé par MetaBAT2 (figure 5.12)	– Le bin rose est un génome composite que d'autres outils ont réussi à reconstruire avec une meilleure résolution taxonomique. – Le bin rose est un bin dont les séquences peuvent être redistribuées dans plusieurs bins.
Un bin inclut plusieurs autres bins proposés par d'autres logiciels		– Bin 4 proposé par le processus itératif (figure 5.12)	– Le bin rose représente potentiellement un groupe taxonomique ou un génome composite tandis que les autres logiciels ont réussi à atteindre une meilleure résolution taxonomique avec les bins bleus.
Deux bins issus de deux logiciels différents sont mutuellement inclus		– Bin 23 proposé par CONCOCT et bin 4 proposé par MetaBAT2 (figure 5.11)	– Les deux logiciels ont reconstruit le même bin ou génome composite. Ces deux bins appartiennent donc à la même espèce selon les critères de Richter *et al.* (2008).
Deux bins issus du même logiciel sont mutuellement inclus		– Bins 11 et 2 proposés par le processus itératif (figure 5.11) – Bins 2 et 40 proposés par CONCOCT (figure 5.12)	– Le logiciel a proposé des bins représentant des génomes composites issus de groupes taxonomiques proches. – Un logiciel a séparé des séquences qui pourraient appartenir à la même espèce.

Une parfaite identité des résultats proposés par trois logiciels de binning montrerait des trinômes de bins mutuellement connectés et donc une absence de diversité dans les résultats. À l'inverse, une totale absence de liens entre les bins montrerait une absence de complémentarité dans les résultats proposés. Les figure 5.11 et figure 5.12 présentent plusieurs motifs présentés au tableau 5.8. On observe ainsi l'existence d'une diversité entre les résultats des binnings ainsi qu'une certaine complémentarité entre ceux-ci.

6 | Discussion

6.1 Étude comparative

6.1.1 Fiabilité des logiciels de binning non supervisé

La reconstruction de génome par binning non supervisé dépend fortement de la composition du métagénome étudié et de la conception de l'expérimentation. La diversification des jeux de données de référence pour conduire des études comparatives s'avère nécessaire.

L'étude comparative réalisée au chapitre 3 cible la ré-analyse de données à grande échelle. Ainsi, les résultats de cette étude peuvent ne pas être pertinents pour la reconstruction de génomes qui s'appuie sur des dispositions expérimentales spécifiques (*e. g.* : série temporelle).

L'intégration des résultats de $CAMI1h$ à notre étude comparative n'a pas modifié de façon significative les indicateurs de fiabilité. Les résultats de binning semblent néanmoins dépendre du jeu de données traité. En effet, MaxBin est le logiciel le plus fiable selon les expérimentations de Sieber *et al.* (2018) alors que les résultats de Meyer *et al.* (2018) désignent BinSanity comme étant le plus fiable.

CONCOCT et MetaBAT proposent d'excellents résultats sur nos jeux de données. Selon les critères de Bowers *et al.* (2017), CONCOCT reconstruit le plus de bins de haute qualité, ce qui permet l'étude approfondie de la physiologie des microorganismes. MetaBAT omet de nombreux contigs, ce qui le pénalise lorsque l'objectif est l'analyse fine d'un métagénome. De plus, les meilleurs outils sont performants sur les bactéries et archées mais peinent à reconstruire les eucaryotes et virus.

6.1.2 Évaluation des résultats

L'utilisation de communautés microbiennes artificielles comme référence nous permet une évaluation rigoureuse des résultats bien qu'elles sont certainement éloignées des communautés microbiennes réelles. En plus des indicateurs proposés par Parks *et al.* (2015), l'évaluation des résultats du binning d'un métagénome réel peut se reposer sur les méthodes suivantes.

— De nombreux génomes sont connus, notamment au sein du microbiote humain (Qin *et al.*, 2010; Pasolli *et al.*, 2019). Ceux-ci peuvent être utilisés comme référence pour une évaluation *a posteriori* de la fiabilité du binning. Cette stratégie n'est cependant pas applicable pour d'autres environnements où la diversité microbienne reste largement inconnue (*e. g.* : eaux, sol).

— Une possible stratégie d'évaluation consiste à incorporer artificiellement (« *spiking* ») des génomes connus aux métagénomes issus de ces environnements complexes. Le choix des génomes d'évaluation est particulièrement sensible dans la mesure où l'on introduit un potentiel biais dans le jeu de données, influençant possiblement la modélisation des données et donc le binning.

6.1.3 Axes pour l'amélioration du binning non supervisé

Un lien a été observé entre la qualité des bins produits par les logiciels testés et l'origine taxonomique de ces bins. Ce comportement suggère que certains logiciels de binning reconstruisent avec une plus grande fiabilité certains groupes taxonomiques.

Nous supposons que la représentation des contigs (choix de modèles et de la méthode d'intégration) joue un rôle majeur dans ce comportement. Un modèle donné peut en effet ne pas collecter les informations nécessaires pour considérer la totalité de la diversité microbienne, ou diluer cette information dans un excès de données. Or, tous les groupes taxonomiques des données de test peuvent être reconstruits, au moins partiellement. Le développement d'un modèle spécifique pour répondre aux lacunes des modèles existants n'apparait donc pas pertinent.

À l'inverse, les approches par aggrégation de multiples résultats semblent particulièrement pertinentes. Sieber *et al.* (2018) implémente cette stratégie en se reposant sur des critères exclusivement biologiques. Exploiter la diversité et la complémentarité des méthodes peut alors être perçu comme un problème de clustering par consensus (Vega-Pons et Ruiz-Shulcloper, 2011; Aggarwal et Reddy, 2013; Xanthopoulos, 2014). Une telle approche requiert cependant le développement de nouvelles modélisations des contigs pour augmenter la diversité nécessaire à la pertinence des consensus (Pividori *et al.*, 2016).

6.2 Modélisation intégrative et adaptative de contigs

Il a donc été proposé un premier outil de modélisation de contigs métagénomiques – `fennec` – ayant pour but, en plus de proposer différents modèles de données existants au sein d'un même outil, de permettre simplement sa réutilisation (chapitre 4). Il s'agit à notre connaissance du seul outil dédié à la représentation non supervisée des contigs métagénomiques en vue de leur traitement par apprentissage automatique.

6.2.1 Modèles de données alternatifs

L'ajout de nouvelles modélisations est anticipé. Ceci permettra l'utilisation de la corrélation intrinsèque des oligonucléotides (Ding *et al.*, 2015) et du biais d'utilisation du code génétique (Yu *et al.*, 2018) au sein de `fennec`, à condition que ces modélisations permettent le calcul de distances cosinus.

Le modèle de connaissance utilisé par MetaBAT laisse à penser que celui-ci est porteur d'informations pertinentes au vu de l'étude comparative. Cependant, il demeure un cas particulier dans la mesure où il n'est pas réutilisable par d'autres outils que MetaBAT, ni reproductible. De ces faits, l'implémentation de cette méthode au sein de `fennec` n'est pas prévue.

6.2.2 Intégration des modélisations brutes

Les normalisations spécifiques aux modèles de données sont limitantes pour l'augmentation de la diversité des approches de binning. Celles-ci sont rendues dispensables grâce à l'utilisation de la distance cosinus. D'autre part, l'idée d'une représentation numérique compacte des contigs et adaptée au jeu de données provient de Alneberg *et al.* (2014). De fait, l'intégration adaptative de multiples modélisations de contigs par une ACP à noyau cosinus s'impose.

6.2.3 Utilisation de réseaux de neurones

Le modèle Contig2Vec est proposé au sein de `fennec`. Ce dernier repose sur un réseau de neurones artificiels non supervisé dédié à la modélisation de mots. La forte contribution constatée (chapitre 4) de ce modèle à la modélisation intégrée des contigs confirme que les méthodes de traitement du langage naturel contribuent à la métagénomique et la bioinformatique plus largement (Zeng *et al.*, 2015; Ditzler *et al.*, 2015; Nauman *et al.*, 2019). Il s'agit à notre connaissance de la première utilisation d'une telle méthode dans le cadre du binning non supervisé.

6.2.4 Évaluation de la pertinence d'une modélisation

La modélisation intégrée des contigs ne remplit pas d'objectif en soi mais sert de point d'entrée à un algorithme de clustering. La pertinence de ce dernier peut être quantifiée alors qu'il n'existe pas d'outil pour quantifier la pertinence d'une modélisation des contigs, en particulier avec une approche adaptative. En effet,

seule une évaluation visuelle est aujourd'hui possible, à l'aide d'une projection bidimensionnelle. Ainsi, la séparation des étapes de modélisation et de clustering, bien qu'effective, ne remplit que partiellement sa mission puisque le bénéfice de la modélisation proposée ne peut être quantifié.

6.2.5 Autre stratégie d'intégration

La stratégie d'intégation utilisée ici permet d'éviter l'explosion du nombre d'attributs à manipuler. Une approche plus directe consisterait à appliquer différents modèles bruts, à les juxtaposer, puis à appliquer une ACP à noyau. Cependant, le nombre d'attributs à manipuler peut rapidement augmenter, ce qui n'est pas souhaitable au vu de la complexité algorithmique temporelle de l'ACP et du contexte de ré-analyse de données à grande échelle qui est visée.

6.2.6 Choix de l'écosystème logiciel

L'apprentissage automatique étant un domaine en plein essor dans toutes les disciplines, de nombreuses bibliothèques sont proposées pour l'application de ces méthodes. Le choix s'est ici porté sur Python 3 et numpy/scipy/sckit-learn de par leur extensibilité et leur interfaçabilité à d'autres langages, notamment R, ou formats, comme HDF5.

Des langages de programmation bas niveau peuvent être envisagés pour gagner en performance, mais au prix d'une réutilisabilité moindre par la communauté concernée. Il en est de même pour des outils axés sur la distribution des calculs, comme MLlib d'Apache Spark. Cependant, de nouvelles méthodes de binning doivent être proposées pour permettre les approches de binning par consensus avant d'être distribuées.

6.3 Extraction itérative de clusters

6.3.1 Stratégie d'extraction

Les approches de clustering itératif existent notamment dans l'algorithme G-means. Chaque itération de cet algorithme réutilise les mêmes représentations des données. Or, une représentation doit pouvoir changer en fonction de l'ensemble de données à traiter de façon à sélectionner les caractéristiques pertinentes.

À la différence du G-means où chaque cluster produit est à nouveau clusterisé, le processus itératif développé ici extrait les clusters satisfaisant certains critères. Cette

approche « en arbre » pourrait davantage se rapprocher de la logique de la taxonomie des êtes vivants, pour ainsi tenter de reconstruire *ab initio* l'arbre phylogénétique des individus en présence. Cette approche reste néanmoins implémentable avec les outils proposés : un cluster produit et validé par le logiciel est à nouveau traité indépendamment du reste des données.

6.3.2 Semi-supervision

Le choix de l'algorithme de clustering s'est ici porté sur VBGMM du fait de sa performance lors de l'étude comparative (chapitre 3) et de la non-nécessité de paramètres difficilement estimables (*e. g.* : nombre attendu de bins).

Sa semi-supervision est implémentée comme une consigne pour l'initialisation de l'algorithme, mais le cœur de l'algorithme peut ne pas respecter les contraintes. Cela pourrait avoir des effets bénéfiques liés à la présence de chimères après l'assemblage des lectures de séquençage, mais la fiabilité des outils d'assemblage actuels tendrait à rendre ce bénéfice marginal.

La conception de `fennec` permet en outre d'utiliser un algorithme de clustering différent de manière très simple. Cet algorithme pourra alors choisir de prendre en compte les contraintes *must-link* de manière douce (comme actuellement implémenté) ou de manière dure (sous forme de contraintes inviolables) comme le propose la méthode hiérarchique `Agglomerative Clustering` de la bibliothèque scikit-learn.

D'autre part, l'utilisation d'un clustering semi-supervisé est particulièrement compatible à l'adjonction de génomes de référence dans le métagénome (« *spiking* »). En effet, ces génomes connus et soigneusement sélectionnés doivent être idéalement retrouvés dans le résultat final du binning. Ces contraintes supplémentaires pourraient alors guider le clustering mais aussi la modélisation des contigs (ce point est discuté après).

6.3.3 Critères d'arrêt du processus itératif

Le processus itératif proposé repose sur une série de conditions d'arrêts établies pour des raisons pratiques (*e. g.* : éviter les boucles infinies) mais aussi empiriquement. Le choix a été fait de laisser la possibilité à l'utilisateur de modifier ces différents paramètres pour éviter des conditions d'arrêt qui seraient délétères pour la reconstruction des génomes, mais cela nécessite de mener une optimisation de ces paramètres.

L'information des violations des contraintes *must-link* après chaque clustering peut également devenir un nouveau critère d'évaluation et de décision non supervisée afin de déterminer la fiabilité de ce clustering. Des clusters respectant très bien les relations *must-link* pourraient être jugés fiables tandis que des clusters ayant de nombreux liens entre eux au travers des *must-link* pourraient être jugés comme peu fiables, auquel cas le processus itératif serait interrompu.

D'autres critères d'arrêt peuvent également être ajoutés, notamment le critère de « clusterabilité » d'un jeu de données (Ackerman *et al.* (2016)). La majorité des algorithmes de clustering perdent en performance lorsqu'ils sont confrontés à des données qui ne sont pas clusterisables (*e. g.* : bruit). Éviter de clusteriser ce type de données permettrait de gagner du temps de calcul mais également de gagner en qualité. Cependant, très peu de méthodes ont été proposées à l'heure actuelle (Ackerman *et al.*, 2016) et leur robustesse n'est pas démontrée.

6.3.4 Temps de calcul

Les différentes étapes d'une itération du processus d'extraction de clusters utilisent des méthodes hautement parallélisables en se reposant sur des outils optimisés (Pedregosa *et al.*, 2011). De plus, à chaque itération, les valeurs propres des modélisations sont calculées à partir d'un échantillon aléatoire de données afin de limiter l'explosion des temps de calcul.

Cependant, avant chaque clustering, il est nécessaire de réorganiser la matrice contenant les relations *must-link*. Cette étape a été optimisée avec les outils à disposition mais reste non parallèle alors qu'elle constitue une part importante du temps consacré à chaque itération. La résolution de ce problème nécessiste l'intervention de développeurs spécialisés afin de garantir les meilleures performances.

6.4 Qualité des génomes reconstruits

6.4.1 Potentiel d'une approche consensuelle

Les critères pour comparer les résultats de binning sans référence sont aujourd'hui purement algorithmiques (*i. e.* : identifiant du contig par Song et Thomas (2017)) ou biologiques (*i. e.* : utilisation de marqueurs comme Sieber *et al.* (2018)). Aucun n'est donc réellement satisfaisant pour une approche non supervisée.

L'ANI est un des outils envisageables pour la comparaison non supervisée des

bins et la construction de consensus. Plusieurs méthodes qui proposent des bins ayant des ANI élevées tendent à confirmer la validité de ces bins. Cependant, l'ANI entre les bins permet seulement d'estimer la diversité des résultats de binning proposés par plusieurs logiciels mais ne permet pas d'estimer la complémentarité de ceux-ci. De plus, l'ANI n'est pas une distance et reste sensible à la taille des génomes qui sont comparés, ce qui peut la rendre inadaptée dans le cadre du binning où les bins peuvent avoir des tailles très variables.

L'utilisation de la diversité et de la complémentarité des bins pourrait être considérée comme un problème de clustering par consensus, problème pour lequel l'apprentissage machine apporte de multiples algorithmes regroupés dans l'« *ensemble learning* ». Une telle approche nécessitera la mise au point de nouvelles méthodes de binning pour améliorer la diversité nécessaire à ce type d'approche (Pividori *et al.*, 2016).

6.4.2 Comprendre leur contamination

Lorque l'outil proposé ici est considéré individuellement, les bins produits souffrent d'un fort taux de contamination et ne satisfont donc pas les critères de Bowers *et al.* (2017). Ils ne peuvent donc être considérés comme des MAG, malgré de meilleurs taux de complétude et un taux d'hétérogénéité de souche étonnamment inférieur par rapport à MetaBAT2 et CONCOCT.

Les raisons de ce résultat doivent être établies pour améliorer la fiabilité de l'outil présenté ici, lorsqu'il est utilisé seul. Or, l'absence d'outil pour l'évaluation quantitative des modélisations des données rend cette investigation difficile. Des travaux supplémentaires sont donc nécessaires pour comprendre ce problème.

7 | Conclusion

Nous nous proposions de développer une approche originale de binning non supervisé. Pour ce faire, une étude comparative a été réalisée pour mieux cerner les méthodes existantes. Une modélisation non supervisée et adaptative des contigs métagénomiques est ensuite proposée. Cette dernière est utilisée pour l'extraction itérative de clusters, puis appliquée à divers jeux de données de test et réels.

7.1 Comparaison des outils de binning

Le binning de séquences métagénomiques étant une méthode relativement récente, peu de comparatifs et peu de données de référence sont disponibles pour la communauté.

Plusieurs méthodes de binning ont été sélectionnées de façon à couvrir un large panel des modélisations des séquences ainsi que des algorithmes de clustering existants d'une part. Par ailleurs, trois métagénomes aux compositions et complexités variées ont été proposés à des fins d'évaluation, en complément notamment de ceux utilisés par le challenge CAMI (Sczyrba *et al.*, 2017).

Ce comparatif indique en premier lieu que les logiciels MetaBAT2 et CONCOCT apparaissent les plus performants pour reconstruire des génomes. MetaBAT n'exploitant qu'une partie des données, il sera préféré pour une première analyse d'un métagénome inconnu du fait de sa rapidité. CONCOCT qui est bien plus lent permet à l'inverse une exploration plus profonde du métagénome en reconstruisant un plus grand nombre de génomes.

En deuxième lieu, ce benchmark a permis de mettre en évidence un lien de dépendance entre la méthode utilisée et la qualité des MAG obtenus, et ce à tous les niveaux taxonomiques. Ainsi, le choix de l'outil peut s'avérer critique pour l'étude d'un écosystème donné.

Enfin, la diversité et la complémentarité des différents résultats de binning a permis de montrer l'apport significatif des approches de binning par consensus pour nos jeux de données de composition connue. Les approches consensuelles existantes souffrent cependant d'un manque de diversité dans les approches de binning qui leur sont nécessaires pour devenir performantes.

Les approches de binning par consensus telles qu'elles sont développées sont très sensibles aux performances des outils sous-jacents (comme montré au tableau 3.1). Notre hypothèse est qu'elles n'intègrent probablement que trop peu de modèles de connaissance (*i. e.* : modélisations des contigs).

7.2 Modélisation non supervisée, intégrative et adaptative des contigs

Ainsi, deux modèles de connaissance connus et largement utilisés pour le binning, trois modèles ayant montré leur pertinence pour d'autres applications bioinforma-

tiques et un modèle original exploitant des méthodes du traitement du langage naturel ont été réunis au sein d'un même outil. Aucun de ces modèles ne fait appel à des connaissances préalables.

Les modèles choisis par l'utilisateur sont appliqués séparement puis intégrés de manière à proposer une représentation numérique de chaque séquence. La méthode d'intégration est elle aussi non supervisée. De plus, elle permet de s'affranchir d'*a priori* sur les données, notamment en ne requérant plus de normalisations spécifiques *via* l'utilisation de l'astuce du noyau.

L'utilisation de méthodes d'extraction de caractéristiques comme l'ACP permet en plus de rendre la modélisation des données adaptée au jeu de données desquelles elles sont issues. On met ainsi en avant les caractéristiques les plus discriminantes pour représenter les contigs.

Cet outil est conçu de manière à être modulable (`https://github.com/keuv-grvl/fennec/`), permettant ainsi de développer plus rapidement de nouveaux outils de binning. Ces représentations des séquences peuvent en effet être utilisées pour n'importe quelle application d'apprentissage automatique. La visualisation de données et le clustering sont ici utilisés mais peuvent être remplacés par d'autres méthodes (*e. g.* : classification, régression).

7.3 Processus itératif d'extraction de clusters

L'adaptabilité dans la représentation des contigs permise par `fennec` a été mise en application dans un processus itératif de reconstruction de génomes.

Celui-ci repose sur l'algorithme de clustering VBGMM qui a été modifié pour devenir semi-supervisé. En effet, les pré-traitements usuels des séquences métagénomiques (assemblage puis fragmentation des contigs) sont porteurs d'informations inutilisées par les logiciels de binning. Cette fragmentation étant maitrisée, l'origine des fragments de contigs devient un ensemble de contraintes que l'algorithme de clustering peut prendre en compte.

Les clusters obtenus à partir de la modélisation des fragments de contigs sont analysés par le score *silhouette* automatiquement et de manière non supervisée pour déterminer lesquels sont suffisament distincts du reste des données pour être extraits.

Les modélisations des données non extraites sont ré-intégrées pour s'adapter à ce nouvel ensemble de données à traiter et le processus est répété. Ainsi, la représentation des données varie à chaque itération selon le contexte, permettant de traiter par clustering uniquement les informations les plus discriminantes.

7.4 Applications

La méthode de binning développée a été évaluée sur des jeux de données : (i) de test, ayant servi à la comparaison des outils existants ; (ii) réels, issus d'un métagénome lacustre.

L'application aux communautés artificielles permet d'obtenir des résultats globalement peu différents de ceux obtenus avec les meilleurs logiciels de binning selon les critères retenus dans ce travail. L'application du binning sur les assemblages de métagénomes lacustres ne peut être comparée aux autres obtenus avec les logiciels de référence (CONCOCT et MetaBAT2). Sur le seul critère de l'évaluation de la qualité des MAG, les bins proposés par la méthode itérative donnent de moins bons résultats.

L'étude de la complémentarité des bins proposés par ces trois méthodes à travers un graphe d'ANI a néanmoins permis d'ouvrir des pistes quant aux possibilités de reconstruction de génomes par consensus à partir de métagénomes réels.

L'intégration de cette nouvelle stratégie de binning dans une approche consensuelle par vote majoritaire permet alors d'améliorer la reconstruction des génomes.

Bibliographie

[Abubucker *et al.* 2012] Abubucker, Sahar ; Segata, Nicola ; Goll, Johannes ; Schubert, Alyxandria M. ; Izard, Jacques ; Cantarel, Brandi L. ; Rodriguez-Mueller, Beltran ; Zucker, Jeremy ; Thiagarajan, Mathangi ; Henrissat, Bernard ; White, Owen ; Kelley, Scott T. ; Methé, Barbara ; Schloss, Patrick D. ; Gevers, Dirk ; Mitreva, Makedonka ; Huttenhower, Curtis : Metabolic Reconstruction for Metagenomic Data and Its Application to the Human Microbiome. 8 (2012), Num. 6. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3374609/. – Visité le : 2018-12-03. – ISSN : 1553-734X

[Ackerman *et al.* 2016] Ackerman, Margareta ; Adolfsson, Andreas ; Brownstein, Naomi : An Effective and Efficient Approach for Clusterability Evaluation. (2016). – URL : http://arxiv.org/abs/1602.06687. – Visité le : 2017-11-02

[Aggarwal et Reddy 2013] Aggarwal, Charu C. ; Reddy, Chandan K. : *Data Clustering : Algorithms and Applications*. URL : https://www.crcpress.com/Data-Clustering-Algorithms-and-Applications/Aggarwal-Reddy/p/book/9781466558212. – Visité le : 2017-10-24, 2013. – ISBN : 91466558210

[Allali *et al.* 2017] Allali, Imane ; Arnold, Jason W. ; Roach, Jeffrey ; Cadenas, Maria B. ; Butz, Natasha ; Hassan, Hosni M. ; Koci, Matthew ; Ballou, Anne ; Mendoza, Mary ; Ali, Rizwana ; Azcarate-Peril, M. A. : A comparison of sequencing platforms and bioinformatics pipelines for compositional analysis of the gut microbiome. 17 (2017), Num. 1, p. 194. – URL : https://doi.org/10.1186/s12866-017-1101-8. – Visité le : 2018-07-25. – ISSN : 1471-2180

[Alneberg *et al.* 2014] Alneberg, Johannes ; Bjarnason, Brynjar S. ; Bruijn, Ino de ; Schirmer, Melanie ; Quick, Joshua ; Ijaz, Umer Z. ; Lahti, Leo ; Loman, Nicholas J. ; Andersson, Anders F. ; Quince, Christopher : Binning metagenomic contigs by coverage and composition. 11 (2014), Num. 11, p. 1144–1146. – URL : http://www.nature.com/doifinder/10.1038/nmeth.3103. – Visité le : 2016-02-05. – ISSN : 1548-7091, 1548-7105

[Altschul *et al.* 1990] Altschul, Stephen F. ; Gish, Warren ; Miller, Webb ; Myers, Eugene W. ; Lipman, David J. : Basic local alignment search tool. 215 (1990), Num. 3, p. 403–410. – URL : http://linkinghub.elsevier.com/retrieve/pii/S0022283605803602. – Visité le : 2018-01-10. – ISSN : 00222836

[Angerer *et al.* 2017] Angerer, Philipp ; Simon, Lukas ; Tritschler, Sophie ; Wolf, F. A. ; Fischer, David ; Theis, Fabian J. : Single cells make big data : New challenges and opportunities in transcriptomics. 4 (2017), p. 85–91. – URL : http://www.sciencedirect.com/science/article/pii/S245231001730077X. – Visité le : 2018-08-03. – ISSN : 2452-3100

[Angly *et al.* 2012] Angly, Florent E. ; Willner, Dana ; Rohwer, Forest ; Hugenholtz, Philip ; Tyson, Gene W. : Grinder : a versatile amplicon and shotgun sequence simulator. 40 (2012), Num. 12, p. e94–e94. – URL : http://nar.oxfordjournals.org/content/40/12/e94. – Visité le : 2016-02-11. – ISSN : 0305-1048, 1362-4962

[Antoine 2011] Antoine, Violaine : *Intégration de contraintes en classification automatique évidentielle*. 2011. – URL : https://www.theses.fr/2011COMP1976

[Antoine *et al.* 2018] Antoine, Violaine ; Gravouil, Kévin ; Labroche, Nicolas : On Evidential Clustering with Partial Supervision. Dans : Destercke, Sébastien (Éditeur) ; Denoeux, Thierry (Éditeur) ; Cuzzolin, Fabio (Éditeur) ; Martin, Arnaud (Éditeur) : *Belief Functions : Theory and Applications*, Springer International Publishing, 2018 (Lecture Notes in Computer Science), p. 14–21. – ISBN : 978-3-319-99383-6

[Antoine *et al.* 2012] Antoine, Violaine ; Quost, Benjamin ; Masson, Marie-Hélène ; Denoeux, Thierry : CECM : Constrained evidential C-means algorithm. 56 (2012), p. 894–914

[Arabidopsis Genome Initiative 2000] Arabidopsis Genome Initiative : Analysis of the genome sequence of the flowering plant Arabidopsis thaliana. 408 (2000), Num. 6814, p. 796–815. – ISSN : 0028-0836

[Arora *et al.* 2016] Arora, Sanjeev ; Liang, Yingyu ; Ma, Tengyu : A simple but tough-to-beat baseline for sentence embeddings. (2016), p. 16. – URL : https://openreview.net/forum?id=SyK00v5xx

[Arthur et Vassilvitskii 2007] Arthur, David ; Vassilvitskii, Sergei : K-means++ : The Advantages of Careful Seeding. Dans : *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Philadelphia, PA, USA : Society for Industrial and Applied Mathematics, 2007 (SODA '07), p. 1027–1035. – URL : http://dl.acm.org/citation.cfm?id=1283383.1283494. – ISBN : 978-0-898716-24-5

[Bair 2013] Bair, Eric : Semi-supervised clustering methods. 5 (2013), Num. 5, p. 349–361. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3979639/. – ISSN : 1939-5108

[Barnett *et al.* 2011] Barnett, Derek W. ; Garrison, Erik K. ; Quinlan, Aaron R. ; Strömberg, Michael P. ; Marth, Gabor T. : BamTools : a C++ API and toolkit for analyzing and managing BAM files. 27 (2011), Num. 12, p. 1691–1692. – URL : https://academic.oup.com/bioinformatics/article/27/12/1691/255399. – Visité le : 2018-08-17. – ISSN : 1367-4803

[Basu *et al.* 2008] Basu, Sugato ; Davidson, Ian ; Wagstaff, Kiri : *Constrained Clustering : Advances in Algorithms, Theory, and Applications*. 1. Chapman & Hall/CRC, 2008. – ISBN : 978-1-58488-996-0

[Batut *et al.* 2018] Batut, Bérénice ; Gravouil, Kévin ; Defois, Clémence ; Hiltemann, Saskia ; Brugère, Jean-François ; Peyretaillade, Eric ; Peyret, Pierre : ASaiM : a Galaxy-based framework to analyze microbiota data. 7 (2018), Num. 6. – URL : https://academic.oup.com/gigascience/article/7/6/giy057/5001424. – Visité le : 2018-10-08

[Birol *et al.* 2015] Birol, Inanç ; Chu, Justin ; Mohamadi, Hamid ; Jackman, Shaun D. ; Raghavan, Karthika ; Vandervalk, Benjamin P. ; Raymond, Anthony ; Warren, René L. : Spaced Seed Data Structures for *De Novo* Assembly. 2015 (2015), p. 1–8. – URL : http://www.hindawi.com/journals/ijg/2015/196591/. – Visité le : 2016-02-05. – ISSN : 2314-436X, 2314-4378

[Bishop 2006] Bishop, Christopher : *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006 (Information Science and Statistics, Springer Nature Textbooks – HE site). – URL : https://www.springer.com/us/book/9780387310732. – Visité le : 2018-08-12. – ISBN : 978-0-387-31073-2

[Boisvert *et al.* 2012] Boisvert, Sébastien ; Raymond, Frédéric ; Godzaridis, Elénie ; Laviolette, François ; Corbeil, Jacques : Ray Meta : scalable de novo

metagenome assembly and profiling. 13 (2012), Num. 12, p. R122. – ISSN : 1474-760X

[Bonet *et al.* 2017] Bonet, Isis ; Escobar, Adriana ; Mesa-Múnera, Andrea ; Alzate, Juan F. : Consensus Clustering for Binning Metagenome Sequences. Dans : Pichardo-Lagunas, Obdulia (Éditeur) ; Miranda-Jiménez, Sabino (Éditeur) : *Advances in Soft Computing* Vol. 10062. Springer International Publishing, 2017, p. 273–284. – URL : http://link.springer.com/10.1007/978-3-319-62428-0_23. – Visité le : 2018-07-18. – ISBN : 978-3-319-62427-3 978-3-319-62428-0

[Bowers *et al.* 2017] Bowers, Robert M. ; Kyrpides, Nikos C. ; Stepanauskas, Ramunas ; Harmon-Smith, Miranda ; Doud, Devin ; Reddy, T B K. ; Schulz, Frederik ; Jarett, Jessica ; Rivers, Adam R. ; Eloe-Fadrosh, Emiley A. ; Tringe, Susannah G. ; Ivanova, Natalia N. ; Copeland, Alex ; Clum, Alicia ; Becraft, Eric D. ; Malmstrom, Rex R. ; Birren, Bruce ; Podar, Mircea ; Bork, Peer ; Weinstock, George M. ; Garrity, George M. ; Dodsworth, Jeremy A. ; Yooseph, Shibu ; Sutton, Granger ; Glöckner, Frank O. ; Gilbert, Jack A. ; Nelson, William C. ; Hallam, Steven J. ; Jungbluth, Sean P. ; Ettema, Thijs J G. ; Tighe, Scott ; Konstantinidis, Konstantinos T. ; Liu, Wen-Tso ; Baker, Brett J. ; Rattei, Thomas ; Eisen, Jonathan A. ; Hedlund, Brian ; McMahon, Katherine D. ; Fierer, Noah ; Knight, Rob ; Finn, Rob ; Cochrane, Guy ; Karsch-Mizrachi, Ilene ; Tyson, Gene W. ; Rinke, Christian ; Kyrpides, Nikos C. ; Schriml, Lynn ; Garrity, George M. ; Hugenholtz, Philip ; Sutton, Granger ; Yilmaz, Pelin ; Meyer, Folker ; Glöckner, Frank O. ; Gilbert, Jack A. ; Knight, Rob ; Finn, Rob ; Cochrane, Guy ; Karsch-Mizrachi, Ilene ; Lapidus, Alla ; Meyer, Folker ; Yilmaz, Pelin ; Parks, Donovan H. ; Eren, A M. ; Schriml, Lynn ; Banfield, Jillian F. ; Hugenholtz, Philip ; Woyke, Tanja : Minimum information about a single amplified genome (MISAG) and a metagenome-assembled genome (MIMAG) of bacteria and archaea. Dans : *Nature Biotechnology* 35 (2017), aug, Num. 8, p. 725–731. – URL : https://doi.org/10.1038/nbt.3893

[Bradnam *et al.* 2013] Bradnam, Keith R. ; Fass, Joseph N. ; Alexandrov, Anton ; Baranay, Paul ; Bechner, Michael ; Birol, Inanç ; Boisvert, Sébastien ; Chapman, Jarrod A. ; Chapuis, Guillaume ; Chikhi, Rayan ; Chitsaz, Hamidreza ; Chou, Wen-Chi ; Corbeil, Jacques ; Del Fabbro, Cristian ; Do-

cking, T. R. ; Durbin, Richard ; Earl, Dent ; Emrich, Scott ; Fedotov, Pavel ; Fonseca, Nuno A. ; Ganapathy, Ganeshkumar ; Gibbs, Richard A. ; Gnerre, Sante ; Godzaridis, Elénie ; Goldstein, Steve ; Haimel, Matthias ; Hall, Giles ; Haussler, David ; Hiatt, Joseph B. ; Ho, Isaac Y. ; Howard, Jason ; Hunt, Martin ; Jackman, Shaun D. ; Jaffe, David B. ; Jarvis, Erich D. ; Jiang, Huaiyang ; Kazakov, Sergey ; Kersey, Paul J. ; Kitzman, Jacob O. ; Knight, James R. ; Koren, Sergey ; Lam, Tak-Wah ; Lavenier, Dominique ; Laviolette, François ; Li, Yingrui ; Li, Zhenyu ; Liu, Binghang ; Liu, Yue ; Luo, Ruibang ; Maccallum, Iain ; Macmanes, Matthew D. ; Maillet, Nicolas ; Melnikov, Sergey ; Naquin, Delphine ; Ning, Zemin ; Otto, Thomas D. ; Paten, Benedict ; Paulo, Octávio S. ; Phillippy, Adam M. ; Pina-Martins, Francisco ; Place, Michael ; Przybylski, Dariusz ; Qin, Xiang ; Qu, Carson ; Ribeiro, Filipe J. ; Richards, Stephen ; Rokhsar, Daniel S. ; Ruby, J. G. ; Scalabrin, Simone ; Schatz, Michael C. ; Schwartz, David C. ; Sergushichev, Alexey ; Sharpe, Ted ; Shaw, Timothy I. ; Shendure, Jay ; Shi, Yujian ; Simpson, Jared T. ; Song, Henry ; Tsarev, Fedor ; Vezzi, Francesco ; Vicedomini, Riccardo ; Vieira, Bruno M. ; Wang, Jun ; Worley, Kim C. ; Yin, Shuangye ; Yiu, Siu-Ming ; Yuan, Jianying ; Zhang, Guojie ; Zhang, Hao ; Zhou, Shiguo ; Korf, Ian F. : Assemblathon 2 : evaluating de novo methods of genome assembly in three vertebrate species. 2 (2013), Num. 1, p. 10. – ISSN : 2047-217X

[Brown 2015] Brown, C. T. : Strain recovery from metagenomes. 33 (2015), Num. 10, p. 1041–1043. – URL : http://www.nature.com/nbt/journal/v33/n10/full/nbt.3375.html. – Visité le : 2016-04-04. – ISSN : 1087-0156

[Břinda *et al.* 2015] Břinda, Karel ; Sykulski, Maciej ; Kucherov, Gregory : Spaced seeds improve k -mer-based metagenomic classification. 31 (2015), Num. 22, p. 3584–3592. – URL : http://bioinformatics.oxfordjournals.org/lookup/doi/10.1093/bioinformatics/btv419. – Visité le : 2016-02-05. – ISSN : 1367-4803, 1460-2059

[Campello *et al.* 2013] Campello, Ricardo J. G. B. ; Moulavi, Davoud ; Sander, Joerg : Density-Based Clustering Based on Hierarchical Density Estimates. Dans : Pei, Jian (Éditeur) ; Tseng, Vincent S. (Éditeur) ; Cao, Longbing (Éditeur) ; Motoda, Hiroshi (Éditeur) ; Xu, Guandong (Éditeur) : *Advances in Knowledge Discovery and Data Mining*, Springer Berlin Heidelberg, 2013 (Lecture Notes in Computer Science), p. 160–172. – ISBN : 978-3-642-37456-2

[Caspi *et al.* 2018] Caspi, Ron ; Billington, Richard ; Fulcher, Carol A. ; Keseler, Ingrid M. ; Kothari, Anamika ; Krummenacker, Markus ; Latendresse, Mario ; Midford, Peter E. ; Ong, Quang ; Ong, Wai K. ; Paley, Suzanne ; Subhraveti, Pallavi ; Karp, Peter D. : The MetaCyc database of metabolic pathways and enzymes. 46 (2018), p. D633–D639. – URL : https://academic.oup.com/nar/article/46/D1/D633/4559117. – Visité le : 2018-11-28. – ISSN : 0305-1048

[Chen *et al.* 2018] Chen, DaYang ; Zhen, HeFu ; Qiu, Yong ; Liu, Ping ; Zeng, Peng ; Xia, Jun ; Shi, QianYu ; Xie, Lin ; Zhu, Zhu ; Gao, Ya ; Huang, Guo-Dong ; Wang, Jian ; Yang, HuanMing ; Chen, Fang : Comparison of single cell sequencing data between two whole genome amplification methods on two sequencing platforms. 8 (2018), Num. 1, p. 4963. – URL : https://www.nature.com/articles/s41598-018-23325-2. – Visité le : 2018-11-28. – ISSN : 2045-2322

[Cowan *et al.* 2015] Cowan, Da ; Ramond, J-B ; Makhalanyane, Tp ; De Maayer, P : Metagenomics of extreme environments. 25 (2015), p. 97–102. – URL : http://linkinghub.elsevier.com/retrieve/pii/S1369527415000569. – Visité le : 2018-08-11. – ISSN : 13695274

[Croft *et al.* 2011] Croft, David ; O'Kelly, Gavin ; Wu, Guanming ; Haw, Robin ; Gillespie, Marc ; Matthews, Lisa ; Caudy, Michael ; Garapati, Phani ; Gopinath, Gopal ; Jassal, Bijay ; Jupe, Steven ; Kalatskaya, Irina ; Mahajan, Shahana ; May, Bruce ; Ndegwa, Nelson ; Schmidt, Esther ; Shamovsky, Veronica ; Yung, Christina ; Birney, Ewan ; Hermjakob, Henning ; D'Eustachio, Peter ; Stein, Lincoln : Reactome : a database of reactions, pathways and biological processes. 39 (2011), p. D691–D697. – URL : https://academic.oup.com/nar/article/39/suppl_1/D691/2505841. – Visité le : 2018-11-28. – ISSN : 0305-1048

[Cronan 2018] Cronan, Christopher S. : Microbial Biogeochemistry. Dans : *Ecosystem Biogeochemistry*. Springer, Cham, 2018 (Springer Textbooks in Earth Sciences, Geography and Environment), p. 31–40. – URL : https://link.springer.com/chapter/10.1007/978-3-319-66444-6_3. – Visité le : 2018-07-24. – ISBN : 978-3-319-66443-9 978-3-319-66444-6

[De Filippis *et al.* 2016] De Filippis, Francesca ; Parente, Eugenio ; Ercolini, Danilo : Metagenomics insights into food fermentations. 10 (2016), Num. 1, p. 91–

102. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5270737/. – Visité le : 2018-08-11. – ISSN : 1751-7915

[Defois *et al.* 2017] Defois, Clémence ; Ratel, Jérémy ; Denis, Sylvain ; Batut, Bérénice ; Beugnot, Réjane ; Peyretaillade, Eric ; Engel, Erwan ; Peyret, Pierre : Environmental Pollutant Benzo[a]Pyrene Impacts the Volatile Metabolome and Transcriptome of the Human Gut Microbiota. 8 (2017), p. 1562. – ISSN : 1664-302X

[Dekker *et al.* 2002] Dekker, Job ; Rippe, Karsten ; Dekker, Martijn ; Kleckner, Nancy : Capturing Chromosome Conformation. 295 (2002), Num. 5558, p. 1306–1311. – URL : http://science.sciencemag.org/content/295/5558/1306. – Visité le : 2018-11-28. – ISSN : 0036-8075, 1095-9203

[Demaree *et al.* 2018] Demaree, Benjamin ; Weisgerber, Daniel ; Lan, Freeman ; Abate, Adam R. : An Ultrahigh-throughput Microfluidic Platform for Single-cell Genome Sequencing. (2018), Num. 135, p. e57598. – URL : https://www.jove.com/video/57598/an-ultrahigh-throughput-microfluidic-platform-for-single-cell-genome. – Visité le : 2018-11-28. – ISSN : 1940-087X

[Denoeux et Masson 2004] Denoeux, T. ; Masson, M. : EVCLUS : evidential clustering of proximity data. 34 (2004), Num. 1, p. 95–109. – ISSN : 1083-4419

[Denonfoux *et al.* 2013] Denonfoux, Jérémie ; Parisot, Nicolas ; Dugat-Bony, Eric ; Biderre-Petit, Corinne ; Boucher, Delphine ; Morgavi, Diego P. ; Le Paslier, Denis ; Peyretaillade, Eric ; Peyret, Pierre : Gene capture coupled to high-throughput sequencing as a strategy for targeted metagenome exploration. 20 (2013), Num. 2, p. 185–196. – ISSN : 1756-1663

[Ding *et al.* 2014] Ding, Xiao ; Cao, Chang-Chang ; Sun, Xiao : Intrinsic correlation of oligonucleotides : A novel genomic signature for metagenome analysis. 353 (2014), p. 9–18. – URL : http://linkinghub.elsevier.com/retrieve/pii/S0022519314001234. – Visité le : 2016-02-05. – ISSN : 00225193

[Ding *et al.* 2015] Ding, Xiao ; Cheng, Fudong ; Cao, Changchang ; Sun, Xiao : DectICO : an alignment-free supervised metagenomic classification method based on feature extraction and dynamic selection. 16 (2015), Num. 1. – URL : http://www.biomedcentral.com/1471-2105/16/323. – Visité le : 2016-02-05. – ISSN : 1471-2105

[Ditzler *et al.* 2015] Ditzler, Gregory ; Polikar, Robi ; Rosen, Gail : Multi-Layer and Recursive Neural Networks for Metagenomic Classification. 14 (2015), Num. 6, p. 608–616. – URL : http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7219432. – Visité le : 2016-02-05

[Edwards *et al.* 2018] Edwards, Arwyn ; Debbonaire, Aliyah R. ; Nicholls, Samuel M. ; Rassner, Sara M. ; Sattler, Birgit ; Cook, Joseph M. ; Davy, Tom ; Mur, Luis A. ; Hodson, Andrew J. : In-field metagenome and 16S rRNA gene amplicon nanopore sequencing robustly characterize glacier microbiota. (2018). – URL : https://www.biorxiv.org/content/early/2018/08/10/073965

[Ekblom *et al.* 2014] Ekblom, Robert ; Smeds, Linnéa ; Ellegren, Hans : Patterns of sequencing coverage bias revealed by ultra-deep sequencing of vertebrate mitochondria. 15 (2014), Num. 1, p. 467. – URL : https://doi.org/10.1186/1471-2164-15-467. – Visité le : 2018-11-28. – ISSN : 1471-2164

[Epskamp *et al.* 2012] Epskamp, Sacha ; Cramer, Angélique O. J. ; Waldorp, Lourens J. ; Schmittmann, Verena D. ; Borsboom, Denny : qgraph : Network Visualizations of Relationships in Psychometric Data. Dans : *Journal of Statistical Software* 48 (2012), Num. 4, p. 1–18. – URL : http://www.jstatsoft.org/v48/i04/

[Eren *et al.* 2015] Eren, A. M. ; Esen, Özcan C. ; Quince, Christopher ; Vineis, Joseph H. ; Morrison, Hilary G. ; Sogin, Mitchell L. ; Delmont, Tom O. : Anvi'o : an advanced analysis and visualization platform for 'omics data. 3 (2015), p. e1319. – URL : https://peerj.com/articles/1319. – Visité le : 2018-12-13. – ISSN : 2167-8359

[Ester *et al.* 1996] Ester, Martin ; Kriegel, Hans-Peter ; Sander, Jörg ; Xu, Xiaowei : A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Dans : *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1996 (KDD'96), p. 226–231. – URL : http://dl.acm.org/citation.cfm?id=3001460.3001507

[Evans *et al.* 2015] Evans, Paul N. ; Parks, Donovan H. ; Chadwick, Grayson L. ; Robbins, Steven J. ; Orphan, Victoria J. ; Golding, Suzanne D. ; Tyson, Gene W. : Methane metabolism in the archaeal phylum Bathyarchaeota revealed by genome-centric metagenomics. 350 (2015), Num. 6259, p. 434–438. – URL :

http://science.sciencemag.org/content/350/6259/434. – Visité le : 2016-08-31. – ISSN : 0036-8075, 1095-9203

[Fierer *et al.* 2012] Fierer, Noah ; Leff, Jonathan W. ; Adams, Byron J. ; Nielsen, Uffe N. ; Bates, Scott T. ; Lauber, Christian L. ; Owens, Sarah ; Gilbert, Jack A. ; Wall, Diana H. ; Caporaso, J. G. : Cross-biome metagenomic analyses of soil microbial communities and their functional attributes. 109 (2012), Num. 52, p. 21390–21395. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3535587/. – Visité le : 2017-10-24. – ISSN : 0027-8424

[Fleischmann *et al.* 1995] Fleischmann, R. D. ; Adams, M. D. ; White, O. ; Clayton, R. A. ; Kirkness, E. F. ; Kerlavage, A. R. ; Bult, C. J. ; Tomb, J. F. ; Dougherty, B. A. ; Merrick, J. M. : Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. 269 (1995), Num. 5223, p. 496–512. – ISSN : 0036-8075

[Forouzan *et al.* 2018] Forouzan, Esmaeil ; Shariati, Parvin ; Mousavi Maleki, Masoumeh S. ; Karkhane, Ali A. ; Yakhchali, Bagher : Practical evaluation of 11 de novo assemblers in metagenome assembly. 151 (2018), p. 99–105. – URL : http://www.sciencedirect.com/science/article/pii/S0167701218301210. – Visité le : 2018-12-04. – ISSN : 0167-7012

[Gasc et Peyret 2017] Gasc, Cyrielle ; Peyret, Pierre : Revealing large metagenomic regions through long DNA fragment hybridization capture. 5 (2017), Num. 1, p. 33. – ISSN : 2049-2618

[Gasc et Peyret 2018] Gasc, Cyrielle ; Peyret, Pierre : Hybridization capture reveals microbial diversity missed using current profiling methods. 6 (2018), Num. 1, p. 61. – ISSN : 2049-2618

[Gasc *et al.* 2015] Gasc, Cyrielle ; Ribière, Céline ; Parisot, Nicolas ; Beugnot, Réjane ; Defois, Clémence ; Petit-Biderre, Corinne ; Boucher, Delphine ; Peyretaillade, Eric ; Peyret, Pierre : Capturing prokaryotic dark matter genomes. 166 (2015), Num. 10, p. 814–830. – URL : http://linkinghub.elsevier.com/retrieve/pii/S0923250815000984. – Visité le : 2016-02-05. – ISSN : 09232508

[Gawad *et al.* 2016] Gawad, Charles ; Koh, Winston ; Quake, Stephen R. : Single-cell genome sequencing : current state of the science. 17 (2016), Num. 3, p. 175–188. – URL : https://www.nature.com/articles/nrg.2015.16. – Visité le : 2018-08-03. – ISSN : 1471-0064

[Ghosh et Acharya 2011] Ghosh, Joydeep ; Acharya, Ayan : Cluster ensembles. 1 (2011), Num. 4, p. 305–315. – URL : http://doi.wiley.com/10.1002/widm.32. – Visité le : 2017-10-24. – ISSN : 19424787

[Ghurye *et al.* 2016] Ghurye, Jay S. ; Cepeda-Espinoza, Victoria ; Pop, Mihai : Metagenomic Assembly : Overview, Challenges and Applications. 89 (2016), Num. 3, p. 353–362. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5045144/. – Visité le : 2018-12-04. – ISSN : 0044-0086

[Girotto *et al.* 2016] Girotto, Samuele ; Pizzi, Cinzia ; Comin, Matteo : MetaProb : accurate metagenomic reads binning based on probabilistic sequence signatures. 32 (2016), Num. 17, p. i567–i575. – URL : http://bioinformatics.oxfordjournals.org/content/32/17/i567. – Visité le : 2016-09-05. – ISSN : 1367-4803, 1460-2059

[Graham *et al.* 2017] Graham, Elaina D. ; Heidelberg, John F. ; Tully, Benjamin J. : BinSanity : unsupervised clustering of environmental microbial assemblies using coverage and affinity propagation. 5 (2017), p. e3035. – URL : https://peerj.com/articles/3035. – Visité le : 2018-08-23. – ISSN : 2167-8359

[Grüning *et al.* 2018] Grüning, Björn ; Dale, Ryan ; Sjödin, Andreas ; Chapman, Brad A. ; Rowe, Jillian ; Tomkins-Tinch, Christopher H. ; Valieris, Renan ; Köster, Johannes : Bioconda : sustainable and comprehensive software distribution for the life sciences. 15 (2018), Num. 7, p. 475–476. – URL : https://www.nature.com/articles/s41592-018-0046-7. – Visité le : 2018-08-17. – ISSN : 1548-7105

[Guerra *et al.* 2018] Guerra, Alaine B. ; Oliveira, Jorge S. ; Silva-Portela, Rita C. B. ; Araújo, Wydemberg ; Carlos, Aline C. ; Vasconcelos, Ana Tereza R. ; Freitas, Ana T. ; Domingos, Yldeney S. ; Farias, Mirna F. de ; Fernandes, Glauber José T. ; Agnez-Lima, Lucymara F. : Metagenome enrichment approach used for selection of oil-degrading bacteria consortia for drill cutting residue bioremediation. 235 (2018), p. 869–880. – URL : http://www.sciencedirect.com/science/article/pii/S0269749117330531. – Visité le : 2018-11-28. – ISSN : 0269-7491

[Haque *et al.* 2013] Haque, Farzin ; Li, Jinghong ; Wu, Hai-Chen ; Liang, Xing-Jie ; Guo, Peixuan : Solid-State and Biological Nanopore for Real-Time Sensing of Single Chemical and Sequencing of DNA. 8 (2013), Num. 1, p. 56–74. –

URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3596169/. – Visité le : 2018-08-11. – ISSN : 1748-0132

[HDF Group 1997] HDF Group, The : *Hierarchical Data Format, version 5*. 1997. – URL : https://www.hdfgroup.org/

[Heisterkamp 2015] Heisterkamp, Douglas R. : Lambda Consensus Clustering. Dans : *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)* IEEE (Event), 2015, p. 410–413

[Huang et Liao 2016] Huang, Yao-Ting ; Liao, Chen-Fu : Integration of String and de Bruijn Graphs for Genome Assembly. (2016), p. btw011. – URL : http://bioinformatics.oxfordjournals.org/content/early/2016/01/09/bioinformatics.btw011.short. – Visité le : 2016-02-05

[Hugerth *et al.* 2015] Hugerth, Luisa W. ; Larsson, John ; Alneberg, Johannes ; Lindh, Markus V. ; Legrand, Catherine ; Pinhassi, Jarone ; Andersson, Anders F. : Metagenome-assembled genomes uncover a global brackish microbiome. 16 (2015), Num. 1. – URL : http://genomebiology.com/2015/16/1/279. – Visité le : 2016-02-05. – ISSN : 1474-760X

[Hughes *et al.* 2014] Hughes, Jim R. ; Roberts, Nigel ; McGowan, Simon ; Hay, Deborah ; Giannoulatou, Eleni ; Lynch, Magnus ; De Gobbi, Marco ; Taylor, Stephen ; Gibbons, Richard ; Higgs, Douglas R. : Analysis of hundreds of *cis*-regulatory landscapes at high resolution in a single, high-throughput experiment. 46 (2014), Num. 2, p. 205–212. – URL : https://www.nature.com/articles/ng.2871. – Visité le : 2018-11-28. – ISSN : 1546-1718

[Huntemann *et al.* 2015] Huntemann, Marcel ; Ivanova, Natalia N. ; Mavromatis, Konstantinos ; Tripp, H. J. ; Paez-Espino, David ; Palaniappan, Krishnaveni ; Szeto, Ernest ; Pillay, Manoj ; Chen, I-Min A. ; Pati, Amrita ; Nielsen, Torben ; Markowitz, Victor M. ; Kyrpides, Nikos C. : The standard operating procedure of the DOE-JGI Microbial Genome Annotation Pipeline (MGAP v.4). 10 (2015), Num. 1. – URL : http://standardsingenomics.biomedcentral.com/articles/10.1186/s40793-015-0077-y. – Visité le : 2018-08-11. – ISSN : 1944-3277

[Hyatt *et al.* 2010] Hyatt, Doug ; Chen, Gwo-Liang ; LoCascio, Philip F. ; Land, Miriam L. ; Larimer, Frank W. ; Hauser, Loren J. : Prodigal : prokaryotic gene

recognition and translation initiation site identification. 11 (2010), Num. 1, p. 119. – URL : https://doi.org/10.1186/1471-2105-11-119. – Visité le : 2018-08-21. – ISSN : 1471-2105

[Imelfort *et al.* 2014] Imelfort, Michael ; Parks, Donovan ; Woodcroft, Ben J. ; Dennis, Paul ; Hugenholtz, Philip ; Tyson, Gene W. : GroopM : an automated tool for the recovery of population genomes from related metagenomes. 2 (2014), p. e603. – URL : https://peerj.com/articles/603. – Visité le : 2018-08-22. – ISSN : 2167-8359

[International Human Genome Sequencing Consortium 2001] International Human Genome Sequencing Consortium : Initial sequencing and analysis of the human genome. 409 (2001), Num. 6822, p. 860–921. – URL : https://www.nature.com/articles/35057062. – Visité le : 2018-12-03. – ISSN : 1476-4687

[Izsák et Pavoine 2012] Izsák, János ; Pavoine, Sandrine : Links between the species abundance distribution and the shape of the corresponding rank abundance curve. 14 (2012), Num. 1, p. 1–6. – URL : http://www.sciencedirect.com/science/article/pii/S1470160X11002019. – Visité le : 2018-12-05. – ISSN : 1470-160X

[Jain 2010] Jain, Anil K. : Data clustering : 50 years beyond K-means. 31 (2010), Num. 8, p. 651–666. – URL : http://linkinghub.elsevier.com/retrieve/pii/S0167865509002323. – Visité le : 2018-08-15. – ISSN : 01678655

[Jain *et al.* 2018] Jain, Chirag ; Rodriguez-R, Luis M. ; Phillippy, Adam M. ; Konstantinidis, Konstantinos T. ; Aluru, Srinivas : High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries. 9 (2018), Num. 1, p. 5114. – URL : https://www.nature.com/articles/s41467-018-07641-9. – Visité le : 2018-12-03. – ISSN : 2041-1723

[Jandhyala 2015] Jandhyala, Sai M. : Role of the normal gut microbiota. 21 (2015), Num. 29, p. 8787. – URL : http://www.wjgnet.com/1007-9327/full/v21/i29/8787.htm. – Visité le : 2018-07-24. – ISSN : 1007-9327

[Jansen 2017] Jansen, Stefan : Word and Phrase Translation with word2vec. (2017). – URL : http://arxiv.org/abs/1705.03127. – Visité le : 2018-09-07

[Jolliffe et Cadima 2016] Jolliffe, Ian T. ; Cadima, Jorge : Principal component analysis : a review and recent developments. 374 (2016), Num. 2065, p. 20150202. – URL : http://rsta.royalsocietypublishing.org/content/374/2065/20150202. – Visité le : 2018-11-28. – ISSN : 1364-503X, 1471-2962

[Kanehisa *et al.* 2017] Kanehisa, Minoru ; Furumichi, Miho ; Tanabe, Mao ; Sato, Yoko ; Morishima, Kanae : KEGG : new perspectives on genomes, pathways, diseases and drugs. 45 (2017), p. D353–D361. – URL : https://academic.oup.com/nar/article/45/D1/D353/2605697. – Visité le : 2018-11-28. – ISSN : 0305-1048

[Kang *et al.* 2015] Kang, Dongwan D. ; Froula, Jeff ; Egan, Rob ; Wang, Zhong : MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. 3 (2015), p. e1165. – URL : https://peerj.com/articles/1165. – Visité le : 2016-02-05. – ISSN : 2167-8359

[Keich *et al.* 2004] Keich, Uri ; Li, Ming ; Ma, Bin ; Tromp, John : On spaced seeds for similarity search. 138 (2004), Num. 3, p. 253–263. – URL : http://www.sciencedirect.com/science/article/pii/S0166218X03003822. – Visité le : 2018-09-07. – ISSN : 0166-218X

[Kwong et Moran 2016] Kwong, Waldan K. ; Moran, Nancy A. : Gut Microbial Communities of Social Bees. 14 (2016), Num. 6, p. 374–384. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5648345/. – Visité le : 2018-07-24. – ISSN : 1740-1526

[Laczny *et al.* 2015] Laczny, Cedric C. ; Sternal, Tomasz ; Plugaru, Valentin ; Gawron, Piotr ; Atashpendar, Arash ; Margossian, Houry H. ; Coronado, Sergio ; Maaten, Laurens v. der ; Vlassis, Nikos ; Wilmes, Paul : VizBin - an application for reference-independent visualization and human-augmented binning of metagenomic data. 3 (2015), p. 1. – URL : http://dx.doi.org/10.1186/s40168-014-0066-1. – Visité le : 2016-02-11. – ISSN : 2049-2618

[Langille *et al.* 2013] Langille, Morgan G. ; Zaneveld, Jesse ; Caporaso, J G. ; McDonald, Daniel ; Knights, Dan ; Reyes, Joshua A. ; Clemente, Jose C. ; Burkepile, Deron E. ; Thurber, Rebecca L V. ; Knight, Rob *et al.* : Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. Dans : *Nature biotechnology* 31 (2013), Num. 9, p. 814. – URL : https://www.nature.com/articles/nbt.2676

[Langmead et Salzberg 2012] Langmead, Ben ; Salzberg, Steven L. : Fast gapped-read alignment with Bowtie 2. 9 (2012), Num. 4, p. 357–359. – ISSN : 1548-7105

[Leggett *et al.* 2018] Leggett, Richard M. ; Alcon-Giner, Cristina ; Heavens, Darren ; Caim, Shabhonam ; Brook, Thomas C. ; Kujawska, Magdalena ; Martin, Samuel ; Hoyles, Lesley ; Clarke, Paul ; Hall, Lindsay ; Clark, Matthew D. : Rapid profiling of the preterm infant gut microbiota using nanopore sequencing aids pathogen diagnostics. (2018), p. 180406. – URL : https://www.biorxiv.org/content/early/2018/10/12/180406. – Visité le : 2018-11-28

[Li *et al.* 2016] Li, Dinghua ; Luo, Ruibang ; Liu, Chi-Man ; Leung, Chi-Ming ; Ting, Hing-Fung ; Sadakane, Kunihiko ; Yamashita, Hiroshi ; Lam, Tak-Wah : MEGAHIT v1. 0 : A fast and scalable metagenome assembler driven by advanced methodologies and community practices. 102 (2016), p. 3–11. – URL : http://www.sciencedirect.com/science/article/pii/S1046202315301183. – Visité le : 2016-03-29. – ISSN : 1046-2023

[Li *et al.* 2012] Li, Zhenyu ; Chen, Yanxiang ; Mu, Desheng ; Yuan, Jianying ; Shi, Yujian ; Zhang, Hao ; Gan, Jun ; Li, Nan ; Hu, Xuesong ; Liu, Binghang ; Yang, Bicheng ; Fan, Wei : Comparison of the two major classes of assembly algorithms : overlap-layout-consensus and de-bruijn-graph. 11 (2012), Num. 1, p. 25–37. – ISSN : 2041-2657

[Liang *et al.* 2018] Liang, Dachao ; Leung, Ross Ka-Kit ; Guan, Wenda ; Au, William W. : Involvement of gut microbiome in human health and disease : brief overview, knowledge gaps and research opportunities. 10 (2018). – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5785832/. – Visité le : 2018-07-24. – ISSN : 1757-4749

[Liao *et al.* 2014] Liao, Ruiqi ; Zhang, Ruichang ; Guan, Jihong ; Zhou, Shuigeng : A New Unsupervised Binning Approach for Metagenomic Sequences Based on N-grams and Automatic Feature Weighting. 11 (2014), Num. 1, p. 42–54. – ISSN : 1557-9964

[Lin et Liao 2016] Lin, Hsin-Hung ; Liao, Yu-Chieh : Accurate binning of metagenomic contigs via automated clustering sequences using information of genomic signatures and marker genes. 6 (2016), p. 24175. – URL : http:

//www.nature.com/articles/srep24175. – Visité le : 2016-07-06. – ISSN : 2045-2322

[Lischer et Shimizu 2017] Lischer, Heidi E. L. ; Shimizu, Kentaro K. : Reference-guided de novo assembly approach improves genome reconstruction for related species. 18 (2017), Num. 1, p. 474. – URL : https://doi.org/10.1186/s12859-017-1911-6. – Visité le : 2018-11-28. – ISSN : 1471-2105

[Liu *et al.* 2010] Liu, Yanchi ; Li, Zhongmou ; Xiong, Hui ; Gao, Xuedong ; Wu, Junjie : Understanding of Internal Clustering Validation Measures. Dans : *2010 IEEE International Conference on Data Mining*, IEEE, dec 2010. – URL : https://doi.org/10.1109/icdm.2010.35

[Lloyd *et al.* 2018] Lloyd, Karen G. ; Steen, Andrew D. ; Ladau, Joshua ; Yin, Junqi ; Crosby, Lonnie : Phylogenetically Novel Uncultured Microbial Cells Dominate Earth Microbiomes. 3 (2018), Num. 5, p. e00055–18. – URL : https://msystems.asm.org/content/3/5/e00055-18. – Visité le : 2018-12-03. – ISSN : 2379-5077

[Lloyd 1982] Lloyd, Stuart : Least squares quantization in PCM. Dans : *IEEE transactions on information theory* 28 (1982), Num. 2, p. 129–137

[Lu *et al.* 2017] Lu, Yang Y. ; Chen, Ting ; Fuhrman, Jed A. ; Sun, Fengzhu ; Sahinalp, Cenk : COCACOLA : binning metagenomic contigs using sequence COmposition, read CoverAge, CO-alignment and paired-end read LinkAge. 33 (2017), Num. 6, p. 791–798. – URL : https://academic.oup.com/bioinformatics/article/33/6/791/2525584. – Visité le : 2018-08-21. – ISSN : 1367-4803

[Luo *et al.* 2012] Luo, Ruibang ; Liu, Binghang ; Xie, Yinlong ; Li, Zhenyu ; Huang, Weihua ; Yuan, Jianying ; He, Guangzhu ; Chen, Yanxiang ; Pan, Qi ; Liu, Yunjie ; Tang, Jingbo ; Wu, Gengxiong ; Zhang, Hao ; Shi, Yujian ; Liu, Yong ; Yu, Chang ; Wang, Bo ; Lu, Yao ; Han, Changlei ; Cheung, David W. ; Yiu, Siu-Ming ; Peng, Shaoliang ; Xiaoqian, Zhu ; Liu, Guangming ; Liao, Xiangke ; Li, Yingrui ; Yang, Huanming ; Wang, Jian ; Lam, Tak-Wah ; Wang, Jun : SOAPdenovo2 : an empirically improved memory-efficient short-read de novo assembler. 1 (2012), p. 18. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3626529/. – Visité le : 2018-12-05. – ISSN : 2047-217X

[Lynch et Neufeld 2015] Lynch, Michael D. J. ; Neufeld, Josh D. : Ecology and exploration of the rare biosphere. 13 (2015), Num. 4, p. 217–229. – URL : https:

//www.nature.com/articles/nrmicro3400. – Visité le : 2018-07-31. – ISSN : 1740-1534

[Magoc *et al.* 2013] Magoc, Tanja ; Pabinger, Stephan ; Canzar, Stefan ; Liu, Xinyue ; Su, Qi ; Puiu, Daniela ; Tallon, Luke J. ; Salzberg, Steven L. : GAGE-B : an evaluation of genome assemblers for bacterial organisms. 29 (2013), Num. 14, p. 1718–1725. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3702249/. – Visité le : 2018-09-03. – ISSN : 1367-4803

[Malmuthuge et Guan 2017] Malmuthuge, Nilusha ; Guan, Le L. : Understanding host-microbial interactions in rumen : searching the best opportunity for microbiota manipulation. 8 (2017). – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5244612/. – Visité le : 2018-07-24. – ISSN : 1674-9782

[Mande *et al.* 2012] Mande, S. S. ; Mohammed, M. H. ; Ghosh, T. S. : Classification of metagenomic sequences : methods and challenges. 13 (2012), Num. 6, p. 669–681. – URL : http://bib.oxfordjournals.org/cgi/doi/10.1093/bib/bbs054. – Visité le : 2016-02-05. – ISSN : 1467-5463, 1477-4054

[Marbouty *et al.* 2014] Marbouty, Martial ; Cournac, Axel ; Flot, Jean-François ; Marie-Nelly, Hervé ; Mozziconacci, Julien ; Koszul, Romain : Metagenomic chromosome conformation capture (meta3C) unveils the diversity of chromosome organization in microorganisms. Dans : *eLife* 3 (2014), dec. – URL : https://doi.org/10.7554/elife.03318

[Marchesi et Ravel 2015] Marchesi, Julian R. ; Ravel, Jacques : The vocabulary of microbiome research : a proposal. 3 (2015), p. 31. – URL : https://doi.org/10.1186/s40168-015-0094-5. – Visité le : 2018-04-24. – ISSN : 2049-2618

[Markowitz *et al.* 2012] Markowitz, Victor M. ; Chen, I.-Min A. ; Palaniappan, Krishna ; Chu, Ken ; Szeto, Ernest ; Grechkin, Yuri ; Ratner, Anna ; Jacob, Biju ; Huang, Jinghua ; Williams, Peter ; Huntemann, Marcel ; Anderson, Iain ; Mavromatis, Konstantinos ; Ivanova, Natalia N. ; Kyrpides, Nikos C. : IMG : the Integrated Microbial Genomes database and comparative analysis system. 40 (2012), p. D115–122. – ISSN : 1362-4962

[Masson et Denœux 2008] Masson, Marie-Hélène ; Denœux, T. : ECM : An evidential version of the fuzzy c-means algorithm. 41 (2008), Num. 4, p. 1384–1397. – URL : http://linkinghub.elsevier.com/retrieve/pii/S0031320307004062. – Visité le : 2018-08-13. – ISSN : 00313203

[Meyer *et al.* 2018] Meyer, Fernando ; Hofmann, Peter ; Belmann, Peter ; Garrido-Oter, Ruben ; Fritz, Adrian ; Sczyrba, Alexander ; McHardy, Alice C. : AMBER : Assessment of Metagenome BinnERs. 7 (2018), Num. 6. – URL : https://academic.oup.com/gigascience/article/7/6/giy069/5034950. – Visité le : 2018-08-17

[Mikolov *et al.* 2013] Mikolov, Tomas ; Chen, Kai ; Corrado, Greg ; Dean, Jeffrey : Efficient Estimation of Word Representations in Vector Space. (2013). – URL : http://arxiv.org/abs/1301.3781. – Visité le : 2017-10-24

[Militon *et al.* 2007] Militon, Cécile ; Rimour, Sébastien ; Missaoui, Mohieddine ; Biderre, Corinne ; Barra, Vincent ; Hill, David ; Moné, Anne ; Gagne, Geneviève ; Meier, Harald ; Peyretaillade, Eric ; Peyret, Pierre : PhylArray : phylogenetic probe design algorithm for microarray. 23 (2007), Num. 19, p. 2550–2557. – URL : https://academic.oup.com/bioinformatics/article/23/19/2550/187167. – Visité le : 2018-11-28. – ISSN : 1367-4803

[Miller *et al.* 2008] Miller, Jason R. ; Delcher, Arthur L. ; Koren, Sergey ; Venter, Eli ; Walenz, Brian P. ; Brownley, Anushka ; Johnson, Justin ; Li, Kelvin ; Mobarry, Clark ; Sutton, Granger : Aggressive assembly of pyrosequencing reads with mates. 24 (2008), Num. 24, p. 2818–2824. – URL : https://academic.oup.com/bioinformatics/article/24/24/2818/197033. – Visité le : 2018-11-28. – ISSN : 1367-4803

[Musso *et al.* 2010] Musso, Giovanni ; Gambino, Roberto ; Cassader, Maurizio : Obesity, Diabetes, and Gut Microbiota : The hygiene hypothesis expanded ? 33 (2010), Num. 10, p. 2277–2284. – URL : http://care.diabetesjournals.org/content/33/10/2277. – Visité le : 2018-11-28. – ISSN : 0149-5992, 1935-5548

[Myers 2005] Myers, Eugene W. : The fragment assembly string graph. 21 (2005), p. ii79–ii85. – URL : https://academic.oup.com/bioinformatics/article/21/suppl_2/ii79/227189. – Visité le : 2018-07-30. – ISSN : 1367-4803

[Namiki *et al.* 2012] Namiki, Toshiaki ; Hachiya, Tsuyoshi ; Tanaka, Hideaki ; Sakakibara, Yasubumi : MetaVelvet : an extension of Velvet assembler to de novo metagenome assembly from short sequence reads. 40 (2012), Num. 20, p. e155. – ISSN : 1362-4962

[Nauman *et al.* 2019] Nauman, Mohammad ; Rehman, Hafeez U. ; Politano, Gianfranco ; Benso, Alfredo : Beyond Homology Transfer : Deep Learning for

Automated Annotation of Proteins. Dans : *Journal of Grid Computing* 17 (2019), Num. 2, p. 225–237

[Nelson *et al.* 2010] Nelson, K. E. ; Weinstock, G. M. ; Highlander, S. K. ; Worley, K. C. ; Creasy, H. H. ; Wortman, J. R. ; Rusch, D. B. ; Mitreva, M. ; Sodergren, E. ; Chinwalla, A. T. ; Feldgarden, M. ; Gevers, D. ; Haas, B. J. ; Madupu, R. ; Ward, D. V. ; Birren, B. W. ; Gibbs, R. A. ; Methe, B. ; Petrosino, J. F. ; Strausberg, R. L. ; Sutton, G. G. ; White, O. R. ; Wilson, R. K. ; Durkin, S. ; Giglio, M. G. ; Gujja, S. ; Howarth, C. ; Kodira, C. D. ; Kyrpides, N. ; Mehta, T. ; Muzny, D. M. ; Pearson, M. ; Pepin, K. ; Pati, A. ; Qin, X. ; Yandava, C. ; Zeng, Q. ; Zhang, L. ; Berlin, A. M. ; Chen, L. ; Hepburn, T. A. ; Johnson, J. ; McCorrison, J. ; Miller, J. ; Minx, P. ; Nusbaum, C. ; Russ, C. ; Sykes, S. M. ; Tomlinson, C. M. ; Young, S. ; Warren, W. C. ; Badger, J. ; Crabtree, J. ; Markowitz, V. M. ; Orvis, J. ; Cree, A. ; Ferriera, S. ; Fulton, L. L. ; Fulton, R. S. ; Gillis, M. ; Hemphill, L. D. ; Joshi, V. ; Kovar, C. ; Torralba, M. ; Wetterstrand, K. A. ; Abouellleil, A. ; Wollam, A. M. ; Buhay, C. J. ; Ding, Y. ; Dugan, S. ; FitzGerald, M. G. ; Holder, M. ; Hostetler, J. ; Clifton, S. W. ; Allen-Vercoe, E. ; Earl, A. M. ; Farmer, C. N. ; Liolios, K. ; Surette, M. G. ; Xu, Q. ; Pohl, C. ; Wilczek-Boney, K. ; Zhu, D. : A Catalog of Reference Genomes from the Human Microbiome. Dans : *Science* 328 (2010), may, Num. 5981, p. 994–999. – URL : https://doi.org/10.1126/science.1183605

[Ng 2017] Ng, Patrick : dna2vec : Consistent vector representations of variable-length k-mers. (2017). – URL : http://arxiv.org/abs/1701.06279. – Visité le : 2017-10-24

[Nielsen *et al.* 2014] Nielsen, H B. ; Almeida, Mathieu ; Juncker, Agnieszka S. ; Rasmussen, Simon ; Li, Junhua ; Sunagawa, Shinichi ; Plichta, Damian R. ; Gautier, Laurent ; Pedersen, Anders G. ; Le Chatelier, Emmanuelle ; Pelletier, Eric ; Bonde, Ida ; Nielsen, Trine ; Manichanh, Chaysavanh ; Arumugam, Manimozhiyan ; Batto, Jean-Michel ; Santos, Marcelo B. Quintanilha dos ; Blom, Nikolaj ; Borruel, Natalia ; Burgdorf, Kristoffer S. ; Boumezbeur, Fouad ; Casellas, Francesc ; Doré, Joël ; Dworzynski, Piotr ; Guarner, Francisco ; Hansen, Torben ; Hildebrand, Falk ; Kaas, Rolf S. ; Kennedy, Sean ; Kristiansen, Karsten ; Kultima, Jens R. ; Léonard, Pierre ; Levenez, Florence ; Lund, Ole ; Moumen, Bouziane ; Le Paslier, Denis ; Pons, Nicolas ;

Pedersen, Oluf ; Prifti, Edi ; Qin, Junjie ; Raes, Jeroen ; Sørensen, Søren ; Tap, Julien ; Tims, Sebastian ; Ussery, David W. ; Yamada, Takuji ; Nielsen, H B. ; Almeida, Mathieu ; Juncker, Agnieszka S. ; Rasmussen, Simon ; Li, Junhua ; Sunagawa, Shinichi ; Plichta, Damian R. ; Gautier, Laurent ; Pedersen, Anders G. ; Le Chatelier, Emmanuelle ; Pelletier, Eric ; Bonde, Ida ; Nielsen, Trine ; Manichanh, Chaysavanh ; Arumugam, Manimozhiyan ; Batto, Jean-Michel ; Santos, Marcelo B. Quintanilha dos ; Blom, Nikolaj ; Borruel, Natalia ; Burgdorf, Kristoffer S. ; Boumezbeur, Fouad ; Casellas, Francesc ; Doré, Joël ; Dworzynski, Piotr ; Guarner, Francisco ; Hansen, Torben ; Hildebrand, Falk ; Kaas, Rolf S. ; Kennedy, Sean ; Kristiansen, Karsten ; Kultima, Jens R. ; Leonard, Pierre ; Levenez, Florence ; Lund, Ole ; Moumen, Bouziane ; Le Paslier, Denis ; Pons, Nicolas ; Pedersen, Oluf ; Prifti, Edi ; Qin, Junjie ; Raes, Jeroen ; Sørensen, Søren ; Tap, Julien ; Tims, Sebastian ; Ussery, David W. ; Yamada, Takuji ; Renault, Pierre ; Sicheritz-Ponten, Thomas ; Bork, Peer ; Wang, Jun ; Brunak, Søren ; Ehrlich, S D. ; Jamet, Alexandre ; Mérieux, Alexandre ; Cultrone, Antonella ; Torrejon, Antonio ; Quinquis, Benoit ; Brechot, Christian ; Delorme, Christine ; M'Rini, Christine ; Vos, Willem M. de ; Maguin, Emmanuelle ; Varela, Encarna ; Guedon, Eric ; Gwen, Falony ; Haimet, Florence ; Artiguenave, François ; Vandemeulebrouck, Gaetana ; Denariaz, Gérard ; Khaci, Ghalia ; Blottière, Hervé ; Knol, Jan ; Weissenbach, Jean ; Hylckama Vlieg, Johan E T. van ; Torben, Jørgensen ; Parkhill, Julian ; Turner, Keith ; Guchte, Maarten van de ; Antolin, Maria ; Rescigno, Maria ; Kleerebezem, Michiel ; Derrien, Muriel ; Galleron, Nathalie ; Sanchez, Nicolas ; Grarup, Niels ; Veiga, Patrick ; Oozeer, Raish ; Dervyn, Rozenn ; Layec, Séverine ; Bruls, Thomas ; Winogradski, Yohanan ; Erwin G, Zoetendal ; Renault, Pierre ; Sicheritz-Ponten, Thomas ; Bork, Peer ; Wang, Jun ; Brunak, Søren ; Ehrlich, S D. : Identification and assembly of genomes and genetic elements in complex metagenomic samples without using reference genomes. 32 (2014), Num. 8, p. 822–828. – URL : http://www.nature.com/doifinder/10.1038/nbt.2939. – Visité le : 2016-02-05. – ISSN : 1087-0156, 1546-1696

[Noguchi *et al.* 2008] Noguchi, Hideki ; Taniguchi, Takeaki ; Itoh, Takehiko : MetaGeneAnnotator : detecting species-specific patterns of ribosomal binding site for precise gene prediction in anonymous prokaryotic and phage genomes. 15 (2008), Num. 6, p. 387–396. – ISSN : 1756-1663

[Noyes *et al.* 2017] Noyes, Noelle R. ; Weinroth, Maggie E. ; Parker, Jennifer K. ; Dean, Chris J. ; Lakin, Steven M. ; Raymond, Robert A. ; Rovira, Pablo ; Doster, Enrique ; Abdo, Zaid ; Martin, Jennifer N. ; Jones, Kenneth L. ; Ruiz, Jaime ; Boucher, Christina A. ; Belk, Keith E. ; Morley, Paul S. : Enrichment allows identification of diverse, rare elements in metagenomic resistome-virulome sequencing. 5 (2017), Num. 1, p. 142. – URL : https://doi.org/10.1186/s40168-017-0361-8. – Visité le : 2018-11-28. – ISSN : 2049-2618

[Oksanen *et al.* 2017] Oksanen, Jari ; Blanchet, F. G. ; Friendly, Michael ; Kindt, Roeland ; Legendre, Pierre ; McGlinn, Dan ; Minchin, Peter R. ; O'Hara, R. B. ; Simpson, Gavin L. ; Solymos, Peter ; Stevens, M. Henry H. ; Szoecs, Eduard ; Wagner, Helene : *vegan : Community Ecology Package*. 2017. – URL : http://cran.us.r-project.org/web/packages/vegan/index.html

[O'Leary *et al.* 2016] O'Leary, Nuala A. ; Wright, Mathew W. ; Brister, J. R. ; Ciufo, Stacy ; Haddad, Diana ; McVeigh, Rich ; Rajput, Bhanu ; Robbertse, Barbara ; Smith-White, Brian ; Ako-Adjei, Danso ; Astashyn, Alexander ; Badretdin, Azat ; Bao, Yiming ; Blinkova, Olga ; Brover, Vyacheslav ; Chetvernin, Vyacheslav ; Choi, Jinna ; Cox, Eric ; Ermolaeva, Olga ; Farrell, Catherine M. ; Goldfarb, Tamara ; Gupta, Tripti ; Haft, Daniel ; Hatcher, Eneida ; Hlavina, Wratko ; Joardar, Vinita S. ; Kodali, Vamsi K. ; Li, Wenjun ; Maglott, Donna ; Masterson, Patrick ; McGarvey, Kelly M. ; Murphy, Michael R. ; O'Neill, Kathleen ; Pujar, Shashikant ; Rangwala, Sanjida H. ; Rausch, Daniel ; Riddick, Lillian D. ; Schoch, Conrad ; Shkeda, Andrei ; Storz, Susan S. ; Sun, Hanzhen ; Thibaud-Nissen, Francoise ; Tolstoy, Igor ; Tully, Raymond E. ; Vatsan, Anjana R. ; Wallin, Craig ; Webb, David ; Wu, Wendy ; Landrum, Melissa J. ; Kimchi, Avi ; Tatusova, Tatiana ; DiCuccio, Michael ; Kitts, Paul ; Murphy, Terence D. ; Pruitt, Kim D. : Reference sequence (RefSeq) database at NCBI : current status, taxonomic expansion, and functional annotation. 44 (2016), p. D733–D745. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4702849/. – Visité le : 2017-10-24. – ISSN : 0305-1048

[Parisot 2014] Parisot, Nicolas : *Détermination de sondes oligonucléotidiques pour l'exploration à haut débit de la diversité taxonomique et fonctionnelle d'envi-*

ronnements complexes. 2014. – URL : http://www.theses.fr/2014CLF22498. – Visité le : 2018-11-28

[Parisot *et al.* 2012] Parisot, Nicolas ; Denonfoux, Jérémie ; Dugat-Bony, Eric ; Peyret, Pierre ; Peyretaillade, Eric : KASpOD—a web service for highly specific and explorative oligonucleotide design. 28 (2012), Num. 23, p. 3161–3162. – URL : https://academic.oup.com/bioinformatics/article/28/23/3161/195663. – Visité le : 2018-11-28. – ISSN : 1367-4803

[Parks 2015] Parks, Donovan : *DBB : v1.0.5*. 2015. – URL : https://zenodo.org/record/17426. – Visité le : 2018-11-28

[Parks *et al.* 2015] Parks, Donovan H. ; Imelfort, Michael ; Skennerton, Connor T. ; Hugenholtz, Philip ; Tyson, Gene W. : CheckM : assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. 25 (2015), Num. 7, p. 1043–1055. – URL : http://genome.cshlp.org/content/25/7/1043. – Visité le : 2016-02-11. – ISSN : 1088-9051, 1549-5469

[Parks *et al.* 2017] Parks, Donovan H. ; Rinke, Christian ; Chuvochina, Maria ; Chaumeil, Pierre-Alain ; Woodcroft, Ben J. ; Evans, Paul N. ; Hugenholtz, Philip ; Tyson, Gene W. : Recovery of nearly 8,000 metagenome-assembled genomes substantially expands the tree of life. 2 (2017), Num. 11, p. 1533–1542. – URL : http://www.nature.com/articles/s41564-017-0012-7. – Visité le : 2017-11-10. – ISSN : 2058-5276

[Pasolli *et al.* 2019] Pasolli, Edoardo ; Asnicar, Francesco ; Manara, Serena ; Zolfo, Moreno ; Karcher, Nicolai ; Armanini, Federica ; Beghini, Francesco ; Manghi, Paolo ; Tett, Adrian ; Ghensi, Paolo ; Collado, Maria C. ; Rice, Benjamin L. ; DuLong, Casey ; Morgan, Xochitl C. ; Golden, Christopher D. ; Quince, Christopher ; Huttenhower, Curtis ; Segata, Nicola : Extensive Unexplored Human Microbiome Diversity Revealed by Over 150, 000 Genomes from Metagenomes Spanning Age, Geography, and Lifestyle. Dans : *Cell* 176 (2019), jan, Num. 3, p. 649–662.e20. – URL : https://doi.org/10.1016/j.cell.2019.01.001

[Pedregosa *et al.* 2011] Pedregosa, Fabian ; Varoquaux, Gaël ; Gramfort, Alexandre ; Michel, Vincent ; Thirion, Bertrand ; Grisel, Olivier ; Blondel, Mathieu ; Prettenhofer, Peter ; Weiss, Ron ; Dubourg, Vincent ; Vanderplas, Jake ; Passos, Alexandre ; Cournapeau, David ; Brucher, Matthieu ;

Perrot, Matthieu ; Duchesnay, Édouard : Scikit-learn : Machine Learning in Python. 12 (2011), p. 2825–2830. – URL : http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html. – Visité le : 2017-11-10

[Pedrós-Alió 2012] Pedrós-Alió, Carlos : The Rare Bacterial Biosphere. 4 (2012), Num. 1, p. 449–466. – URL : https://doi.org/10.1146/annurev-marine-120710-100948. – Visité le : 2018-10-08

[Peng *et al.* 2012] Peng, Yu ; Leung, Henry C. M. ; Yiu, S. M. ; Chin, Francis Y. L. : IDBA-UD : a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. 28 (2012), Num. 11, p. 1420–1428. – ISSN : 1367-4811

[Peris-Bondia *et al.* 2011] Peris-Bondia, Francesc ; Latorre, Amparo ; Artacho, Alejandro ; Moya, Andrés ; D'Auria, Giuseppe : The Active Human Gut Microbiota Differs from the Total Microbiota. 6 (2011), Num. 7, p. e22448. – URL : https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0022448. – Visité le : 2018-12-04. – ISSN : 1932-6203

[Pevzner *et al.* 2001] Pevzner, Pavel A. ; Tang, Haixu ; Waterman, Michael S. : An Eulerian path approach to DNA fragment assembly. 98 (2001), Num. 17, p. 9748–9753. – URL : http://www.pnas.org/content/98/17/9748. – Visité le : 2018-12-04. – ISSN : 0027-8424, 1091-6490

[Pividori *et al.* 2016] Pividori, Milton ; Stegmayer, Georgina ; Milone, Diego H. : Diversity control for improving the analysis of consensus clustering. Dans : *Information Sciences* 361-362 (2016), sep, p. 120–134. – URL : https://doi.org/10.1016/j.ins.2016.04.027

[Popic *et al.* 2018] Popic, Victoria ; Kuleshov, Volodymyr ; Snyder, Michael ; Batzoglou, Serafim : Fast Metagenomic Binning via Hashing and Bayesian Clustering. Dans : *Journal of Computational Biology* (2018). – URL : http://www.liebertpub.com/doi/10.1089/cmb.2017.0250. – Visité le : 2018-04-24. – ISSN : 1557-8666

[Pritchard *et al.* 2016] Pritchard, Leighton ; Glover, Rachel H. ; Humphris, Sonia ; Elphinstone, John G. ; Toth, Ian K. : Genomics and taxonomy in diagnostics for food security : soft-rotting enterobacterial plant pathogens. Dans : *Analytical Methods* 8 (2016), Num. 1, p. 12–24. – URL : https://doi.org/10.1039/c5ay02550h

[Qin *et al.* 2010] Qin, Junjie ; Li, Ruiqiang ; Raes, Jeroen ; Arumugam, Manimozhiyan ; Burgdorf, Kristoffer S. ; Manichanh, Chaysavanh ; Nielsen, Trine ; Pons, Nicolas ; Levenez, Florence ; Yamada, Takuji ; Mende, Daniel R. ; Li, Junhua ; Xu, Junming ; Li, Shaochuan ; Li, Dongfang ; Cao, Jianjun ; Wang, Bo ; Liang, Huiqing ; Zheng, Huisong ; Xie, Yinlong ; Tap, Julien ; Lepage, Patricia ; Bertalan, Marcelo ; Batto, Jean-Michel ; Hansen, Torben ; Paslier, Denis L. ; Linneberg, Allan ; Nielsen, H. B. ; Pelletier, Eric ; Renault, Pierre ; Sicheritz-Ponten, Thomas ; Turner, Keith ; Zhu, Hongmei ; Yu, Chang ; Li, Shengting ; Jian, Min ; Zhou, Yan ; Li, Yingrui ; Zhang, Xiuqing ; Li, Songgang ; Qin, Nan ; Yang, Huanming ; Wang, Jian ; Brunak, Søren ; Doré, Joel ; Guarner, Francisco ; Kristiansen, Karsten ; Pedersen, Oluf ; Parkhill, Julian ; Weissenbach, Jean ; Consortium, MetaHIT ; Antolin, Maria ; Artiguenave, François ; Blottiere, Hervé ; Borruel, Natalia ; Bruls, Thomas ; Casellas, Francesc ; Chervaux, Christian ; Cultrone, Antonella ; Delorme, Christine ; Denariaz, Gérard ; Dervyn, Rozenn ; Forte, Miguel ; Friss, Carsten ; Guchte, Maarten van d. ; Guedon, Eric ; Haimet, Florence ; Jamet, Alexandre ; Juste, Catherine ; Kaci, Ghalia ; Kleerebezem, Michiel ; Knol, Jan ; Kristensen, Michel ; Layec, Severine ; Roux, Karine L. ; Leclerc, Marion ; Maguin, Emmanuelle ; Minardi, Raquel M. ; Oozeer, Raish ; Rescigno, Maria ; Sanchez, Nicolas ; Tims, Sebastian ; Torrejon, Toni ; Varela, Encarna ; Vos, Willem d. ; Winogradsky, Yohanan ; Zoetendal, Erwin ; Bork, Peer ; Ehrlich, S. D. ; Wang, Jun : A human gut microbial gene catalogue established by metagenomic sequencing. 464 (2010), Num. 7285, p. 59–65. – URL : https://www.nature.com/articles/nature08821. – Visité le : 2018-08-22. – ISSN : 1476-4687

[Qin *et al.* 2012] Qin, Junjie ; Li, Yingrui ; Cai, Zhiming ; Li, Shenghui ; Zhu, Jianfeng ; Zhang, Fan ; Liang, Suisha ; Zhang, Wenwei ; Guan, Yuanlin ; Shen, Dongqian ; Peng, Yangqing ; Zhang, Dongya ; Jie, Zhuye ; Wu, Wenxian ; Qin, Youwen ; Xue, Wenbin ; Li, Junhua ; Han, Lingchuan ; Lu, Donghui ; Wu, Peixian ; Dai, Yali ; Sun, Xiaojuan ; Li, Zesong ; Tang, Aifa ; Zhong, Shilong ; Li, Xiaoping ; Chen, Weineng ; Xu, Ran ; Wang, Mingbang ; Feng, Qiang ; Gong, Meihua ; Yu, Jing ; Zhang, Yanyan ; Zhang, Ming ; Hansen, Torben ; Sanchez, Gaston ; Raes, Jeroen ; Falony, Gwen ; Okuda, Shujiro ; Almeida, Mathieu ; LeChatelier, Emmanuelle ; Renault, Pierre ; Pons, Nicolas ; Batto, Jean-Michel ; Zhang, Zhaoxi ; Chen, Hua ; Yang, Ruifu ; Zheng, Weimou ; Li, Songgang ; Yang,

Huanming ; Wang, Jian ; Ehrlich, S. D. ; Nielsen, Rasmus ; Pedersen, Oluf ; Kristiansen, Karsten ; Wang, Jun : A metagenome-wide association study of gut microbiota in type 2 diabetes. 490 (2012), Num. 7418, p. 55–60. – URL : https://www.nature.com/articles/nature11450. – Visité le : 2018-11-28. – ISSN : 1476-4687

[Razin et Hayflick 2010] Razin, Shmuel ; Hayflick, Leonard : Highlights of mycoplasma research—An historical perspective. 38 (2010), Num. 2, p. 183–190. – URL : http://www.sciencedirect.com/science/article/pii/S1045105609001808. – Visité le : 2018-10-05. – ISSN : 1045-1056

[Rho *et al.* 2010] Rho, Mina ; Tang, Haixu ; Ye, Yuzhen : FragGeneScan : predicting genes in short and error-prone reads. 38 (2010), Num. 20, p. e191. – ISSN : 1362-4962

[Ribière *et al.* 2016] Ribière, Céline ; Beugnot, Réjane ; Parisot, Nicolas ; Gasc, Cyrielle ; Defois, Clémence ; Denonfoux, Jérémie ; Boucher, Delphine ; Peyretaillade, Eric ; Peyret, Pierre : Targeted Gene Capture by Hybridization to Illuminate Ecosystem Functioning. 1399 (2016), p. 167–182. – ISSN : 1940-6029

[Richter *et al.* 2008] Richter, Daniel C. ; Ott, Felix ; Auch, Alexander F. ; Schmid, Ramona ; Huson, Daniel H. : MetaSim—A Sequencing Simulator for Genomics and Metagenomics. 3 (2008), Num. 10, p. e3373. – URL : http://dx.doi.org/10.1371/journal.pone.0003373. – Visité le : 2016-02-11

[Rogers *et al.* 2013] Rogers, Scott O. ; Shtarkman, Yury M. ; Koçer, Zeynep A. ; Edgar, Robyn ; Veerapaneni, Ram ; D'Elia, Tom : Ecology of Subglacial Lake Vostok (Antarctica), Based on Metagenomic/Metatranscriptomic Analyses of Accretion Ice. 2 (2013), Num. 2, p. 629–650. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3960894/. – Visité le : 2018-10-08. – ISSN : 2079-7737

[Rousseeuw 1987] Rousseeuw, Peter J. : Silhouettes : A graphical aid to the interpretation and validation of cluster analysis. 20 (1987), p. 53–65. – URL : http://www.sciencedirect.com/science/article/pii/0377042787901257. – Visité le : 2018-10-04. – ISSN : 0377-0427

[Salzberg *et al.* 2011] Salzberg, Steven L. ; Phillippy, Adam M. ; Zimin, Aleksey ; Puiu, Daniela ; Magoc, Tanja ; Koren, Sergey ; Treangen, Todd J. ; Schatz, Michael C. ; Delcher, Arthur L. ; Roberts, Michael ; Marçais, Guillaume ; Pop,

Mihai ; Yorke, James A. : GAGE : A critical evaluation of genome assemblies and assembly algorithms. (2011). – URL : http://genome.cshlp.org/content/early/2012/01/12/gr.131383.111. – Visité le : 2018-07-31. – ISSN : 1088-9051, 1549-5469

[Sanger *et al.* 1977] Sanger, F. ; Nicklen, S. ; Coulson, A. R. : DNA sequencing with chain-terminating inhibitors. 74 (1977), Num. 12, p. 5463–5467. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC431765/. – Visité le : 2018-12-03. – ISSN : 0027-8424

[Sangwan *et al.* 2016] Sangwan, Naseer ; Xia, Fangfang ; Gilbert, Jack A. : Recovering complete and draft population genomes from metagenome datasets. 4 (2016), Num. 1. – URL : http://www.microbiomejournal.com/content/4/1/8. – Visité le : 2016-03-11. – ISSN : 2049-2618

[Sarkar 2008] Sarkar, Deepayan : *Lattice : Multivariate Data Visualization with R*. Springer-Verlag, 2008 (Use R !). – URL : //www.springer.com/la/book/9780387759685. – Visité le : 2018-10-22. – ISBN : 978-0-387-75968-5

[Schölkopf 2000] Schölkopf, Bernhard : The Kernel Trick for Distances. Dans : *Proceedings of the 13th International Conference on Neural Information Processing Systems*. Cambridge, MA, USA : MIT Press, 2000 (NIPS'00), p. 283–289. – URL : http://dl.acm.org/citation.cfm?id=3008751.3008793

[Schwartz *et al.* 2015] Schwartz, Rachel S. ; Harkins, Kelly M. ; Stone, Anne C. ; Cartwright, Reed A. : A composite genome approach to identify phylogenetically informative data from next-generation sequencing. Dans : *BMC bioinformatics* 16 (2015), Num. 1, p. 193

[Schölkopf *et al.* 1998] Schölkopf, Bernhard ; Smola, Alexander ; Müller, Klaus-Robert : Nonlinear Component Analysis as a Kernel Eigenvalue Problem. 10 (1998), Num. 5, p. 1299–1319. – URL : http://www.mitpressjournals.org/doi/10.1162/089976698300017467. – Visité le : 2018-06-08. – ISSN : 0899-7667, 1530-888X

[Sczyrba *et al.* 2017] Sczyrba, Alexander ; Hofmann, Peter ; Belmann, Peter ; Koslicki, David ; Janssen, Stefan ; Dröge, Johannes ; Gregor, Ivan ; Majda, Stephan ; Fiedler, Jessika ; Dahms, Eik ; Bremges, Andreas ; Fritz, Adrian ; Garrido-Oter, Ruben ; Jørgensen, Tue S. ; Shapiro, Nicole ; Blood,

Philip D. ; Gurevich, Alexey ; Bai, Yang ; Turaev, Dmitrij ; DeMaere, Matthew Z. ; Chikhi, Rayan ; Nagarajan, Niranjan ; Quince, Christopher ; Meyer, Fernando ; Balvočiūtė, Monika ; Hansen, Lars H. ; Sørensen, Søren J. ; Chia, Burton K. H. ; Denis, Bertrand ; Froula, Jeff L. ; Wang, Zhong ; Egan, Robert ; Kang, Dongwan D. ; Cook, Jeffrey J. ; Deltel, Charles ; Beckstette, Michael ; Lemaitre, Claire ; Peterlongo, Pierre ; Rizk, Guillaume ; Lavenier, Dominique ; Wu, Yu-Wei ; Singer, Steven W. ; Jain, Chirag ; Strous, Marc ; Klingenberg, Heiner ; Meinicke, Peter ; Barton, Michael D. ; Lingner, Thomas ; Lin, Hsin-Hung ; Liao, Yu-Chieh ; Silva, Genivaldo Gueiros Z. ; Cuevas, Daniel A. ; Edwards, Robert A. ; Saha, Surya ; Piro, Vitor C. ; Renard, Bernhard Y. ; Pop, Mihai ; Klenk, Hans-Peter ; Göker, Markus ; Kyrpides, Nikos C. ; Woyke, Tanja ; Vorholt, Julia A. ; Schulze-Lefert, Paul ; Rubin, Edward M. ; Darling, Aaron E. ; Rattei, Thomas ; McHardy, Alice C. : Critical Assessment of Metagenome Interpretation—a benchmark of metagenomics software. 14 (2017), Num. 11, p. 1063–1071. – URL : https://www.nature.com/articles/nmeth.4458. – Visité le : 2018-08-17. – ISSN : 1548-7105

[Sedlar *et al.* 2017] Sedlar, Karel ; Kupkova, Kristyna ; Provaznik, Ivo : Bioinformatics strategies for taxonomy independent binning and visualization of sequences in shotgun metagenomics. 15 (2017), p. 48–55. – URL : http://www.sciencedirect.com/science/article/pii/S2001037016300678. – Visité le : 2018-07-18. – ISSN : 2001-0370

[Segata *et al.* 2012] Segata, Nicola ; Waldron, Levi ; Ballarini, Annalisa ; Narasimhan, Vagheesh ; Jousson, Olivier ; Huttenhower, Curtis : Metagenomic microbial community profiling using unique clade-specific marker genes. 9 (2012), Num. 8, p. 811–814. – URL : https://www.nature.com/articles/nmeth.2066. – Visité le : 2018-10-08. – ISSN : 1548-7105

[Shaiber et Eren 2019] Shaiber, Alon ; Eren, A M. : Composite Metagenome-Assembled Genomes Reduce the Quality of Public Genome Repositories. Dans : *mBio* 10 (2019), Num. 3, p. e00725–19

[Shreiner *et al.* 2015] Shreiner, Andrew B. ; Kao, John Y. ; Young, Vincent B. : The gut microbiome in health and in disease. 31 (2015), Num. 1, p. 69–75. – ISSN : 1531-7056

[Sieber *et al.* 2018] Sieber, Christian M. K. ; Probst, Alexander J. ; Sharrar, Allison ; Thomas, Brian C. ; Hess, Matthias ; Tringe, Susannah G. ; Banfield,

Jillian F. : Recovery of genomes from metagenomes via a dereplication, aggregation and scoring strategy. 3 (2018), Num. 7, p. 836–843. – URL : https://www.nature.com/articles/s41564-018-0171-1. – Visité le : 2018-08-19. – ISSN : 2058-5276

[Sime-Ngando *et al.* 2016] Sime-Ngando, Telesphore ; Boivin, Pierre ; Chapron, Emmanuel ; Jezequel, Didier ; Meybeck, Michel : *Lake Pavin : History, geology, biogeochemistry, and sedimentology of a deep meromictic maar lake*. 2016. – URL : //www.springer.com/la/book/9783319399607. – Visité le : 2018-12-05

[Smets et Kennes 2008] Smets, Philippe ; Kennes, Robert : The Transferable Belief Model. Dans : Yager, Roland R. (Éditeur) ; Liu, Liping (Éditeur) : *Classic Works of the Dempster-Shafer Theory of Belief Functions*. Springer Berlin Heidelberg, 2008 (Studies in Fuzziness and Soft Computing), p. 693–736. – URL : https://doi.org/10.1007/978-3-540-44792-4_28. – Visité le : 2018-12-14. – ISBN : 978-3-540-44792-4

[Sohn et Nam 2016] Sohn, Jang-Il ; Nam, Jin-Wu : The present and future of de novo whole-genome assembly. (2016). – ISSN : 1477-4054

[Song et Thomas 2017] Song, Wei-Zhi ; Thomas, Torsten : Binning_refiner : improving genome bins through the combination of different binning programs. 33 (2017), Num. 12, p. 1873–1875. – URL : https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btx086. – Visité le : 2017-12-16. – ISSN : 1367-4803, 1460-2059

[Strous *et al.* 2012] Strous, Marc ; Kraft, Beate ; Bisdorf, Regina ; Tegetmeyer, Halina E. : The Binning of Metagenomic Contigs for Microbial Physiology of Mixed Cultures. 3 (2012). – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3514610/. – Visité le : 2017-10-24. – ISSN : 1664-302X

[Sunagawa *et al.* 2015] Sunagawa, Shinichi ; Coelho, Luis P. ; Chaffron, Samuel ; Kultima, Jens R. ; Labadie, Karine ; Salazar, Guillem ; Djahanschiri, Bardya ; Zeller, Georg ; Mende, Daniel R. ; Alberti, Adriana ; Cornejo-Castillo, Francisco M. ; Costea, Paul I. ; Cruaud, Corinne ; d'Ovidio, Francesco ; Engelen, Stefan ; Ferrera, Isabel ; Gasol, Josep M. ; Guidi, Lionel ; Hildebrand, Falk ; Kokoszka, Florian ; Lepoivre, Cyrille ; Lima-Mendez, Gipsi ; Poulain, Julie ; Poulos, Bonnie T. ; Royo-Llonch, Marta ; Sarmento, Hugo ; Vieira-Silva, Sara ; Dimier, Céline ; Picheral, Marc ;

Searson, Sarah ; Kandels-Lewis, Stefanie ; Coordinators, Tara O. ; Bowler, Chris ; Vargas, Colomban d. ; Gorsky, Gabriel ; Grimsley, Nigel ; Hingamp, Pascal ; Iudicone, Daniele ; Jaillon, Olivier ; Not, Fabrice ; Ogata, Hiroyuki ; Pesant, Stephane ; Speich, Sabrina ; Stemmann, Lars ; Sullivan, Matthew B. ; Weissenbach, Jean ; Wincker, Patrick ; Karsenti, Eric ; Raes, Jeroen ; Acinas, Silvia G. ; Bork, Peer : Structure and function of the global ocean microbiome. 348 (2015), Num. 6237, p. 1261359. – URL : http://science.sciencemag.org/content/348/6237/1261359. – Visité le : 2018-11-28. – ISSN : 0036-8075, 1095-9203

[Teeling *et al.* 2004] Teeling, Hanno ; Waldmann, Jost ; Lombardot, Thierry ; Bauer, Margarete ; Glöckner, Frank O. : TETRA : a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences. 5 (2004), p. 163. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC529438/. – Visité le : 2018-08-11. – ISSN : 1471-2105

[Terrat 2010] Terrat, Sébastien : *New design of probes for functional DNA microarrays and characterization of the biodegradation capacities of bacterial communities in hydrocarbon polluted soils*, Université Blaise Pascal - Clermont-Ferrand II, Thèse de doctorat, october 2010. – URL : https://tel.archives-ouvertes.fr/tel-00629656

[Thursby et Juge 2017] Thursby, Elizabeth ; Juge, Nathalie : Introduction to the human gut microbiota. 474 (2017), Num. 11, p. 1823–1836. – URL : http://www.biochemj.org/content/474/11/1823. – Visité le : 2018-11-28. – ISSN : 0264-6021, 1470-8728

[Turnbaugh *et al.* 2009] Turnbaugh, Peter J. ; Hamady, Micah ; Yatsunenko, Tanya ; Cantarel, Brandi L. ; Duncan, Alexis ; Ley, Ruth E. ; Sogin, Mitchell L. ; Jones, William J. ; Roe, Bruce A. ; Affourtit, Jason P. ; Egholm, Michael ; Henrissat, Bernard ; Heath, Andrew C. ; Knight, Rob ; Gordon, Jeffrey I. : A core gut microbiome in obese and lean twins. 457 (2009), Num. 7228, p. 480–484. – URL : https://www.nature.com/articles/nature07540. – Visité le : 2018-12-05. – ISSN : 1476-4687

[Turnbaugh *et al.* 2007] Turnbaugh, Peter J. ; Ley, Ruth E. ; Hamady, Micah ; Fraser-Liggett, Claire ; Knight, Rob ; Gordon, Jeffrey I. : The human microbiome project : exploring the microbial part of ourselves in a changing world.

449 (2007), Num. 7164, p. 804–810. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3709439/. – Visité le : 2018-08-28. – ISSN : 0028-0836

[Tyson *et al.* 2004] Tyson, Gene W. ; Chapman, Jarrod ; Hugenholtz, Philip ; Allen, Eric E. ; Ram, Rachna J. ; Richardson, Paul M. ; Solovyev, Victor V. ; Rubin, Edward M. ; Rokhsar, Daniel S. ; Banfield, Jillian F. : Community structure and metabolism through reconstruction of microbial genomes from the environment. 428 (2004), Num. 6978, p. 37–43. – URL : https://www.nature.com/articles/nature02340. – Visité le : 2018-09-21. – ISSN : 1476-4687

[Ulyanov 2016] Ulyanov, Dmitry : *Multicore-TSNE*. https://github.com/DmitryUlyanov/Multicore-TSNE. 2016

[Uritskiy *et al.* 2018] Uritskiy, Gherman V. ; DiRuggiero, Jocelyne ; Taylor, James : MetaWRAP—a flexible pipeline for genome-resolved metagenomic data analysis. 6 (2018), Num. 1, p. 158. – URL : https://doi.org/10.1186/s40168-018-0541-1. – Visité le : 2018-10-01. – ISSN : 2049-2618

[Ursell et Knight 2013] Ursell, Luke K. ; Knight, Rob : Xenobiotics and the Human Gut Microbiome : Metatranscriptomics Reveal the Active Players. 17 (2013), Num. 3, p. 317–318. – URL : http://www.sciencedirect.com/science/article/pii/S1550413113000612. – Visité le : 2018-12-04. – ISSN : 1550-4131

[Van Der Maaten 2014] Van Der Maaten, Laurens : Accelerating t-SNE using tree-based algorithms. Dans : *The Journal of Machine Learning Research* 15 (2014), Num. 1, p. 3221–3245

[Van Der Maaten et Hinton 2008] Van Der Maaten, Laurens ; Hinton, Geoffrey : Visualizing data using t-SNE. Dans : *Journal of machine learning research* 9 (2008), Num. Nov, p. 2579–2605

[Vega-Pons et Ruiz-Shulcloper 2011] Vega-Pons, Sandro ; Ruiz-Shulcloper, José : A survey of clustering ensemble algorithms. 25 (2011), Num. 3, p. 337–372. – URL : https://www.worldscientific.com/doi/abs/10.1142/S0218001411008683. – Visité le : 2018-08-25. – ISSN : 0218-0014

[Vollmers *et al.* 2017] Vollmers, John ; Wiegand, Sandra ; Kaster, Anne-Kristin : Comparing and Evaluating Metagenome Assembly Tools from a Microbiologist's Perspective - Not Only Size Matters ! 12 (2017), Num. 1, p. e0169662.

– URL : http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0169662. – Visité le : 2018-07-30. – ISSN : 1932-6203

[Wagstaff *et al.* 2001] Wagstaff, Kiri ; Cardie, Claire ; Rogers, Seth ; Schrödl, Stefan *et al.* : Constrained k-means clustering with background knowledge. Dans : *Proceedings of the Eighteenth International Conference on Machine Learning* Vol. 1. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2001, p. 577–584. – URL : https://dl.acm.org/citation.cfm?id=655669. – Visité le : 2019. – ISBN : 1-55860-778-1

[van der Walt *et al.* 2017] Walt, Andries J. van der ; Goethem, Marc W. van ; Ramond, Jean-Baptiste ; Makhalanyane, Thulani P. ; Reva, Oleg ; Cowan, Don A. : Assembling metagenomes, one community at a time. 18 (2017), Num. 1, p. 521. – URL : https://doi.org/10.1186/s12864-017-3918-9. – Visité le : 2018-11-28. – ISSN : 1471-2164

[Wang *et al.* 2012] Wang, Y. ; Leung, H. C. M. ; Yiu, S. M. ; Chin, F. Y. L. : MetaCluster 5.0 : a two-round binning approach for metagenomic data for low-abundance species in a noisy sample. 28 (2012), Num. 18, p. i356–i362. – URL : http://bioinformatics.oxfordjournals.org/cgi/doi/10.1093/bioinformatics/bts397. – Visité le : 2016-02-05. – ISSN : 1367-4803, 1460-2059

[Wang *et al.* 2015] Wang, Ying ; Hu, Haiyan ; Li, Xiaoman : MBBC : an efficient approach for metagenomic binning based on clustering. 16 (2015). – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4339733/. – Visité le : 2017-10-24. – ISSN : 1471-2105

[Waters *et al.* 2003] Waters, E. ; Hohn, M. J. ; Ahel, I. ; Graham, D. E. ; Adams, M. D. ; Barnstead, M. ; Beeson, K. Y. ; Bibbs, L. ; Bolanos, R. ; Keller, M. ; Kretz, K. ; Lin, X. ; Mathur, E. ; Ni, J. ; Podar, M. ; Richardson, T. ; Sutton, G. G. ; Simon, M. ; Soll, D. ; Stetter, K. O. ; Short, J. M. ; Noordewier, M. : The genome of Nanoarchaeum equitans : Insights into early archaeal evolution and derived parasitism. Dans : *Proceedings of the National Academy of Sciences* 100 (2003), oct, Num. 22, p. 12984–12988. – URL : https://doi.org/10.1073/pnas.1735403100

[Wayne *et al.* 1987] Wayne, L. G. ; Moore, W. E. C. ; Stackebrandt, E. ; Kandler, O. ; Colwell, R. R. ; Krichevsky, M. I. ; Truper, H. G. ;

Murray, R. G. E. ; Grimont, P. A. D. ; Brenner, D. J. ; Starr, M. P. ; Moore, L. H. : Report of the Ad Hoc Committee on Reconciliation of Approaches to Bacterial Systematics. Dans : *International Journal of Systematic and Evolutionary Microbiology* 37 (1987), oct, Num. 4, p. 463–464. – URL : https://doi.org/10.1099/00207713-37-4-463

[Whitman *et al.* 1998] Whitman, William B. ; Coleman, David C. ; Wiebe, William J. : Prokaryotes : The unseen majority. 95 (1998), Num. 12, p. 6578–6583. – URL : http://www.pnas.org/content/95/12/6578. – Visité le : 2017-11-24. – ISSN : 0027-8424, 1091-6490

[Wood et Salzberg 2014] Wood, Derrick E. ; Salzberg, Steven L. : Kraken : ultrafast metagenomic sequence classification using exact alignments. 15 (2014), Num. 3, p. R46. – URL : https://doi.org/10.1186/gb-2014-15-3-r46. – Visité le : 2018-11-28. – ISSN : 1474-760X

[Wu *et al.* 2016] Wu, Yu-Wei ; Simmons, Blake A. ; Singer, Steven W. : MaxBin 2.0 : an automated binning algorithm to recover genomes from multiple metagenomic datasets. 32 (2016), Num. 4, p. 605–607. – ISSN : 1367-4811

[Wu et Ye 2011] Wu, Yu-Wei ; Ye, Yuzhen : A Novel Abundance-Based Algorithm for Binning Metagenomic Sequences Using l-tuples. 18 (2011), Num. 3, p. 523–534. – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3123841/. – Visité le : 2018-08-11. – ISSN : 1066-5277

[Xanthopoulos 2014] Xanthopoulos, Petros : A Review on Consensus Clustering Methods. Dans : Rassias, Themistocles M. (Éditeur) ; Floudas, Christodoulos A. (Éditeur) ; Butenko, Sergiy (Éditeur) : *Optimization in Science and Engineering : In Honor of the 60th Birthday of Panos M. Pardalos*. Springer New York, 2014, p. 553–566. – URL : https://doi.org/10.1007/978-1-4939-0808-0_26. – Visité le : 2018-08-22. – ISBN : 978-1-4939-0808-0

[Xie *et al.* 2017] Xie, Xian-Hua ; Yu, Zu-Guo ; Ma, Yuan-Lin ; Han, Guo-Sheng ; Anh, Vo : A novel genome signature based on inter-nucleotide distances profiles for visualization of metagenomic data. 482 (2017), p. 87–94. – URL : http://www.sciencedirect.com/science/article/pii/S037843711730359X. – Visité le : 2017-10-24. – ISSN : 0378-4371

[Xu et Zhao 2018] Xu, Yuan ; Zhao, Fangqing : Single-cell metagenomics : challenges and applications. 9 (2018), Num. 5, p. 501–510. – URL : https:

//doi.org/10.1007/s13238-018-0544-5. – Visité le : 2018-12-04. – ISSN : 1674-8018

[Ye *et al.* 2016] Ye, Chengxi ; Hill, Christopher M. ; Wu, Shigang ; Ruan, Jue ; Ma, Zhanshan (. : DBG2OLC : Efficient Assembly of Large Genomes Using Long Erroneous Reads of the Third Generation Sequencing Technologies. 6 (2016), p. 31900. – URL : https://www.nature.com/articles/srep31900. – Visité le : 2018-11-28. – ISSN : 2045-2322

[Yu *et al.* 2018] Yu, Guoxian ; Jiang, Yuan ; Wang, Jun ; Zhang, Hao ; Luo, Haiwei ; Berger, Bonnie : BMC3C : binning metagenomic contigs using codon usage, sequence composition and read coverage. 34 (2018), Num. 24, p. 4172–4179. – URL : https://academic.oup.com/bioinformatics/advance-article/doi/10.1093/bioinformatics/bty519/5045915. – Visité le : 2018-08-27

[Yáñez-Ruiz *et al.* 2015] Yáñez-Ruiz, David R. ; Abecia, Leticia ; Newbold, Charles J. : Manipulating rumen microbiome and fermentation through interventions during early life : a review. 6 (2015). – URL : https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4604304/. – Visité le : 2018-07-24. – ISSN : 1664-302X

[Zeng *et al.* 2015] Zeng, Zhiqiang ; Shi, Hua ; Wu, Yun ; Hong, Zhiling : Survey of natural language processing techniques in bioinformatics. Dans : *Computational and mathematical methods in medicine* 2015 (2015)

[Zhao *et al.* 2009] Zhao, Zhonghua ; Guo, Shanqing ; Xu, Qiuliang ; Ban, Tao : G-Means : A Clustering Algorithm for Intrusion Detection. Dans : Köppen, Mario (Éditeur) ; Kasabov, Nikola (Éditeur) ; Coghill, George (Éditeur) : *Advances in Neuro-Information Processing*, Springer Berlin Heidelberg, 2009 (Lecture Notes in Computer Science), p. 563–570. – ISBN : 978-3-642-02490-0

[Zhou *et al.* 2013] Zhou, Yanjiao ; Gao, Hongyu ; Mihindukulasuriya, Kathie A. ; La Rosa, Patricio S. ; Wylie, Kristine M. ; Vishnivetskaya, Tatiana ; Podar, Mircea ; Warner, Barb ; Tarr, Phillip I. ; Nelson, David E. ; Fortenberry, J. D. ; Holland, Martin J. ; Burr, Sarah E. ; Shannon, William D. ; Sodergren, Erica ; Weinstock, George M. : Biogeography of the ecosystems of the healthy human body. 14 (2013), Num. 1, p. R1. – URL : https://doi.org/10.1186/gb-2013-14-1-r1. – Visité le : 2018-08-27. – ISSN : 1474-760X

[Zielezinski *et al.* 2017] Zielezinski, Andrzej ; Vinga, Susana ; Almeida, Jonas ; Karlowski, Wojciech M. : Alignment-free sequence comparison : benefits, applications, and tools. 18 (2017), Num. 1, p. 186. – URL : https://doi.org/10.1186/s13059-017-1319-7. – Visité le : 2018-08-30. – ISSN : 1474-760X

[Zimin *et al.* 2013] Zimin, Aleksey V. ; Marçais, Guillaume ; Puiu, Daniela ; Roberts, Michael ; Salzberg, Steven L. ; Yorke, James A. : The MaSuRCA genome assembler. 29 (2013), Num. 21, p. 2669–2677. – URL : https://academic.oup.com/bioinformatics/article/29/21/2669/195975. – Visité le : 2018-07-30. – ISSN : 1367-4803

A | Annexes

A.1 Développement des technologies de séquençage

A.1.1 Première génération

Le séquençage par la méthode de Sanger (Sanger *et al.*, 1977), dit de première génération, a permis d'établir les génomes de référence comme celui de *Haemophilus influenza* (Fleischmann *et al.*, 1995) ainsi que d'autres organismes modèles (Arabidopsis Genome Initiative, 2000; International Human Genome Sequencing Consortium, 2001), permettant ainsi de mieux appréhender la biologie de ces organismes, mais aussi la recherche de biomarqueurs d'intérêt biotechnologies et/ou cliniques. Brièvement, la technologie Sanger technologies permet de marquer un fragment de la molécule à séquencer, par arrêt de la polymérisation à l'aide de didésoxynucléotides marqués. Les fragments sont alors séparés par électrophorèse sur gel permettant ainsi de déduire la séquence nucléotidique de la molécule. Bien que de considérables avancées aient été réalisées dans la compréhension de la structure et du fonctionnement des génomes grâce au séquençage, ces travaux restaient longs et producteurs d'une quantité très limitée de données pour des organismes isolés et cultivables en condition de laboratoire.

A.1.2 Deuxième génération

Le continuel développement des méthodes de séquençage des acides nucléiques, notamment par la miniaturisation et la parallélisation très massive des réactions de séquençage, ont permis l'éclosion de la deuxième génération des technologies de séquençage, dites « à haut débit », incarnée principalement par le pyroséquençage (Roche), le séquençage par terminateur réversible (Illumina) et le séquençage par semi-conducteurs ionique (IonTorrent) (Allali *et al.*, 2017). La première technologie incorpore les nucléotides un à un lors de la polymérisation d'un brin complémentaire à un brin matrice en exploitant la capacité du couple luciférine/luciférase à émettre un signal lumineux en utilisant les pyrophosphates libérés lors de l'incorporation des nucléotides. Le deuxième technologie polymérise le brin complémentaire avec des terminateurs réversibles, permettant ainsi de « mettre en pause » la réaction de polymérisation pendant la détection du signal. En effet, chaque type de nucléotide est marqué par un fluorochrome spécifique qu'il est possible d'exciter après son incorporation. La troisième technologie repose sur le fait qu'un électron est libéré lors de l'incorporation d'un nucléotide lors de la polymérisation du brin complémentaire,

induisant un changement de pH de la solution. Les variations de pH sont mesurées alors qu'on apporte les nucléotides un à un, permettant ainsi de savoir quel type de nucléotide a été incorporé. Ces trois technologies dites de « séquençage par synthèse » ont également la particularité d'être très massivement parallélisables. Ces réactions sont réalisées sur des plaques munies de micro-puits ou sur des lames de silicium où l'on peut effectuer plusieurs millions de réactions simultanément, offrant un débit de séquençage important, même si les lectures sont de tailles relativement modestes (de 75 à 300 bases). La technologie d'Illumina est actuellement la plus utilisée. Les technologies de seconde génération sont toujours en développement.

A.1.3 Troisième génération

La troisième génération de séquençage est encore en développement et a pour but de proposer des lectures beaucoup plus longues directement à partir d'une seule molécule. On s'affranchit donc des biais induits par l'amplification du matériel génétique et on peut envisager la résolution de structures génomiques jusqu'alors inexploitables (*e. g.* : régions répétées chez les eucaryotes) avec les deux premières générations. Les technologies principales sont le séquençage single molecule real time (SMRT ; Pacific Biosciences) et le séquençage par nanopore (Oxford Nanopore Technologies). Dans le premier cas, une ADN polymérase est fixée au fond d'un puits ayant un volume de l'ordre du zeptolitre et la molécule à séquencer est apportée avec des nucléotides dont le phosphate terminal porte un fluorochrome spécifique à chaque type de nucléotides. En cas d'incorporation du nucléotide, le fluorochrome sera libéré à proximité d'un guide d'ondes lumineuses adapté à ce type de puits (le « *zero-mode waveguide* ») qui va permettre la mesure de la longueur d'onde émise par le fluorochrome. Dans le deuxième cas, des nanopores sont synthétisées à partir de protéines transmembranaires ou d'un substrat solide (*e. g.* : graphène), puis une molécule d'ADN linéaire va traverser le pore (Haque *et al.*, 2013). Or, il est possible de mesurer le courant électrique du canal du nanopore, et donc de connaître le type de nucléotide qui est en train de le traverser. Les applications à l'étude des génomes de cette troisième génération de séquençage restent toutefois encore limitées du fait de la qualité moindre des séquences produites et du débit plus faible que celui de la deuxième génération (Edwards *et al.*, 2018; Leggett *et al.*, 2018) mais permettent de compléter les données issues de la deuxième génération de séquençage pour l'assemblage de génomes (Ye *et al.*, 2016).

A.1.4 Informations apportées

A.1.4.1 Score de qualité

En plus de connaitre une portion de la séquence nucléotidique d'un matériel génétique donné, ces technologies de séquençage apporte l'information relative à la fiabilité de la lecture.

La qualité de lecture d'un nucléotide prend la forme d'un score. Ce score est défini par le standard Phred comme suit : $Q = -10 \times log_{10}(P)$, avec P la probabilité de lecture incorrecte. Chaque technologie et chaque constructeur dispose de sa propre méthodologie pour définir cette probabilité.

En pratique, les scores de qualité Phred couvre un intervalle de 0 à 50. Ce dernier représente une probabilité de lecture correcte de 99,999 %. Un nucléotide est généralement considéré comme de bonne qualité si son score est supérieur à 30 (variable selon les technologies). Une lecture est donc considérée comme de bonne qualité si la moyenne des scores Phred des nucléotides qui la compose est supérieure à 30.

A.1.4.2 Lecture appariées

D'autre part, le séquençage du matériel génétique peut être réalisé en séquençant les deux extremités d'un brin d'ADN. Ainsi, après avoir physiquement fragementés le matériel génétique et selectionné les brins d'une longueur donnée, il est possible de connaitre des paires de lectures et la distance qui les séparent. On parlera de « *paired-end reads* ».

A.2 Exploration de la diversité microbienne *in situ* nécessitant des dispositions expérimentales particulières

La diversité et la complexité biologique des microbiomes ne permettent pas de décrire exhaustivement leur contenu génétique. Ainsi, plusieurs dispositions expérimentales spécifiques visant à réduire la complexité de l'échantillon environnemental ont été développées pour répondre à des questions ciblant un certain niveau d'analyse du métagénome – génomes individuels ou population globale. Cette dichotomie, souvent nécessaire pour des raisons pratiques, limite cependant l'exploration du

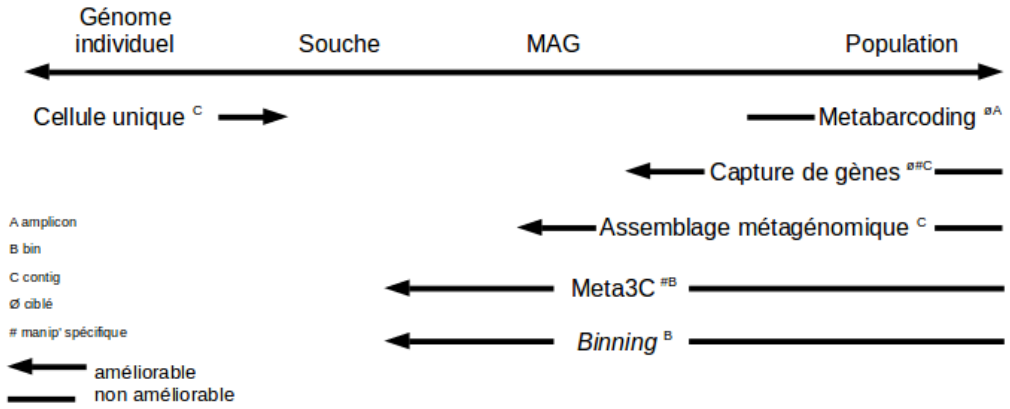


Figure A.1 | **Niveau d'analyse permis (du génome à la population) de la diversité microbienne directement depuis l'environnement**.
• Les approches cellules unique permettent la caractérisation de génomes individuels, voire de souche représentatives si suffisamment d'individus sont séquences. • Le métabarcoding est limité à l'étude de la population du fait de l'amplification d'un marqueur. • La capture de gènes permet le séquençage des régions flanquantes d'un marqueur et nécessite la production de données nouvelles. Elle peut produire des contigs. • L'assemblage des lectures de séquençage permet la construction de contigs représentant potentiellement des MAG si la profondeur de séquençage le permet. • Meta3C permet la construction de contigs regroupés en bins mais nécessite la production de données dédiées. • Le binning permet la construction de bins à partir des contigs issu de l'assemblage du métagénome.

microbiote en freinant (voire en empêchant) ces changements d'échelle (figure A.1).

A.2.1 Approche par enrichissement en micro-organismes d'intérêt

L'étude d'un métagénome peut également être réalisée dans le but d'identifier des micro-organismes assurant une fonction biologique d'intérêt. Pour cela, un échantillon environnemental est soumis à un traitement sélectionnant les micro-organismes spécifiques à l'activité biologique étudiée (*e. g.* : résistance à un antibiotique en l'exposant à cet antibiotique). Cette méthode dite « par enrichissement » est notamment utilisée dans le cadre de l'étude de micro-organismes ou de gènes rares voire dans les conditions normales de l'environnement étudié (Noyes *et al.*, 2017; Guerra *et al.*, 2018). Cette réduction de la diversité d'un microbiome est donc particulièrement utile lorsque l'effort de séquençage nécessaire est trop important si il est appliqué directement sur l'environnement naturel. Cependant, cette approche nécessite un *a priori* fort sur ce qui est recherché et demande de maintenir le microbiome dans

des conditions adaptées aux populations recherchées. Une fois obtenu, ce nouveau microbiome peut être analysé par les différentes méthodes décrites ci-après, pour par exemple établir l'inventaire des gènes responsables de l'activité biologique d'intérêt ou identifier les micro-organismes en présence.

A.2.2 Métataxonomique et métabarcoding

Les approches de métataxonomique et de métabarcoding permettent de quantifier un marqueur phylogénétique ou fonctionnel d'un microbiote mais ne permettent pas d'obtenir en même temps ces deux informations. Il n'est donc pas possible d'établir clairement le lien entre la structure et les fonctions de ce microbiote par ces approches. Ce type de données nous cantonne à une analyse de la population dans sa globalité. Il est cependant possible de prédire *in silico* des fonctions biologiques à partir de gènes utilisables également comme marqueurs phylogénétiques (Langille *et al.*, 2013) ou de chercher des associations statistiques entre les abondances des espèces et des niveaux d'expression de gènes (Abubucker *et al.*, 2012; Batut *et al.*, 2018). Ces approches prédictives se basent sur des connaissances *a priori* et sur des corrélations statistiques assumant que les micro-organismes les plus abondants sont les plus actifs métaboliquement mais ne prouvant pas la causalité (Turnbaugh *et al.*, 2009; Peris-Bondia *et al.*, 2011; Ursell et Knight, 2013).

A.2.3 Capture de gènes

La capture de gènes par hybridation permet la sélection de certaines séquences issues d'un métagénome à l'aide de sondes oligonucléotidiques. Le génome est en effet fragmenté puis seuls les fragments s'hybridant aux sondes sont conservés puis séquencés, permettant ainsi l'accès aux séquences des régions flanquantes de la séquence ciblée par les sondes (usuellement un gène), et d'obtenir des informations sur le contexte génomique du gène d'intérêt. Cette approche a été appliquée à l'étude d'un microbiome permettant de caractériser la dynamique du gène mcrA dans un milieu lacustre afin de mieux appréhender la diversité fonctionnelle des espèces méthanogènes (Denonfoux *et al.*, 2013). L'utilisation de sondes oligonucléotidiques reposent sur des bases de connaissances et peuvent donc limiter l'aspect exploratoire de l'approche (Militon *et al.*, 2007; Parisot *et al.*, 2012; Parisot, 2014). L'augmentation de la longueur des fragments d'ADN capturés et séquencés permet de révéler le gène cible ainsi que son contexte génomique. Des contigs de plus de 70 kb ont pu ainsi être reconstruits sur la seule base de la connaissance d'un gène cible (Gasc et

Peyret, 2017). En appliquant cette stratégie de capture de grands fragments à un métagénome en ciblant initialement un marqueur phylogénétique (*e. g.* : ADNr 16S), il devient envisageable d'obtenir en une seule expérience des informations relatives à la structure et à quelques fonctions biologiques d'une communauté (Gasc et Peyret, 2018).

A.2.4 Capture de la conformation chromosomique

La capture de la conformation chromosomique (3C) est une technique permettant de connaître l'organisation spatiale d'un génome par la cartographie des points de contacts physiques entre différents loci (Dekker *et al.*, 2002). Couplée à une méthode de séquençage à haut débit, il devient alors possible de connaître la séquence des régions génomiques physiquement en contact (Hughes *et al.*, 2014). Marbouty et al. ont appliqué la méthode 3C au séquençage à haut débit de ces points et à une approche métagénomique (meta3C) (Marbouty *et al.*, 2014). Ainsi, il a été possible de discriminer sans *a priori* différentes portions de génomes issues de métagénomes simulés mais également de reconstruire des génomes issus du microbiote des sédiments de la Seine directement depuis leur environnement. Un protocole expérimental spécifique est cependant nécessaire.

A.2.5 Approche « cellule unique »

L'amplification par déplacement multiple (Multiple displacement amplification, ou MDA) permet l'amplification de quelques femtogrammes d'ADN. Ainsi, il est possible d'amplifier suffisamment le matériel génétique d'une unique cellule pour permettre son séquençage par des approches à haut débit. Cette stratégie de séquençage de cellule unique (single cell sequencing, SCS) (Gawad *et al.*, 2016; Angerer *et al.*, 2017) est applicable à une cellule extraite physiquement d'un environnement pour accéder à son génome (single cell metagenomics) (Xu et Zhao, 2018). L'approche n'est donc pas une approche aussi globale que la métagénomique puisqu'elle ne cible pas l'échelle de la population mais permet néanmoins d'étudier *in situ* les génomes microbiens et notamment des génomes peu abondants. De nouvelles plateformes ont été développées afin d'augmenter le débit de l'approche (Chen *et al.*, 2018) permettant alors d'envisager le séquençage de nombreux génomes d'individus issus d'un microbiome complexe. Certaines méthodes permettent de construire une librairie d'ADN prête à être séquencée à partir de 50 000 cellules par run (Demaree *et al.*, 2018), ce qui reste bien en deçà du nombre de cellules microbiennes que l'on

peut retrouver dans un sol ou un tube digestif (jusqu'à 10^{14} cellules composent le microbiote intestinal humain (Thursby et Juge, 2017)). Bien que très prometteuse, cette approche n'autorise pas encore un débit assez important pour l'exploration de microbiomes complexes à l'échelle de la population.

A.2.6 Limites des approches par réduction de complexité

L'obtention des génomes complets d'un microbiote permettrait de clairement établir le lien entre la structure du microbiote et ses fonctions et ainsi de savoir quelle espèce assure quelles fonctions biologiques. Les stratégies précédemment citées, couplées à un séquençage à haut débit, reposent soit sur de forts *a priori* sur ce qui est recherché (métabarcoding, capture de gènes) limitant donc la capacité d'exploration des microbiomes ; soit sur des dispositions expérimentales particulières et la production de données nouvelles (enrichissement, méta3C, cellule unique). Ainsi, l'étude exploratoire d'un microbiote dans ses conditions environnementales, et notamment pour pouvoir l'étudier à la fois à l'échelle de la population et à l'échelle des génomes individuels, devra donc recourir à une approche métagénomique globale. Une analyse innovante de ces données est une solution pour (i) traiter de nouvelles données ou (ii) procéder à une fouille de données dans les bases publiques (*i. e.* : méta-analyse).

A.3 Complexités algorithmiques

Tableau A.1 | **Algorithmes utilisés par les outils de binning**, leur complexité temporelle et paramètres requis et ajustables, avec le nombre d'individus n, nombre de dimensions d, nombre de clusters ou composantes k, nombre d'itérations i, nombre d'itérations de G-means g, nombre d'itérations pour l'approximation variationelle v. Les paramètres requis sont nécessaires pour pouvoir exécuter l'algorithme et peuvent être estimés ou fournis par l'utilisateur. Les paramètres ajustables sont des paramètres pour lesquels une valeur par défaut existe et se suffit à elle-même. Les paramètres n et d sont des paramètres de la complexité temporelle des algorithmes qui influence le temps d'exécution de l'algorithme mais qui lui sont extérieurs.

Algorithme	**Logiciels**	**Complexité temporelle**	**Paramètres requis**	**Paramètres ajustables**
k-means	– COCACOLA – MetaProb – MetaCluster5	$O(nkdi)$	k	i
E-M	– MaxBin	$O(nkdi)$	k	i
NMF	– COCACOLA	$O(ndk)$	–	k
G-means	– MetaProb	$O(nkdig)$	–	i, g
VBGMM	– CONCOCT – GATTACA	$O(nkdiv)$	–	k, i, v
k-médoïdes	– MetaBAT – MetaBAT2	$O((n-k)^2 \times id)$	k	i
AP	– AbundanceBin – BinSanity – MyCC	$O(n^2kd)$	–	–
AP	– GroopM	$O(n^2)$	–	–
ACP	– CONCOCT – GATTACA	$O(max(n,d)^2 \times min(n,d))$	d	–
BH-tSNE	– VizBin	$O(n \times log(n))$	–	–

A.4 Calcul des métriques pour l'évaluation du binning

Diverses métriques peuvent être calculés après association des bins proposés aux bins de références puis construction d'une matrice de confusion :

- Vrai positif (TP) : nombre d'individus retrouvés dans le bin de référence et le bin associé ;
- Vrai négatif (TN) : nombre d'individus non retrouvés dans le bin de référence ni dans le bin associé ;
- Faux positif (FP) : nombre d'individus non retrouvés dans le bin de référence alors que retrouvés dans le bin associé ;
- Vrai négatif (TN) : nombre d'individus retrouvés dans le bin de référence mais non retrouvés dans le bin associé ;
- Sensibilité = $\frac{TP}{FP+FN}$, ou *recall, true positive rate*
- Sépcificité = $\frac{TN}{TN+FP}$
- Précision = $\frac{TP}{TN+FP}$
- Taux de faux positif (FPR) = $\frac{FP}{FP+TP}$
- Taux de fausse découverte (FDR) = $\frac{FP}{FN+TP}$
- Taux de vraie découverte (TPR) = $\frac{TP}{TP+FN}$
- Fiabilité (*accuracy*) = $\frac{TP+FN}{TP+TN+FP+FN}$
- Score F1 = $\frac{2\times TP}{2\times TP+FP+FN}$

Il est à noté que les taux de complétude et de contamination proposés par CheckM sont respectivement des estimations du TPR et du FDR.

A.5 Données additonnelles

Les tables additionnelles de l'article en cours de révision sont disponibles à l'adresse suivante : https://keuv-grvl.github.io/thesis_data/chap_02/.

Posters et articles

Stratégies de reconstruction de génomes microbiens à partir de métagénomes

Kévin Gravouil [1,2,3], Corentin Hochart [2], Bérénice Batut [1], Clémence Defois [1], Cyrielle Gasc [1], Pierre Peyret [1], Didier Debroas [2], Marie Pailloux [3], Eric Peyretaillade [1]

[1] EA 4678 CIDAM ; [2] UMR CNRS 6023 LMGE ; [3] UMR CNRS 6158 LIMOS

Contacts : kevin.gravouil@udamail.fr ; didier.debroas@univ-bpclermont.fr ; pailloux@isima.fr ; eric.peyretaillade@udamail.fr

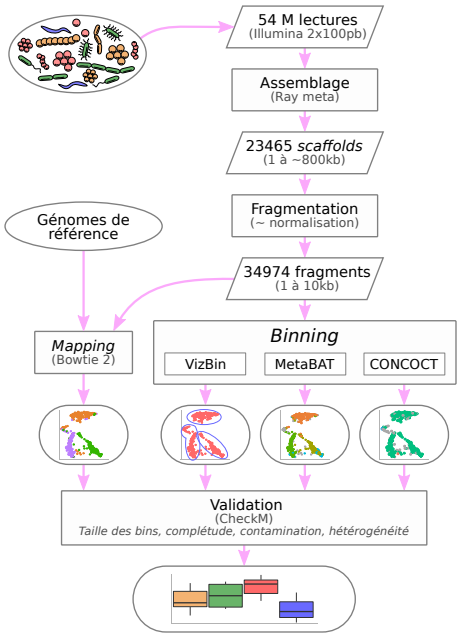
Introduction

Le séquençage à haut-débit permet d'accéder à l'extraordinaire diversité des micro-organismes notamment par des **approches métagénomiques**. Néanmoins, la compréhension globale d'un écosystème nécessite de relier efficacement **structure et fonctions**. De ce fait, la **reconstruction de génomes individuels** à partir de métagénomes devient une approche nécessaire pour remplir cet objectif.

La diversité génomique n'est pas connue *a priori* et dépend fortement de l'environnement étudié. Il convient donc d'employer des méthodes **non ciblées** et ***de novo*** pour explorer ces environnements.

Malgré la grande diversité de micro-organismes, le ***binning*** est une approche qui rend possible la reconstruction de génomes [1]. Cette approche repose sur le fait que deux **séquences similaires** en terme de composition appartiendraient à un **même génome**. Plusieurs stratégies ont depuis été proposées mais **aucun consensus** n'a pu être dégagé.

Matériel et méthodes



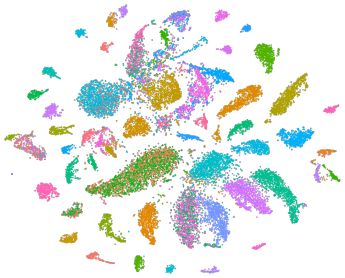
Afin d'évaluer la pertinence des méthodes de *binning* existantes, différents outils ont été testés sur un jeu de données **simulant un métagénome** composé de **64 micro-organismes** dont les génomes sont disponibles (SRR606249). Les bins de références sont obtenus par alignement (*mapping*) sur ces génomes.

Différents logiciels de *binning* ont été testés : (i) **VizBin** [2] qui exploite la composition nucléotidique ; (ii) **MetaBAT** [3] et (iii) **CONCOCT** [4] qui utilisent à la fois la composition et les différences de couverture des séquences. Ces trois approches diffèrent également par leurs méthodes de *clustering*. VizBin propose à l'utilisateur de définir les ***bins* manuellement** ; MetaBAT utilise un algorithme des **k-medoids** modifié ; CONCOCT utilise un **modèle de mélanges gaussiens** complétée d'une **approche bayesienne**.

La validation des *bins* avec **CheckM** [5] consiste à rechercher des **gènes-marqueurs uniques** au sein d'une lignée phylogénétique, évaluant ainsi : (i) la « **complétude** » (nombre de marqueurs au sein d'un *bin* par rapport à l'attendu) ; (ii) la **contamination** (nombre de marqueurs en plusieurs copies) et (iii) l'**hétérogénéité** de souche.

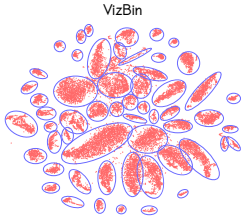
Etude comparative des outils de *binning*

• ***Binning*** - Les 34974 fragments de séquences (de 1 à 10kb) issus de l'assemblage *de novo* des 54 M de lectures ont été alignés aux génomes de référence ou regroupés par *binning*.

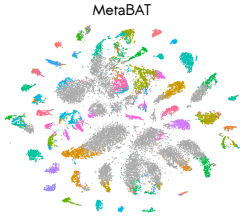


Référence
• *Mapping* sur les génomes de référence
» **33507 séquences** alignées (95.8%)
» **64 organismes** connus

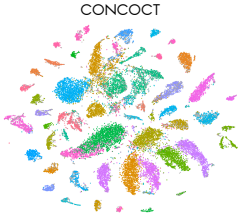
Légende
Chaque point représente un fragment de séquence.
Chaque couleur représente un *bin*.
Les points gris représentent les fragments non assignés.



VizBin
• Dénombrement des hexamers
• *Clustering* manuel
» **34559 séquences** *binnées* (98.8%)
» **55** ***bins***

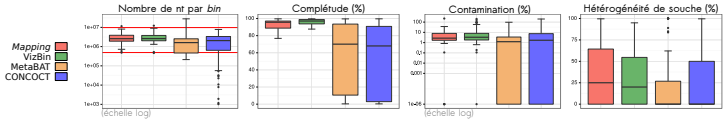


MetaBAT
• Dénombrement des tetramers + couverture
• *Clustering* par k-medoids
» **19111 séquences** *binnées* (54.6%)
» **84** ***bins***

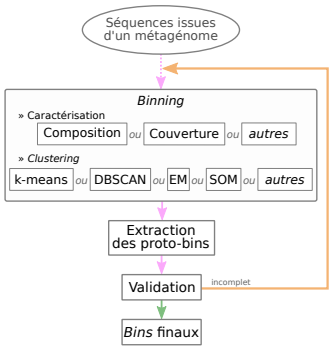


CONCOCT
• Dénombrement des tetramers + couverture
• *Clustering* par mélange gaussien
» **31817 séquences** *binnées* (87.1%)
» **68** ***bins***

• **Validation** - Des marqueurs sont recherchés pour chaque *bin* afin d'évaluer leur taux de complétude, de contamination et d'hétérogénéité de souche.



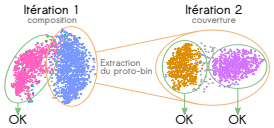
Stratégie alternative



Les méthodes de *binning* testées ne permettent pas toujours de dissocier correctement deux génomes d'espèces phylogénétiquement proches.

Pour pallier ce problème, il est possible de réaliser une **approche itérative**. Un premier *binning* permet d'**extraire des prototypes de *bins*** (ou « proto-bins »), et de les **valider individuellement**. Lorsque les critères de validation d'un proto-bin ne sont pas remplis (fig. 4, symbolisé par la flèche jaune), une autre méthode de binning est appliquée sur ce proto-bin.

• **Exemple** - Deux bins confondus deviennent séparables.



Conclusion et perspectives

» **Pas de consensus** en matière de reconstruction de génomes à partir de métagénomes
» Succession de **plusieurs méthodes** pour de meilleurs résultats
» Utilisation de **données de référence** pour la **validation** (si l'environnement étudié le permet)
» **Caractérisations alternatives** des séquences (par exemple, avec les *spaced-seed*)
» Utilisation de différentes méthodes de ***clustering***
» **Ré-analyse** des données existantes (*e.g.* : microbiote humain)
» Utilisation de méthodes issues du **Big Data**

Références

[1] Sangwan *et al.*, Microbiome, 2016, DOI: 10.1186/s40168-016-0154-5
[2] Laczny *et al.*, Microbiome, 2015, DOI: 10.1186/s40168-014-0066-1
[3] Kang *et al.*, PeerJ, 2015, DOI: 10.7717/peerj.1165
[4] Alneberg *et al.*, Nature Methods, 2014, DOI: 10.1038/nmeth.3103
[5] Parks *et al.*, Genome Research, 2014, DOI: 10.1101/gr.186072.114

frama.link/kg-j16

Reconstruction de génomes microbiens à partir de données métagénomiques

Kévin Gravouil [1,2,3], Didier Debroas [2], Marie Pailloux [3], Eric Peyretaillade [1]

[1] UMR 454 MEDIS ; [2] UMR 6023 LMGE ; [3] UMR 6158 LIMOS

Contacts : kevin.gravouil@uca.fr ; didier.debroas@uca.fr ; marie.pailloux@uca.fr ; eric.peyretaillade@uca.fr

Introduction

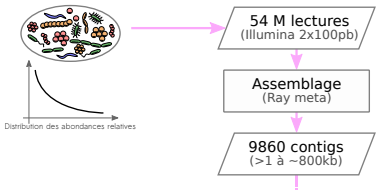
• Le séquençage à haut-débit permet d'accéder à l'extraordinaire diversité des micro-organismes notamment par des **approches métagénomiques**. Néanmoins, la compréhension globale d'un écosystème nécessite de relier efficacement **structure et fonctions**. De ce fait, la **reconstruction de génomes individuels** à partir de métagénomes devient une approche nécessaire pour remplir cet objectif.

• La diversité génomique n'est pas connue *a priori* et dépend fortement de l'environnement étudié. Il convient donc d'employer des méthodes **non ciblées** et ***de novo*** pour l'exploration de ces environnements.

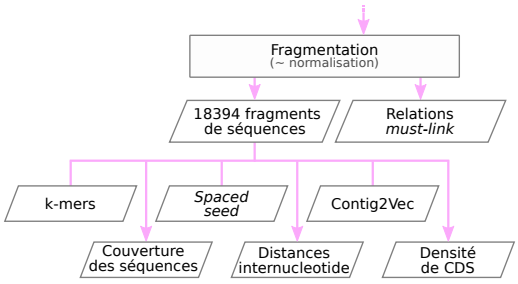
• Malgré la grande diversité de micro-organismes, le ***binning*** est une approche qui rend possible la reconstruction de génomes [1]. Cette approche repose sur le fait que deux séquences ayant des **caractéristiques similaires** appartiendraient à un **même génome**. Plusieurs caractérisations ont été proposées, essentiellement basées sur la composition en k-mers et la couverture des séquences, mais **aucun consensus** n'a pu être dégagé. Nous présentons ici une méthode adaptative d'intégration des différentes caractérisations de séquences connues en vue d'un *clustering*.

Matériel et méthodes

• **Données métagénomiques simulées** - Trois métagénomes de complexités variables et ayant une distribution des abondances relatives des génomes suivant une loi de puissance ont été simulés à partir de respectivement 64, 170 et 350 génomes de référence. Chaque métagénome a été assemblé afin de procéder au *binning* des contigs. L'origine de chaque contig étant connu, le résultat théorique du binning servira à l'évaluation de celui-ci. Le schéma suivant présente le protocole appliqué au premier jeu de données.



• **Modélisation des séquences** - Les 9860 contigs sont fragmentés pour limiter les effets liés à la variabilité des tailles de séquences, tout en conservant la provenance de chaque fragment via des relations *must-link*. Chaque fragment est ensuite caractérisé selon les six modèles proposés.



• **Intégration des modèles** - Les **composantes principales** de chacun de ces modèles bruts sont ensuite extraites à l'aide d'une ***kernel PCA*** [2] qui conserve 90% de l'information pour chaque modèle. L'utilisation du *kernel* **cosinus** permet de s'affranchir des étapes usuelles de normalisation.

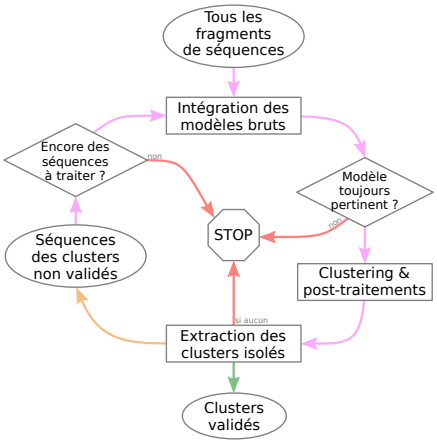
Toutes ces composantes principales sont centrées, réduites puis concaténées. Les doublons sont ensuite éliminés à l'aide d'une analyse en composante principale qui conserve 99,99% de l'information. On obtient ainsi une **représentation vectorielle** pour chaque fragment de séquence qui sera utilisé par l'algorithme de *clustering* [3].

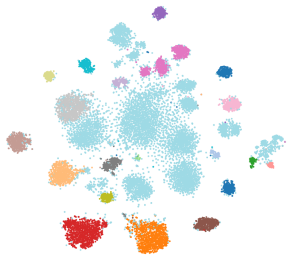
Références

[1] Sangwan *et al.*, Microbiome, 2016, DOI: 10.1186/s40168-016-0154-5
[2] Schölkopf *et al.*, Neural Computation, 1997, DOI: 10.1162/089976698300017467
[3] Alneberg et al., Nature Methods, 2014, DOI: 10.1038/nmeth.3103
[4] Rousseeuw, Comp. and Appl. Math., 1987; DOI: 10.1016/0377-0427(87)90125-7

Extraction itérative des *bins*

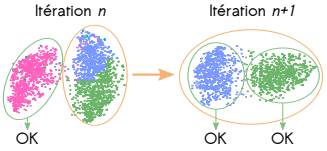
• Lors de la premières itération de l'algorithme, les modèles bruts des 18934 fragments de séquences sont intégrés en un seul modèle. S'il est jugé pertinent, les fragments de séquences sont clusterisés en s'appuyant sur ce modèle. L'intégration des modèles brutes telle qu'elle est proposée permet de trouver automatique les caractéristiques les plus discriminantes par rapport à la globablité du jeu de données et donc d'isoler certains clusters. Le post-traitement du *clustering* permet notamment de prendre en compte les relations *must-link* entre les fragments et de fusionner des clusters au besoin afin de préserver la cohérence avec les contigs originaux. Chaque cluster est ensuite évalué selon son isolement par rapport aux autres clusters à l'aide d'une analyse *silhouette* [4]. Ceux suffisament isolés sont validés. Les séquences non validées entre alors dans l'itération suivante.





• Après un itération, plusieurs clusters ont été extraits. Chaque point représente un fragment de séquence et sont colorés selon le cluster qui leur a été assignées. Les points bleu pâles entreront dans l'itéation suivante.

• L'intégration des modèles brutes se faisant par rapport à l'ensemble ou à un sous-ensemble de données, celle-ci sera différentes d'une itération à l'autre, conduisant à un clustering différent et donc à l'extraction de nouveaux clusters.



Intégration adaptative des modèles

• L'intégration des modèles bruts cherche les **attributs les plus discrimants** pour un ensemble de fragments donné. Celle-ci a été réalisé à partir des 373 attributs répartis entre 4 modèles : (i) couverture des séquence «*cov*» (1 attribut) ; (ii) Contig2Vec «*c2v*» (100 attributs) ; (iii) Dénombrement des tétramers «*4mer*» (136 attributs) et (iv) Spaced seed «*space*» en utilisant le masque «110011» (136 attributs).

• Lors de la première itération, la première composante principale (PC0) porte **24.6% de l'inertie** du modèle intégré et **corrèle très fortement** avec le modèle «Couverture des séquences» (corrélation linéaire de +0.99), mais aussi avec les modèles Contig2Vec et du dénombrement des tétramers avec des corrélations respectives de +0.16 et -0.13 pour les attributes 19 et 60.

• Les autres composantes principales sont alimentées par différents attributs des autres modèles. La couverture soit la principale source d'information pour la composante principale 0, l'intégration des modèles utilise différents attributs provenant de tous les modèles pour les autres composantes d'une itération à l'autre.

	PC0			PC1			PC2			PC3		
Itération 0	Inertie=24.6 %			Inertie=15.1 %			Inertie=14.4 %			Inertie=14.3 %		
	Modèle	*Attribut*	*Correl.*	*M*	*A*	*C*	*M*	*A*	*C*	*M*	*A*	*C*
nb seq=18934	cov	0	+0.99	c2v	2	-0.96	c2v	19	+0.67	c2v	56	-0.54
nb PC=15	c2v	19	+0.16	c2v	69	-0.96	c2v	85	-0.62	c2v	33	-0.51
	4mer	60	-0.13	c2v	79	+0.96	4mer	80	+0.62	c2v	18	+0.51
Itération 1	Inertie=22.5 %			Inertie=13.7 %			Inertie=13.5 %			Inertie=13.0 %		
	M	*A*	*C*	*M*	*A*	*C*	*M*	*A*	*C*	*M*	*A*	*C*
nb seq=10923	cov	0	+0.99	c2v	28	-0.49	c2v	74	-0.96	kmer4	65	+0.56
nb PC=17	4mer	131	-0.16	c2v	65	+0.45	c2v	97	-0.96	space	14	+0.55
	4mer	80	-0.16	c2v	23	+0.41	c2v	21	-0.95	space	32	+0.53
Itération 2	Inertie=10.4 %			Inertie=6.3 %			Inertie=6.0 %			Inertie=5.9 %		
	M	*A*	*C*	*M*	*A*	*C*	*M*	*A*	*C*	*M*	*A*	*C*
nb seq=8182	cov	0	+0.98	c2v	86	+0.74	c2v	47	-0.68	c2v	57	+0.81
nb PC=43	c2v	47	+0.19	c2v	54	+0.74	c2v	2	-0.57	c2v	26	-0.78
	c2v	60	+0.17	c2v	60	+0.73	c2v	66	-0.54	c2v	82	-0.77

Tableau: Contribution des différents modèles bruts à la construction des quatres premières composantesprincipales du modèle intéré lors des trois premières itérations

Conclusion et perspectives

» **Pas de consensus** en matière de reconstruction de génomes à partir de métagénomes
» Intégration de **multiples caractérisations** de séquences
» Sélection automatique des **attributs les plus discriminants** pour un ensemble de séquences donné
» **Extraction** des clusters les mieux discriminés

» **Ré-analyse** des données existantes (e.g. : microbiote intestinal humain)
» Proposition de **nouveaux modèles vectoriels** de données

GigaScience, 7, 2018, 1–7

doi: 10.1093/gigascience/giy057
Advance Access Publication Date: 15 May 2018
Technical Note

TECHNICAL NOTE

ASaiM: a Galaxy-based framework to analyze microbiota data

Bérénice Batut[1,2,*], Kévin Gravouil[1,3,4,5], Clémence Defois[1,3], Saskia Hiltemann[6], Jean-François Brugère[1], Eric Peyretaillade[1,4] and Pierre Peyret[1,3,*]

[1]Université Clermont Auvergne, EA 4678 CIDAM, 63000 Clermont-Ferrand, France (previous address), [2]Bioinformatics Group, Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany, [3]Université Clermont Auvergne, INRA, MEDIS, 63000 Clermont-Ferrand, France, [4]Université Clermont Auvergne, CNRS, LMGE, 63000 Clermont–Ferrand, France, [5]Université Clermont Auvergne, CNRS, LIMOS, 63000 Clermont–Ferrand, France and [6]Department of Bioinformatics, Erasmus University Medical Center, Rotterdam, 3015 CE, Netherlands

*Correspondence address. Université Clermont Auvergne, EA 4678 CIDAM, 63000 Clermont-Ferrand, France - Bérénice Batut. E-mail: berenice.batut@gmail.com http://orcid.org/0000-0001-9852-1987 and Pierre Peyret. E-mail: pierre.peyret@uca.fr http://orcid.org/0000-0003-3114-0586

Abstract

Background: New generations of sequencing platforms coupled to numerous bioinformatics tools have led to rapid technological progress in metagenomics and metatranscriptomics to investigate complex microorganism communities. Nevertheless, a combination of different bioinformatic tools remains necessary to draw conclusions out of microbiota studies. Modular and user-friendly tools would greatly improve such studies. **Findings:** We therefore developed ASaiM, an Open-Source Galaxy-based framework dedicated to microbiota data analyses. ASaiM provides an extensive collection of tools to assemble, extract, explore, and visualize microbiota information from raw metataxonomic, metagenomic, or metatranscriptomic sequences. To guide the analyses, several customizable workflows are included and are supported by tutorials and Galaxy interactive tours, which guide users through the analyses step by step. ASaiM is implemented as a Galaxy Docker flavour. It is scalable to thousands of datasets but also can be used on a normal PC. The associated source code is available under Apache 2 license at https://github.com/ASaiM/framework and documentation can be found online (http://asaim.readthedocs.io). **Conclusions:** Based on the Galaxy framework, ASaiM offers a sophisticated environment with a variety of tools, workflows, documentation, and training to scientists working on complex microorganism communities. It makes analysis and exploration analyses of microbiota data easy, quick, transparent, reproducible, and shareable.

Keywords: metagenomics; metataxonomics; user-friendly; Galaxy; Docker; microbiota; training

Findings

Background

The study of microbiota and microbial communities has been facilitated by the evolution of sequencing techniques and the development of metataxonomics, metagenomics, and metatranscriptomics. These techniques are giving insight into taxonomic profiles and genomic components of microbial communities. However, meta'omic data exploitation is not trivial due to the large amount of data, their complexity, the incompleteness of reference databases, and the difficulty to find, configure, use, and combine the dedicated bioinformatics tools, etc. Hence, to

Received: 8 September 2017; **Revised:** 6 January 2018; **Accepted:** 10 May 2018

extract useful information, a sequenced microbiota sample has to be processed by sophisticated workflows with numerous successive bioinformatics steps [1]. Each step may require execution of several tools or software. For example, to extract taxonomic information with the widely used QIIME [2] or Mothur [3], at least 10 different tools with at least four parameters each are needed. Designed for amplicon data, both QIIME and Mothur cannot be directly applied to shotgun metagenomics data. In addition, the tools can be complex to use; they are command-line tools and may require extensive computational resources (memory, disk space). In this context, selecting the best tools, configuring them to use the correct parameters and appropriate computational resources, and combining them together in an analysis chain is a complex and error-prone process. These issues and the involved complexity are prohibiting scientists from participating in the analysis of their own data. Furthermore, bioinformatics tools are often manually executed and/or patched together with custom scripts. These practices raise doubts about a science gold standard: reproducibility [3, 4]. Web services and automated pipelines such as MG-RAST [5] and EBI metagenomics [6] offer solutions to the accessibility issue. However, these web services work as a black box and are lacking in transparency, flexibility, and even reproducibility as the version and parameters of the tools are not always available. Alternative approaches to improve accessibility, modularity, and reproducibility can be found in open-source workflow systems such as Galaxy [6-8]. Galaxy is a lightweight environment providing a web-based, intuitive, and accessible user interface to command-line tools, while automatically managing computation and transparently managing data provenance and workflow scheduling [6-8]. More than 5,500 tools can be used inside any Galaxy environment. For example, the main Galaxy server [9] integrates many genomic tools, and the few integrated metagenomics tools such as Kraken [10] or VSearch [11] have been showcased in the published windshield splatter analysis [12]. The tools can also be selected and combined to build Galaxy flavors focusing on specific type of analysis, for example, the Galaxy RNA workbench [13] or the specialized Galaxy server of the Huttenhower lab [14]. However, none of these solutions is dedicated to microbiota data analysis in general and with the community-standard tools.

In this context, we developed ASaiM (Auvergne Sequence analysis of intestinal Microbiota, RRID:SCR_015878), an Open-Source opinionated Galaxy-based framework. It integrates more than 100 tools and several workflows dedicated to microbiota analyses with an extensive documentation [15] and training support.

Goals of ASaiM

ASaiM is developed as a modular, accessible, redistributable, sharable, and user-friendly framework for scientists working with microbiota data. This framework is unique in combining curated tools and workflows and providing easy access and support for scientists.

ASaiM is based on four pillars: (1) easy and stable dissemination via Galaxy, Docker, and Conda, (2) a comprehensive set of microbiota-related tools, (3) a set of predefined and tested workflows, and (4) extensive documentation and training to help scientists in their analyses.

A framework built on the shoulders of giants

The ASaiM framework is built on existing tools and infrastructures and combines all their forces to create an easily accessible and reproducible analysis platform.

ASaiM is implemented as a portable virtualized container based on the Galaxy framework [8]. Galaxy provides researchers with means to reproduce their own workflows analyses, rerun entire pipelines, or publish and share them with others. Based on Galaxy, ASaiM is scalable from single CPU installations to large multi-node high performance computing environments and manages efficiently job submission as well as memory consumption of the tools. Deployments can be achieved by using a pre-built ASaiM Docker image, which is based on the Galaxy Docker project [16]. This ASaiM Docker flavour is customized with a variety of selected tools, workflows, interactive tours, and data that have been added as additional layers on top of the generic Galaxy Docker instance. The containerization keeps the deployment task to a minimum. The selected Galaxy tools are automatically installed from the Galaxy ToolShed [17] using the Galaxy API BioBlend [18], and the installation of the tools and their dependencies are automatically resolved using packages available through Bioconda [19]. To populate ASaiM with the selected microbiota tools, we migrated the 12 tools/suites of tools and their dependencies to Bioconda (e.g., HUMAnN2), integrated 16 suites (>100 tools) into Galaxy (e.g., HUMAn2 or QIIME with its approximately 40 tools), and updated the already available ones (Table 1).

Tools for microbiota data analyses

The tools integrated in ASaiM can be seen in Table 1. They are expertly selected for their relevance with regard to microbiota studies, such as Mothur (mothur, RRID:SCR_011947) [3], QIIME (QIIME, RRID:SCR_008249) [2], MetaPhlAn2 (MetaPhlAn, RRID:SCR_004915) [45], HUMAnN2 [46], or tools used in existing pipelines such as EBI Metagenomics' one. We also added general tools used in sequence analysis such as quality control, mapping, or similarity search tools.

An effort in development was made to integrate these tools into Conda and the Galaxy environment (>100 tools integrated) with the help and support of the Galaxy community. We also developed two new tools to search and get data from EBI Metagenomics and ENA databases (EBISearch [20] and ENASearch [21]) and a tool to group HUMAnN2 outputs into Gene Ontology Slim Terms [47]. Tools inside ASaiM are documented [15] and organized to make them findable.

Diverse source of data

An easy way to upload user-data into ASaiM is provided by a web interface or more sophisticatedly via FTP or SFTP. On the top, we added specialised tools that can interact with external databases like NCBI, ENA, or EBI Metagenomics to query them and download data into the ASaiM environment.

Visualization of the data

An analysis often ends with summarizing figures that conclude and represent the findings. ASaiM includes standard interactive plotting tools to draw bar charts and scatter plots for all kinds of tabular data. Phinch visualization [52] is also included to interactively visualize and explore any BIOM file and generate different types of ready-to-publish figures. We also integrated two

Table 1: Available tools in ASaiM

Section	Subsection	Tools
File and meta tools	Data retrieval	EBISearch [20], ENASearch [21], SRA Tools
	Text manipulation	Tools from Galaxy ToolShed
	Sequence file manipulation	Tools from Galaxy ToolShed
	BAM/SAM file manipulation	SAM tools [22-24]
	BIOM file manipulation	BIOM-Format tools [25]
Genomics tools	Quality control	FastQC [26], PRINSEQ [27], Trim Galore! [28, Trimmomatic [29], MultiQC [30]
	Clustering	CD-Hit [31], Format CD-HIT outputs
	Sorting and prediction	SortMeRNA [32], FragGeneScan [33]
	Mapping	BWA [34], Bowtie [35]
	Similarity search	NCBI Blast+ [36, 37], Diamond [38]
	Alignment	HMMER3 [39]
Microbiota dedicated tools	Metagenomics data manipulation	VSEARCH [11], Nonpareil [40]
	Assembly	MEGAHIT [41], metaSPAdes [42], metaQUAST [43], VALET [44]
	Metataxonomic sequence analysis	Mothur [3], QIIME [2]
	Taxonomy assignation on WGS sequences	MetaPhlAn2 [45], Format MetaPhlan2, Kraken [10]
	Metabolism assignation	HUMAnN2 [46], Group HUMAnN2 to GO slim terms [47], Compare HUMAnN2 outputs, PICRUST [48], InterProScan
	Combination of functional and taxonomic results	Combine MetaPhlAn2 and HUMAnN2 outputs
	Visualization	Export2graphlan [49], GraPhlAn [50], KRONA [51]

This table presents the tools, organized in sections and subsections to help users. A more detailed table of the available tools and some documentation can be found in the online documentation (http://asaim.readthedocs.io/en/latest/tools/).

other tools to explore and represent the community structure: KRONA [51] and GraPhlAn [53]. Moreover, as in any Galaxy instance, other visualizations are included such as Phyloviz [54] for phylogenetic trees or the genome browser Trackster [55] for visualizing SAM/BAM, BED, GFF/GTF, WIG, bigWig, bigBed, bedGraph, and VCF datasets.

Workflows

Each tool can be used separately in an explorative manner, the Galaxy tool form helping users in setting meaningful parameters. Tools can be also orchestrated inside workflows using the powerful Galaxy workflow manager. To assist in microbiota analyses, several workflows, including a few well-known pipelines, are offered and documented (tools and their default parameters) in ASaiM. These workflows can be used as is; customized either on the fly to tune the parameters or globally to change the tools, their order, and their default parameters; or even used as subworkflows. Moreover, users can also design novel meaningful workflows via the Galaxy workflow interface using the >100 available tools.

Analysis of raw metagenomic or metatranscriptomic shotgun data

The workflow quickly produces, from raw metagenomic or metatranscriptomic shotgun data, accurate and precise taxonomic assignations, wide extended functional results, and taxonomically related metabolism information (Fig. 1). This workflow consists of (i) processing with quality control/trimming (FastQC and Trim Galore!) and dereplication (VSearch [11]); (ii) taxonomic analyses with assignation (MetaPhlAn2 [45]) and visualization (KRONA, GraPhlAn); (iii) functional analyses with metabolic assignation and pathway reconstruction (HUMAnN2 [46]); (iv) functional and taxonomic combination with developed tools combining HUMAnN2 and MetaPhlAn2 outputs.

This workflow has been tested on two mock metagenomic datasets with controlled communities (Supplementary material). We have compared the extracted taxonomic and functional information to such information extracted with the EBI metagenomics' pipeline and to the expectations from the mock datasets to illustrate the potential of the ASaiM workflow. With ASaiM, we generate accurate and precise data for taxonomic analyses (Fig. 2), and we can access information at the level of the species. More functional information (e.g., gene families, gene ontologies, pathways) are also extracted with ASaiM compared to the ones available on EBI metagenomics. With this workflow, we can go one step further and investigate which taxons are involved in a specific pathway or a gene family (e.g., involved species and their relative involvement in different step of fatty acid biosynthesis pathways, Fig. 3).

For the tests, ASaiM was deployed on a computer with Debian GNU/Linux System, 8 cores Intel(R) Xeon(R) at 2.40 GHz and 32 Go of RAM. The workflow processed the 1,225,169 and 1,386,198 454 GS FLX Titanium reads of each datasets, with a stable memory usage, in 4h44 and 5h22 respectively (Supplementary material). The execution time is logarithmically linked to the input data size. With this workflow, it is then easy and quick to process raw microbiota data and extract diverse useful information.

Assembly of metagenomics data

Microbiota data usually come with quite short reads. To reconstruct genomes or to get longer sequences for further analysis, microbiota sequences have to be assembled with dedicated metagenome assemblers. To help in this task, two workflows

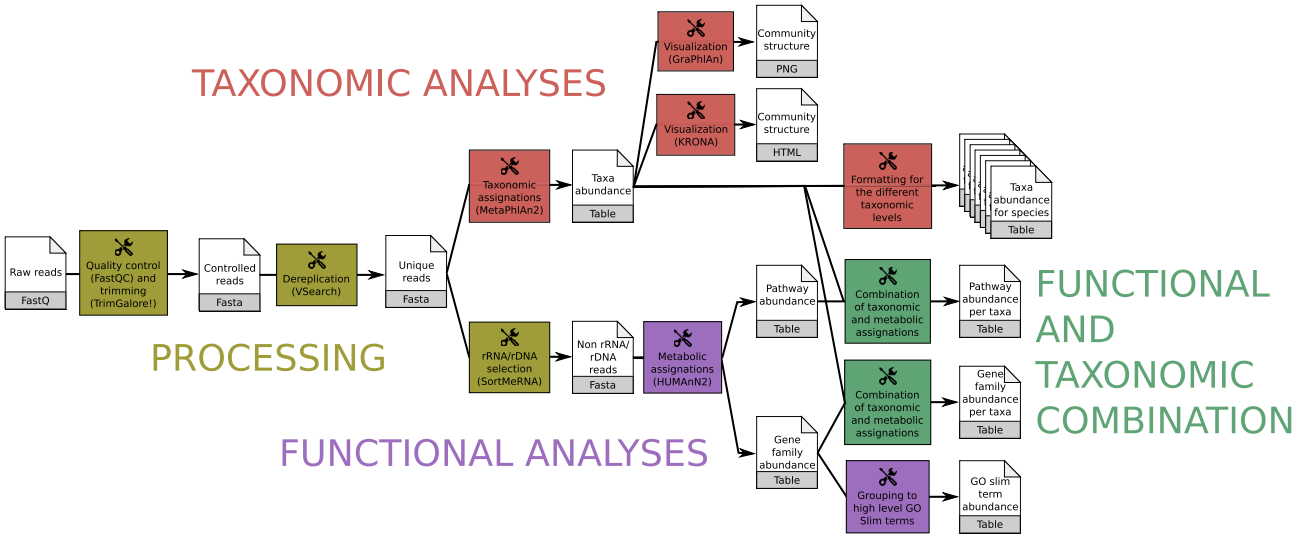


Figure 1: Main ASaiM workflow to analyze raw sequences. This workflow takes as input a dataset of raw shotgun sequences (in FastQ format) from microbiota, preprocess it (yellow boxes), extracts taxonomic (red boxes) and functional (purple boxes) assignations, and combines them (green boxes). Image available under CC-BY license (https://doi.org/10.6084/m9.figshare.5371396.v3).

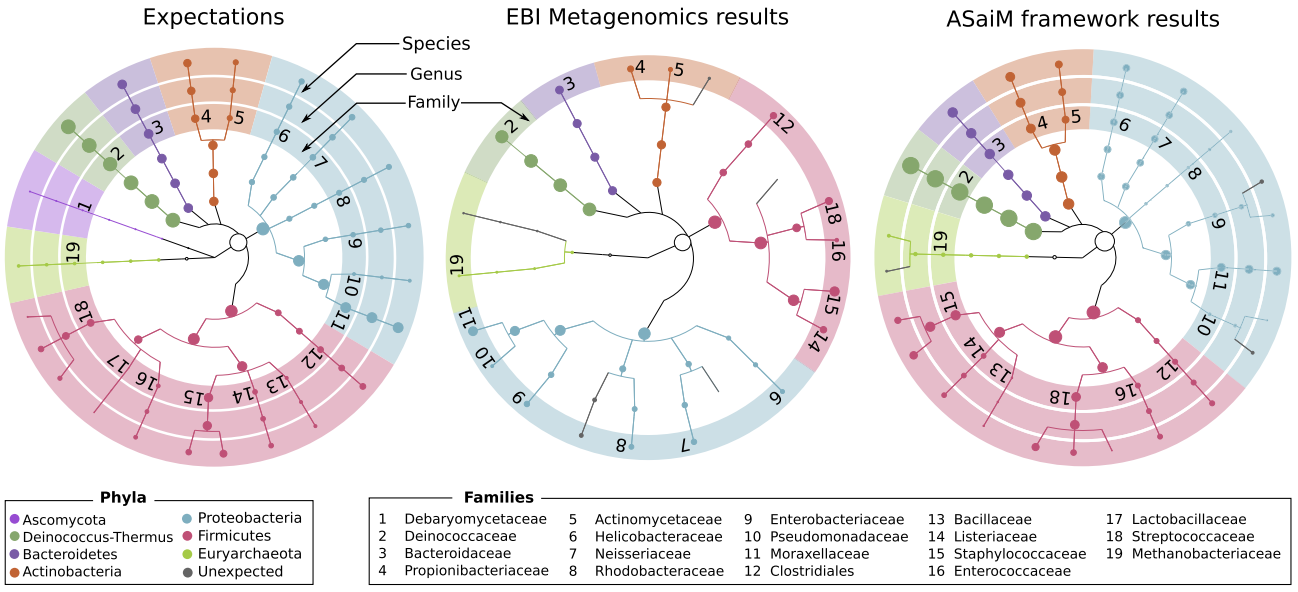


Figure 2: Comparisons of the community structure for SRR072233. This figure compares the community structure between the expectations (mapping of the sequences on the expected genomes), data found on EBI Metagenomics database (extracted with the EBI Metagenomics pipeline), and the results of the main ASaiM workflow (Fig. 1).

have been developed in ASaiM, each one using one of the well-performing assemblers [56-62]: MEGAHIT [41] and MetaSPAdes [42]. Both workflows consists of: (1) processing with quality control/trimming (FastQC and Trim Galore!); (2) assembly with either MEGAHIT or MetaSPAdes; (3) estimation of the assembly quality statistics with MetaQUAST [43]; (4) identification of potential assembly error signature with VALET; and (5) determination of percentage of unmapped reads with Bowtie2 (Bowtie, RRID:SCR_005476) [36] combined with MultiQC [30] to aggregate the results.

Analysis of metataxonomic data

To analyze amplicon or internal transcribed spacer data, the Mothur and QIIME tool suites are available in ASaiM. We integrated the workflows described in tutorials of Mothur and QIIME as an example of metataxonomic data analyses as well as support for the training material.

Running as in EBI Metagenomics

As the tools used in the EBI Metagenomics pipeline (version 3) are also available in ASaiM, we integrate them in a workflow with the same steps as the EBI Metagenomics pipeline. Analyses made in the EBI Metagenomics website can be then re-

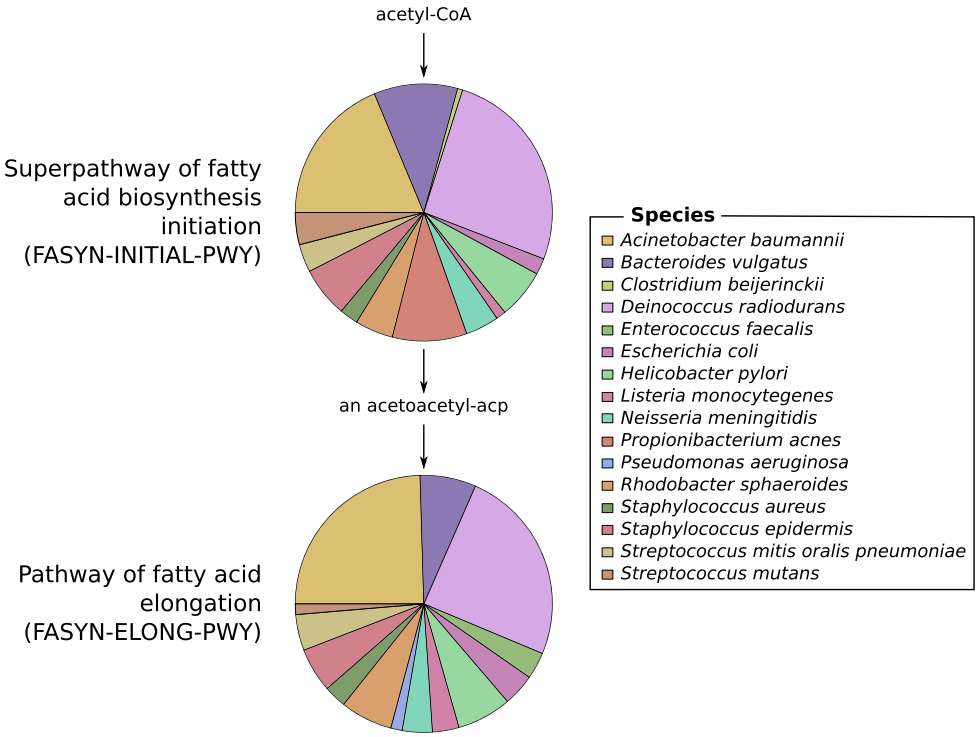


Figure 3: Example of an investigation of the relation between community structure and functions. The involved species and their relative involvement in fatty acid biosynthesis pathways have been extracted with ASaiM workflow (Fig. 1) for SRR072233.

produced locally without having to wait for availability of EBI Metagenomics or to upload any data on EBI Metagenomics. However, the parameters must be defined by the user, as we cannot find them on EBI Metagenomics documentation. In ASaiM, the entire provenance and every parameter are tracked to guarantee the reproducibility.

Documentation and training

A tool or software is easier to use if it is well documented. Hence, extensive documentation helps the users to be familiar with the tool and also prevents mis-usage. For ASaiM, we developed an extensive online documentation [15] , mainly to explain how to use it, how to deploy it, which tools are integrated with small documentation about these tools, which workflows are available, and how to use them.

In addition to this online documentation, training materials have been developed. Some Galaxy interactive tours are included inside the Galaxy instance to guide users through entire microbiota analyses in an interactive (step-by-step) way. We also developed several step-by-step tutorials to explain the concepts of microbiota analyses, the different tools and parameters, and ASaiM workflows with toy datasets. Hosted within the Galaxy Training Material [63], the tutorials are available online at [64] and also directly accessible from ASaiM and its documentation for self-training. These tutorials and ASaiM have been used during several workshops on metagenomics data analysis and some undergraduate courses to explain and use the EBI Metagenomics workflow in a reproducible way. ASaiM is also used as support for a citizen science and education project (BeerDeCoded [65]).

Installation and running ASaiM

Running the containerized ASaiM simply requires the user to install Docker and to start the ASaiM image with:

```
$ docker run -d -p 8080:80 quay.io/bebatut/asaim-framework:latest
```

As Galaxy, ASaiM is production ready and can be configured to use external accessible computer clusters or cloud environments. It is also possible and easy to install all or only a subset of tools of the ASaiM framework on existing Galaxy instances, as we did on the European Galaxy instance [66]. More details about the installation and the use of ASaiM are available on the online documentation [15].

Conclusion

ASaiM provides a powerful framework to easily and quickly analyze microbiota data in a reproducible, accessible, and transparent way. Built on a Galaxy instance wrapped in a Docker image, ASaiM can be easily deployed with its extensive set of tools and their dependencies, saving users from the hassle of installing all software. These tools are complemented with a set of predefined and tested workflows to address the main questions of microbiota research (assembly, community structure, and function). All these tools and workflows are extensively documented online [15] and supported by interactive tours and tutorials.

With this complete infrastructure, ASaiM offers a sophisticated environment for microbiota analyses to any scientist while promoting transparency, sharing, and reproducibility.

Methods

For the tests, ASaiM was deployed on a computer with Debian GNU/Linux System, 8 cores Intel(R) Xeon(R) at 2.40 GHz and 32 Go of RAM. The workflow has been run on two mock community samples of the Human Microbiome Project containing a genomic mixture of 22 known microbial strains. The details of comparison analyses are described in the Supplementary Material.

Availability of supporting data

Archival copies of the code and mock data are available in the *GigaScience* GigaDB repository [67].

Availability of supporting source code and requirements

- Project name: ASaiM
- Project home page: https://github.com/ASaiM/framework
- Operating system(s): Platform independent
- Other requirements: Docker
- License: Apache 2
- RRID:SCR_015878GTN

All tools described herein are available in the Galaxy Toolshed (https://toolshed.g2.bx.psu.edu). The Dockerfile to automatically deploy ASaiM is provided in the GitHub repository (https://github.com/ASaiM/framework) and a pre-built Docker image is available at https://quay.io/repository/bebatut/asaim-framework.

Additional files

sup_mat_1.pdf

Abbreviations

API: application programming interface; AsaiM: Auvergne Sequence analysis of intestinal Microbiota; CPU: central processing unit; Galaxy Training Network.

Competing interests

The author(s) declare that they have no competing interests.

Funding

The Auvergne Regional Council and the European Regional Development Fund supported this work.

Authors' contributions

B.B., K.G., C.D., S.H., J.F.B., E.P., and P.P. contributed equally to the conceptualization, methodology, and writing process; J.F.B. and P.P. contributed equally to the funding acquisition; B.B., K.G., and S.H. contributed equally to the software development; and B.B., K.G., C.D., and J.F.B. contributed equally to the validation.

Acknowledgements

The authors would like to thank EA 4678 CIDAM, UR 454 INRA, M2iSH, LIMOS, AuBi, Mésocentre, and de.NBI for their involvement in this project, as well as Réjane Beugnot, Thomas Eymard, David Parsons, and Björn Grüning for their help.

References

1. Ladoukakis E, Kolisis FN, Chatziioannou AA. Integrative workflows for metagenomic analysis. Front Cell Dev Biol 2014;**2**:70.
2. Caporaso JG, Kuczynski J, Stombaugh J, et al., QIIME allows analysis of high-throughput community sequencing data. Nature methods 2010, **7**, 5, 335–336:.
3. Schloss PD, Westcott SL, Ryabin T, et al. Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. Appl Environ Microbiol 2009;**75**:7537–41.
4. Nekrutenko A, Taylor J. Next-generation sequencing data interpretation: enhancing reproducibility and accessibility. Nat Rev Genet 2012;**13**:667–72.
5. Meyer F, Paarmann D, D'Souza M, et al. The metagenomics RAST server – a public resource for the automatic phylogenetic and functional analysis of metagenomes. BMC Bioinformatics 2008;**9**:386.
6. Hunter S, Corbett M, Denise H, et al. EBI metagenomics–a new resource for the analysis and archiving of metagenomic data. Nucleic Acids Res 2014;**42**:D600–6.
7. Goecks J, Nekrutenko A, Taylor J, et al. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol 2010;**11**:R86.
8. Afgan E, Baker D, van den Beek M, et al. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. Nucleic Acids Res 2016;**44**:W3–10.
9. Main Galaxy instance, http://usegalaxy.org
10. Wood DE, Salzberg SL. Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biol 2014;**15**:R46.
11. Rognes T, Flouri T, Nichols B, et al. VSEARCH: a versatile open source tool for metagenomics. PeerJ 2016;**4**:e2584.
12. Kosakovsky Pond S, Wadhawan S, Chiaromonte F, et al. Windshield splatter analysis with the Galaxy metagenomic pipeline. Genome Res 2009;**19**:2144–53.
13. Grüning BA, Fallmann J, Yusuf D, et al. The RNA workbench: best practices for RNA and high-throughput sequencing bioinformatics in Galaxy. Nucleic Acids Res 2017;**45**:W560–6.
14. Galaxy instance of the Huttenhower Lab, http://huttenhower.sph.harvard.edu/galaxy
15. ASaiM Documentation, http://asaim.readthedocs.io
16. Docker images tracking the stable Galaxy releases, http://bgruening.github.io/docker-galaxy-stable
17. Blankenberg D, Von Kuster G, Bouvier E, et al. Dissemination of scientific software with Galaxy ToolShed. Genome Biol 2014;**15**:403.
18. Sloggett C, Goonasekera N, Afgan E. BioBlend: automating pipeline analyses within Galaxy and CloudMan. Bioinformatics 2013;**29**:1685–6.
19. Grüning B, , Dale R, Sjödin A, et al. Bioconda: A sustainable and comprehensive software distribution for the life sciences. bioRxiv 2017. http://dx.doi.org/10.1101/207092.
20. EBISearch,http://github.com/bebatut/ebisearch
21. Batut B, Grüning B. ENASearch: A Python library for interacting with ENA's API. The Journal of Open Source Software 2017;**2**:418.
22. Li H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. Bioinformatics 2011;**27**:2987–93.

23. Li H. Improving SNP discovery by base alignment quality. Bioinformatics 2011;**27**:1157–8.
24. Li H, Handsaker B, Wysoker A, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics 2009;**25**:2078–9.
25. McDonald D, Clemente JC, Kuczynski J, et al. The Biological Observation Matrix (BIOM) format or: how I learned to stop worrying and love the ome-ome. Gigascience 2012;**1**:7.
26. FastQC, https://www.bioinformatics.babraham.ac.uk/projects/fastqc.
27. Schmieder R, Edwards R. Quality control and preprocessing of metagenomic datasets. Bioinformatics 2011;**27**:863–4.
28. Trim Galore!, https://www.bioinformatics.babraham.ac.uk/projects/trim_galore.
29. Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics 2014;**30**:2114–20.
30. Ewels P, Magnusson M, Lundin S, et al. MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics 2016;**32**:3047–8.
31. Fu L, Niu B, Zhu Z, et al. CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics 2012;**28**:3150–2.
32. Kopylova E, Noé L, Touzet H. SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. Bioinformatics 2012;**28**:3211–7.
33. Rho M, Tang H, Ye Y. FragGeneScan: predicting genes in short and error-prone reads. Nucleic Acids Res 2010;**38**:e191–.
34. Li H, Durbin R. Fast and accurate long-read alignment with Burrows-Wheeler transform. Bioinformatics 2010;**26**:589–95.
35. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nature Methods 2012, **9**, 357–359.
36. Camacho C, Coulouris G, Avagyan V, et al. BLAST+: architecture and applications. BMC Bioinformatics 2009;**10**:421.
37. Cock PJA, Chilton JM, Grüning B, et al. NCBI BLAST+ integrated into Galaxy. Gigascience 2015;**4**:39.
38. Buchfink B, Xie C, Huson DH. Fast and sensitive protein alignment using DIAMOND. Nat Methods 2015;**12**:59–60.
39. Mistry J, Finn RD, Eddy SR, et al. Challenges in homology search: HMMER3 and convergent evolution of coiled-coil regions. Nucleic Acids Res 2013;**41**(12):e121.
40. Rodriguez-R LM, Konstantinidis KT. Nonpareil: a redundancy-based approach to assess the level of coverage in metagenomic datasets. Bioinformatics 2014;**30**:629–35.
41. Li D, Luo R, Liu C-M, et al. MEGAHIT v1.0: A fast and scalable metagenome assembler driven by advanced methodologies and community practices. Methods 2016;**102**:3–11.
42. Nurk S, Meleshko D, Korobeynikov A, et al. metaSPAdes: a new versatile metagenomic assembler. Genome Res 2017;**27**:824–34.
43. Mikheenko A, Saveliev V, Gurevich A. MetaQUAST: evaluation of metagenome assemblies. Bioinformatics 2016;**32**:1088–90.
44. VALET, http://github.com/jgluck/valet.
45. Truong DT, Franzosa EA, Tickle TL, et al. MetaPhlAn2 for enhanced metagenomic taxonomic profiling. Nat Methods 2015;**12**:902–3.
46. Abubucker S, Segata N, Goll J, et al. Metabolic reconstruction for metagenomic data and its application to the human microbiome. PLoS Comput Biol 2012;**8**:e1002358.
47. Group HUMAnN2 to GO slim terms, https://github.com/asaim/group_humann2_uniref_abundances_to_GO.
48. Langille MGI, Zaneveld J, Caporaso JG, et al. Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences. Nat Biotechnol 2013;**31**:814–21.
49. export2graphlan, http://bitbucket.org/CibioCM/export2graphlan.
50. Asnicar F, Weingart G, Tickle TL, et al. Compact graphical representation of phylogenetic data and metadata with GraPhlAn. PeerJ 2015;**3**:e1029.
51. Ondov BD, Bergman NH, Phillippy AM. Interactive metagenomic visualization in a Web browser. BMC Bioinformatics 2011;**12**:385.
52. Bik HM, Phinch: an interactive, exploratory data visualization framework for -Omics datasets. bioRxiv 2014 . http://dx.doi.org/10.1101/009944.
53. GraPhlAn, http://huttenhower.sph.harvard.edu/graphlan.
54. Nascimento M, Sousa A, Ramirez M, et al., PHYLOViZ 2.0: providing scalable data integration and visualization for multiple phylogenetic inference methods. Bioinformatics 2017;**33**(1):128–129.
55. Goecks J, Coraor N, Galaxy Team, NGS analyses by visualization with Trackster. Nat Biotechnol 2012;**30**(11):1036–9.
56. Awad S, Irber L, Titus Brown C, . Evaluating metagenome assembly on a simple defined community with many strain variants , bioRxiv. 2017. http://dx.doi.org/10.1101/155358. .
57. Greenwald WW, Klitgord N, Seguritan V, et al. Utilization of defined microbial communities enables effective evaluation of meta-genomic assemblies. BMC Genomics 2017;**18**:296.
58. Olson ND, Treangen TJ, Hill CM, et al. Metagenomic assembly through the lens of validation: recent advances in assessing and improving the quality of genomes assembled from metagenomes. Brief Bioinform 2017, **bbx098**; http://dx.doi.org/10.1093/bib/bbx098.
59. Quince C, Walker AW, Simpson JT, et al. Shotgun metagenomics, from sampling to analysis. Nat Biotechnol 2017;**35**:833–44.
60. Sczyrba A, Hofmann P, Belmann P, et al. Critical Assessment of Metagenome Interpretation-a benchmark of metagenomics software. Nat Methods 2017;**14**:1063–71.
61. van der Walt AJ, Van Goethem MW, Ramond J-B, et al. Assembling Metagenomes, One Community At A Time, BMC Genomics. 2017, **18**:521.
62. Vollmers J, Wiegand S, Kaster A-K. Comparing and evaluating metagenome assembly tools from a microbiologist's perspective - not only size matters!. PLoS One 2017;**12**:e0169662.
63. Batut B, Hiltemann S, Bagnacani A, et al., Community-driven data analysis training for biology, bioRxiv, 2017, http://dx.doi.org/10.1101/225680
64. Galaxy Training Material for metagenomics, http://training.galaxyproject.org/topics/metagenomics
65. Sobel J, Henry L, Rotman N, et al. BeerDeCoded: the open beer metagenome project. F1000Res 2017;**6**:1676.
66. Metagenomics flavor of the European Galaxy instance, https://metagenomics.usegalaxy.eu
67. Batut B, Gravouil K, Defois C, et al. Supporting data for "ASaiM: a Galaxy-based framework to analyze microbiota data" GigaScience Database 2018 http://dx.doi.org/10.5524/100451.

Bioconda: A sustainable and comprehensive software distribution for the life sciences

Björn Grüning[1], Ryan Dale[*,2], Andreas Sjödin[3,4], Brad A. Chapman[5], Jillian Rowe[6], Christopher H. Tomkins-Tinch[7,8], Renan Valieris[9], Adam Caprez[10], Bérénice Batut[1], Mathias Haudgaard[11], Thomas Cokelaer[12], Kyle A. Beauchamp[13], Brent S Pedersen[14], Youri Hoogstrate[15], Anthony Bretaudeau[16], Devon Ryan[17], Gildas Le Corguillé[18], Dilmurat Yusuf[1], Sebastian Luna-Valero[19], Rory Kirchner[20], Karel Brinda[21], Thomas Wollmann[22], Martin Raden[1], Simon J. van Heeringen[23], Nicola Soranzo[24], Lorena Pantano[5], Zachary Charlop-Powers[25], Per Unneberg[26], Matthias De Smet[27], Marcel Martin[28], Greg Von Kuster[29], Tiago Antao[30], Milad Miladi[1], Kevin Thornton[31], Christian Brueffer[32], Marius van den Beek[33], Daniel Maticzka[1], Clemens Blank[1], Sebastian Will[34], Kévin Gravouil[35], Joachim Wolff[1], Manuel Holtgrewe[36,37], Jörg Fallmann[38], Vitor C. Piro[39,40], Ilya Shlyakhter[8], Ayman Yousif[41], Philip Mabon[42], Xiao-Ou Zhang[43], Wei Shen[44], Jennifer Cabral[42], Cristel Thomas[45], Eric Enns[42], Joseph Brown[46], Jorrit Boekel[47], Mattias de Hollander[48], Jerome Kelleher[49], Nitesh Turaga[50], Julian R. de Ruiter[51], Dave Bouvier[52], Simon Gladman[53], Saket Choudhary[54], Nicholas Harding[49], Florian Eggenhofer[1], Arne Kratz[11], Zhuoqing Fang[55], Robert Kleinkauf[56], Henning Timm[57], Peter J. A. Cock[58], Enrico Seiler[39], Colin Brislawn[59], Hai Nguyen[60], Endre Bakken Stovner[61], Philip Ewels[62], Matt Chambers[63], James E. Johnson[64], Emil Hägglund[65], Simon Ye[66], Roman Valls Guimera[67], Elmar Pruesse[68], W. Augustine Dunn[69], Lance Parsons[70], Rob Patro[71], David Koppstein[72], Elena Grassi[73], Inken Wohlers[74], Alex Reynolds[75], MacIntosh Cornwell[76], Nicholas Stoler[77], Daniel Blankenberg[78], Guowei He[79], Marcel Bargull[57], Alexander Junge[80], Rick Farouni[81], Mallory Freeberg[82], Sourav Singh[83], Daniel R. Bogema[84], Fabio Cumbo[85,86,77,87], Liang-Bo Wang[88,89], David E Larson[90], Matthew L. Workentine[91], Upendra Kumar Devisetty[92], Sacha Laurent[93], Pierrick Roger[94], Xavier Garnier[16,95], Rasmus Agren[96], Aziz Khan[97], John M Eppley[98], Wei Li[99], Bianca Katharina Stöcker[57], Tobias Rausch[100], James Taylor[101], Patrick R. Wright[1], Adam P. Taranto[102], Davide Chicco[103], Bengt Sennblad[26], Jasmijn A. Baaijens[104], Matthew Gopez[42], Nezar Abdennur[66], Iain Milne[58], Jens Preussner[105], Luca Pinello[81], Avi Srivastava[71], Aroon T. Chande[106], Philip Reiner Kensche[107], Yuri Pirola[108], Michael Knudsen[109], Ino de Bruijn[110], Kai Blin[111], Giorgio Gonnella[112], Oana M. Enache[8], Vivek Rai[113], Nicholas R. Waters[114], Saskia Hiltemann[115], Matthew L. Bendall[116,117], Christoph Stahl[118], Alistair Miles[49], Yannick Boursin[119], Yasset Perez-Riverol[120], Sebastian Schmeier[121], Erik Clarke[122], Kevin Arvai[123], Matthieu Jung[124], Tomás Di Domenico[125], Julien Seiler[124], Eric Rasche[1], Etienne Kornobis[126], Daniela Beisser[127], Sven Rahmann[128], Alexander S Mikheyev[129,130], Camy Tran[42], Jordi Capellades[131], Christopher Schröder[132], Adrian Emanuel Salatino[133], Simon Dirmeier[134], Timothy H. Webster[135], Oleksandr Moskalenko[136], Gordon Stephen[58], and Johannes Köster[†,137,138]

[1]Bioinformatics Group, Department of Computer Science, University of Freiburg, Freiburg, Germany

[2]Laboratory of Cellular and Developmental Biology, National Institute of Diabetes and Digestive and Kidney Diseases, National Institutes of Health, Bethesda, United States
[3]Division of CBRN Security and Defence, FOI - Swedish Defence Research Agency, Umeå. Sweden
[4]Department of Chemistry, Computational Life Science Cluster (CLiC), Umeå University, Umeå, Sweden
[5]Harvard T.H. Chan School of Public Health, Boston, United States
[6]NYU Abu Dhabi, Abu Dhabi, United Arab Emirates
[7]Department of Organismic and Evolutionary Biology, Harvard University, Cambridge, United States
[8]Broad Institute of MIT and Harvard, Cambridge, United States
[9]Laboratory of Bioinformatics and Computational Biology, A. C. Camargo Cancer Center, São Paulo, Brazil
[10]Holland Computing Center, University of Nebraska, Lincoln, United States
[11]Bioinformatics Research Centre, Aarhus University, Aarhus, Denmark
[12]Bioinformatics and Biostatistics Hub - C3BI, USR IP CNRS, Institut Pasteur, Paris, France
[13]Counsyl, South San Francisco, United States
[14]Department of Human Genetics, University of Utah, Eccles Institute of Human Genetics, Salt Lake City
[15]Erasmus Medical Center, Department of Urology, Rotterdam, The Netherlands
[16]INRA, UMR IGEPP, BioInformatics Platform for Agroecosystems Arthropods (BIPAA), Campus Beaulieu, Rennes, France
[17]Bioinformatics core facility, Max Planck Institute for Immunobiology and Epigenetics, Freiburg, Germany
[18]UPMC, CNRS, FR2424, ABiMS, Station Biologique, Roscoff, France
[19]MRC Weatherall Institute of Molecular Medicine, University of Oxford, Oxford, United Kingdom
[20]Department of Biostatistics, Harvard T.H. Chan School of Public Health, Boston, United States
[21]Center for Communicable Disease Dynamics, Department of Epidemiology, Harvard TH Chan School of Public Health, Boston, United States
[22]University of Heidelberg and DKFZ, Heidelberg, Germany
[23]Radboud University, Faculty of Science, Department of Molecular Developmental Biology, Radboud Institute for Molecular Life Sciences, Nijmegen, The Netherlands
[24]Earlham Institute, Norwich Research Park, Norwich, United Kindgom
[25]The Laboratory for Genetically Encoded Small Molecules, The Rockefeller University, New York, United States
[26]Department of Cell and Molecular Biology, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Uppsala University, Uppsala, Sweden
[27]Ghent University Hospital, Ghent University, Belgium
[28]Department of Biochemistry and Biophysics, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Stockholm University, Sweden
[29]Institute for CyberScience, Pennsylvania State University, University Park, United States

[30]Division of Biological Sciences, University of Montana, Missoula, United States of America
[31]Department of Ecology and Evolutionary Biology, University of California Irvine, Irvine, United States
[32]Division of Oncology and Pathology, Department of Clinical Sciences, Lund University, Lund, Sweden
[33]Stem Cells and Tissue Homeostasis, Institut Curie, Paris, France
[34]Institute for Theoretical Chemistry, University of Vienna, Vienna, Austria
[35]Université Clermont Auvergne, INRA, MEDIS, Clermont-Ferrand, France
[36]Core Unit Bioinformatics, Berlin Institute of Health, Berlin, Germany
[37]Charité Universitätsmedizin Berlin, Berlin, Germany
[38]Bioinformatics Group, Department of Computer Science and Interdisciplinary Center for Bioinformatics, University of Leipzig, Leipzig, Germany
[39]Bioinformatics Unit, Robert Koch Institute, Berlin, Germany
[40]CAPES Foundation, Ministry of Education of Brazil, Brasília, Brazil
[41]Center for Genomics and System Biology, New York University, Abu Dhabi, United Arab Emirates
[42]National Microbiology Laboratory, Public Health Agency of Canada, Winnipeg, Canada
[43]Program in Bioinformatics and Integrative Biology, University of Massachusetts Medical School, Worcester, United States
[44]Department of Clinical Laboratory, Chengdu Military General Hospital, Chengdu, China
[45]Northrop Grumman Corporation, Technology Services, Rockville, United States
[46]Biological Sciences Division, Pacific Northwest National Laboratory, Richland, United States
[47]Department of Oncology-Pathology, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Karolinska Institutet, Solna, Sweden
[48]Netherlands Institute of Ecology (NIOO-KNAW), Wageningen, The Netherlands
[49]Big Data Institute, Li Ka Shing Centre for Health Information and Discovery, University of Oxford, Oxford, United Kingdom
[50]Department of Biology, Johns Hopkins University, Baltimore, United States
[51]Divisions of Molecular Pathology and Molecular Carcinogenesis, The Netherlands Cancer Institute, Amsterdam, The Netherlands
[52]Department of Biochemistry Molecular Biology, Pennsylvania State University, University Park, United States
[53]Melbourne Bioinformatics, University of Melbourne, Melbourne, Australia
[54]Computational Biology and Bioinformatics, University of Southern California, Los Angeles, United States
[55]Shanghai Institutes for Biological Sciences, University of Chinese Academy of Sciences, Chinese Academy of Sciences, Shanghai China
[56]_
[57]Genome Informatics, Institute of Human Genetics, University Hospital Essen, University of Duisburg-Essen, Essen, Germany
[58]The James Hutton Institute, Dundee, United Kingdom
[59]Earth and Biological Sciences Directorate, Pacific Northwest National Laboratory,

Richland, United States
[60]Department of Chemistry Chemical Biology, Rutgers University, Piscataway, United States
[61]Department of Computer Science, Norwegian University of Science and Technology
[62]Department of Biochemistry and Biophysics, Science for Life Laboratory, Stockholm University, Stockholm, Sweden
[63]Department of Biochemistry, Molecular Biology and Biophysics (as contractor, not employee), University of Minnesota, Minneapolis, United States
[64]Minnesota Supercomputing Institute, University of Minnesota, Minneapolis, United States
[65]Department of Molecular Evolution, Cell and Molecular Biology, Science for Life Laboratory, Biomedical Centre, Uppsala University, Uppsala, Sweden
[66]Massachusetts Institute of Technology, Cambridge, United States
[67]Center for Cancer Research, University of Melbourne, Melbourne, Australia
[68]University of Colorado, Denver, United States
[69]Boston Children's Hospital, Boston, United States
[70]Lewis-Sigler Institute for Integrative Genomics, Princeton University, Princeton, United States
[71]Department of Computer Science, Stony Brook University, Stony Brook, United States
[72]The Kirby Institute of Infection and Immunity, University of New South Wales, Sydney, Australia
[73]Transcription and Chromatin Lab, Humanitas University, Rozzano, Italy
[74]Lübeck Interdisciplinary Platform for Genome Analytics (LIGA), Institutes of Neurogenetics and Integrative Experimental Genomics, University of Lübeck, Lübeck, Germany
[75]Altius Institute for Biomedical Sciences, Seattle, United States
[76]New York University School of Medicine, New York City, United States
[77]Department of Biochemistry and Molecular Biology, Pennsylvania State University, University Park, United States
[78]Genomic Medicine Institute, Lerner Research Institute, Cleveland Clinic, Cleveland, United States
[79]High Performance Computing, NYU Abu Dhabi, Abu Dhabi, United Arab Emirates
[80]Disease Systems Biology Program, Novo Nordisk Foundation Center for Protein Research, University of Copenhagen, Copenhagen, Denmark
[81]Massachusetts General Hospital and Harvard Medical School, Boston, United States
[82]EMBL-EBI, Wellcome Genome Campus, Hinxton, Cambridgeshire, United Kingdom
[83]Savitribai Phule Pune University, Pune, Maharashtra, India
[84]NSW Department of Primary Industries, Elizabeth Macarthur Agricultural Institute, Menangle, Australia
[85]Department of Engineering, Roma Tre University, Rome, Italy
[86]Institute for Systems Analysis and Computer Science "Antonio Ruberti", National Research Council of Italy, Rome, Italy
[87]SYSBIO.IT Center for Systems Biology, Milan, Italy
[88]Oncology Division, Department of Medicine, Washington University School of Medicine,

St. Louis, United States
[89]McDonnell Genome Institute, Washington University School of Medicine, St. Louis, United States
[90]The McDonnell Genome Institute, Washington University, St. Louis, United States
[91]Faculty of Veterinary Medicine, University of Calgary, Calgary, Canada
[92]CyVerse, Bio5 institute, University of Arizona, Tucson, United States
[93]Institute of Microbiology, Universitary Hospital of Lausanne, Switzerland
[94]CEA, LIST, Laboratory for data analysis and systems' intelligence, MetaboHUB, France
[95]Dyliss - Dynamics, Logics and Inference for biological Systems and Sequences, Inria/IRISA, Campus Beaulieu, Rennes, France
[96]Department of Biology and Biological Engineering, National Bioinformatics Infrastructure Sweden, Science for Life Laboratory, Chalmers University of Technology, Sweden
[97]Centre for Molecular Medicine Norway (NCMM), Nordic EMBL Partnership, University of Oslo, Oslo, Norway
[98]Daniel K. Inouye Center for Microbial Oceanography: Research and Education, Department of Oceanography, University of Hawaii, Honolulu, United States
[99]Department of Biostatistics and Computational Biology, Dana-Farber Cancer Institute, Harvard T.H. Chan School of Public Health, Boston, United States
[100]European Molecular Biology Laboratory (EMBL), Genomics Core Facility, Heidelberg, Germany
[101]Departments of Biology and Computer Science, Johns Hopkins University, Baltimore, United States
[102]Plant Sciences Division, Research School of Biology, The Australian National University, Canberra, Australia
[103]Princess Margaret Cancer Centre, Toronto, Canada
[104]Centrum Wiskunde and Informatica, Amsterdam, Netherlands
[105]ECCPS Bioinformatics Core Unit, Max Planck Institute for Heart and Lung Research, Bad Nauheim, Germany
[106]Applied Bioinformatics Laboratory, 2 Ravinia Drive, Suite 1200 Atlanta, GA 30346, United States
[107]German Cancer Research Center (DKFZ), Foundation under Public Law, Heidelberg, Germany
[108]Dip. di Informatica Sistemistica e Comunicazione, Univ. degli Studi di Milano-Bicocca, Milan, Italy
[109]Department of Molecular Medicine, Aarhus University Hospital, Aarhus, Denmark
[110]Department of Pathology, Memorial Sloan Kettering Cancer Center, New York, United States
[111]The Novo Nordisk Foundation Center for Biosustainability, Technical University of Denmark, Lyngby, Denmark
[112]ZBH - Center for Bioinformatics, MIN-Fakultät, Universität Hamburg, Hamburg, Germany
[113]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, United States
[114]Department of Microbiology, School of Natural Sciences, National University of Ireland,

Galway, Ireland Information and Computational Sciences, James Hutton Institute, Invergowrie, Scotland
[115]Erasmus Medical Center, Rotterdam, The Netherlands
[116]Computational Biology Institute, Milken Institute School of Public Health, The George Washington University, Washington, D.C., United States
[117]Department of Microbiology, Immunology Tropical Medicine, The George Washington University School of Medicine and Health Sciences, Washington, D.C., United States
[118]Genome Informatics, Institute of Human Genetics, University Hospital Essen, University of Duisburg Essen, Essen, Germany
[119]Institut Gustave Roussy, Villejuif, France
[120]European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Trust Genome Campus , Hinxton, Cambridge, United Kingdom
[121]Massey University, Institute of Natural and Mathematical Sciences, North Shore City, New Zealand
[122]Department of Microbiology, University of Pennsylvania, United States
[123]GeneDx, Gaithersburg, United States
[124]Institut de Génétique et de Biologie Moléculaire et Cellulaire, CNRS, Illkirch, France
[125]Wellcome Trust Sanger Institute, Wellcome Trust Genome Campus, United Kingdom
[126]Epigenetic Regulation Unit, Pasteur Institute, Paris, France
[127]Biodiversity, Faculty of Biology, University of Duisburg-Essen, Essen, Germany
[128]Genome Informatics, Institute of Human Genetics, University of Duisburg-Essen, University Hospital Essen, Essen, Germany
[129]Evolutionary Genomics Lab, Research School of Biology, The Australian National University, Canberra, Australia
[130]Ecology and Evolution Unit, Okinawa Institute of Science and Technology Graduate University, Onna-son, Kunigami-gun, Okinawa, Japan
[131]Universitat Rovira i Virgili, Spanish Biomedical Research Center in Diabetes and Associated Metabolic Disorders (CIBERDEM), Reus Spain
[132]Genome Informatics, Institute of Human Genetics, University of Duisburg-Essen, Essen, Germany
[133]Department of Molecular Genetics and Biology of Complex Diseases, Institute of Medical Research A Lanari-IDIM, University of Buenos Aires, National Scientific and Technical Research Council (CONICET), Ciudad Autónoma de Buenos Aires, Argentina.
[134]Department of Biosystems Science and Engineering, ETH Zurich, Basel, Switzerland
[135]School of Life Sciences, Arizona State University, Tempe, United States
[136]UFIT Research Computing, University of Florida, Gainesville, United States
[137]Algorithms for reproducible bioinformatics, Genome Informatics, Institute of Human Genetics, University Hospital Essen, University of Duisburg-Essen
[138]Dana Farber Cancer Institute, Harvard Medical School, Boston, United States

October 27, 2017

*Co-first author
†To whom correspondence should be addressed.

Abstract

We present Bioconda (`https://bioconda.github.io`), a distribution of bioinformatics software for the lightweight, multi-platform and language-agnostic package manager Conda. Currently, Bioconda offers a collection of over 3000 software packages, which is continuously maintained, updated, and extended by a growing global community of more than 200 contributors. Bioconda improves analysis reproducibility by allowing users to define isolated environments with defined software versions, all of which are easily installed and managed without administrative privileges.

Introduction

Thousands of new software tools have been released for bioinformatics in recent years, in a variety of programming languages. Accompanying this diversity of construction is an array of installation methods. Often, Software has to be compiled manually for different hardware architectures and operating systems, with management left to the user or system administrator. Scripting languages usually deliver their own package management tools for installing, updating, and removing packages, though these are often limited in scope to packages written in the same scripting language and cannot handle external dependencies (e.g., C libraries). Published scientific software often consists of simple collections of custom scripts distributed with textual descriptions of the manual steps required to install the software. New analyses often require novel combinations of multiple tools, and the heterogeneity of scientific software makes management of a software stack complicated and error-prone. Moreover, it inhibits reproducible science (Mesirov, 2010; Baker, 2016; Munafò et al., 2017), because it is hard to reproduce a software stack on different machines. System-wide deployment of software has traditionally been handled by administrators, but reproducibility often requires that the researcher (who is often not an expert in administration) is able to maintain full control of the software environment and rapidly modify it without administrative privileges.

The Conda package manager (`https://conda.io`) has become an increasingly popular approach to overcome these challenges. Conda normalizes software installations across language ecosystems by describing each software package with a *recipe* that defines meta-information and dependencies, as well as a *build script* that performs the steps necessary to build and install the software. Conda prepares and builds software packages within an isolated environment, transforming them into relocatable binaries. Conda packages can be built for all three major operating systems: Linux, macOS, and Windows. Importantly, installation and management of packages requires no administrative privileges, such that a researcher can control the available software tools regardless of the underlying infrastructure. Moreover, Conda obviates reliance on system-wide installation by allowing users to generate isolated software environments, within which versions and tools can be managed per-project, without generating conflicts or incompatibilities (see online methods). These environments support reproducibility, as they can can be rapidly exchanged via files that describe their installation state. Conda is tightly integrated into popular solutions for reproducible scientific data analysis like Galaxy (Afgan et al., 2016), bcbio-nextgen (`https://github.com/chapmanb/bcbio-nextgen`), and Snakemake (Köster and Rahmann, 2012). Finally, while Conda provides many commonly-used packages by default, it also allows users to optionally include additional repositories (termed *channels)* of packages that can be installed.

Results

In order to unlock the benefits of Conda for the life sciences, the Bioconda project was founded in 2015. The mission of Bioconda is to make bioinformatics software easily installable and manageable via the Conda package manager. Via its channel for the Conda package manager, Bioconda currently provides over 3000 software packages for Linux and macOS. Development is driven by an open community of over 200 international scientists. In the prior two years, package count and the number of contributors have increased

linearly, on average, with no sign of saturation (Fig. 1a,b). The barrier to entry is low, requiring a willingness to participate and adherence to community guidelines. Many software developers contribute recipes for their own tools, and many Bioconda contributors are invested in the project as they are also users of Conda and Bioconda. Bioconda provides packages from various language ecosystems like Python, R (CRAN and Bioconductor), Perl, Haskell, as well as a plethora of C/C++ programs (Fig. 1c). Many of these packages have complex dependency structures that require various manual steps to install when not relying on a package manager like Conda (Fig. 2a, Online Methods). With over 6.3 million downloads, the service has become a backbone of bioinformatics infrastructure (Fig. 1d). Bioconda is complemented by the conda-forge project (https://conda-forge.github.io), which hosts software not specifically related to the biological sciences. The two projects collaborate closely, and the Bioconda team maintains over 500 packages hosted by conda-forge. Among all currently available distributions of bioinformatics software, Bioconda is by far the most comprehensive, while being among the youngest (Fig. 2d).

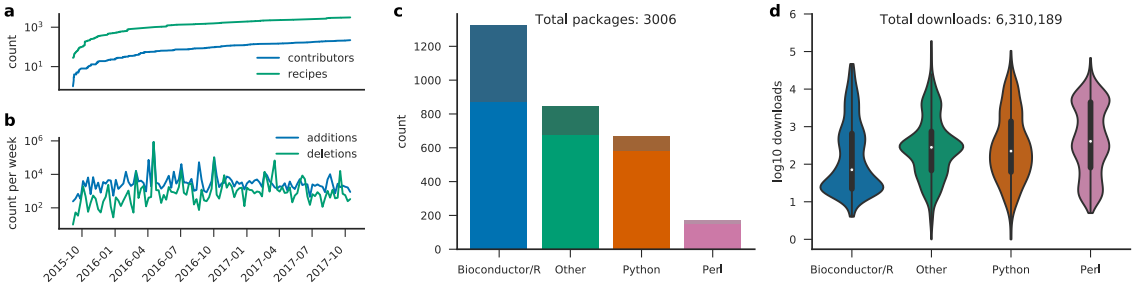


Figure 1: Bioconda development and usage since the beginning of the project. (a) contributing authors and added recipes over time. (b) code line additions and deletions per week. (c) package count per language ecosystem (saturated colors on bottom represent explicitly life science related packages). (d) total downloads per language ecosystem. The term "other" entails all recipes that do not fall into one of the specific categories. Note that a subset of packages that started in Bioconda have since been migrated to the more appropriate, general-purpose conda-forge channel. Older versions of such packages still reside in the Bioconda channel, and as such are included in the recipe count (a) and download count (d). Statistics obtained Oct. 25, 2017.

To ensure reliable maintenance of such numbers of packages, we use a semi-automatic, agent-assisted development workflow (Fig. 2b). All Bioconda recipes are hosted in a GitHub repository (https://github.com/bioconda/bioconda-recipes). Both the addition of new recipes and the update of existing recipes in Bioconda is handled via *pull requests*. Thereby, a modified version of one or more recipes is compared against the current state of Bioconda. Once a pull request arrives, our infrastructure performs several automatic checks. Problems discovered in any step are reported to the contributor and further progress is blocked until they are resolved. First, the modified recipes are checked for syntactic anti-patterns, i.e., formulations that are syntactically correct but bad style (termed *linting*). Second, the modified recipes are built on Linux and macOS, via a cloud based, free-of-charge service (https://travis-ci.org). Successfully built recipes are tested (e.g., by running the generated executable). Since Bioconda packages must be able to run on any supported system, it is important to check that the built packages do not rely on particular elements from the build environment. Therefore, testing happens in two stages: (a) test cases are executed in the build environment (b) test cases are executed in a minimal Docker (https://docker.com) container which purposefully lacks all non-common system libraries (hence, a dependency that is not explicitly defined will lead to a failure). Once the *build* and *test* steps have succeeded, a member of the Bioconda team reviews the proposed changes and, if acceptable, merges the modifications into the official repository. Upon merging, the recipes are built again and uploaded to the hosted Bioconda channel (https://anaconda.org/bioconda), where they become available via the Conda package manager. When a Bioconda package is updated to a new version, older builds are generally preserved, and recipes for multiple older versions may be maintained

in the Bioconda repository. The usual turnaround time of above workflow is short (Fig. 2d). 61% of the pull requests are merged within 5 hours. Of those, 36% are even merged within 1 hour. Only 18% of the pull requests need more than a day. Hence, publishing software in Bioconda or updating already existing packages can be accomplished typically within minutes to a few hours.

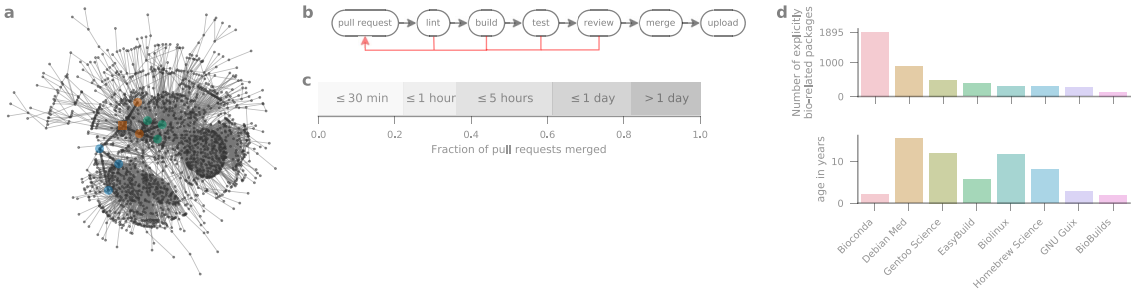


Figure 2: Dependency structure, workflow, comparison with other resources, and turnaround time. (a) largest connected component of directed acyclic graph of Bioconda packages (nodes) and dependencies (edges). Highlighted is the induced subgraph of the CNVkit (Talevich et al., 2016) package and it's dependencies (node coloring as defined in Fig. 1c, squared node represents CNVkit). (b) GitHub based development workflow: a contributor provides a pull request that undergoes several build and test steps, followed by a human review. If any of these checks does not succeed, the contributor can update the pull request accordingly. Once all steps have passed, the changes can be merged. (c) Turnaround time from submission to merge of pull requests in Bioconda. (d) Comparison of explicitly life science related packages in Bioconda with Debian Med (https://www.debian.org/devel/debian-med), Gentoo Science Overlay (category sci-biology, `https://github.com/gentoo/sci`), EasyBuild (module bio, https://easybuilders.github.io/easybuild), Biolinux (Field et al., 2006), Homebrew Science (tag bioinformatics, https://brew.sh), GNU Guix (category bioinformatics, https://www.gnu.org/s/guix), and BioBuilds (`https://biobuilds.org`). The lower panel shows the project age since the first release or commit. Statistics obtained Oct. 23, 2017.

Reproducible software management and distribution is enhanced by other current technologies. Conda integrates itself well with environment modules (`http://modules.sourceforge.net/`), a technology used nearly universally across HPC systems. An administrator can use Conda to easily define software stacks for multiple labs and project-specific configurations. Popularized by Docker, containers provide another way to publish an entire software stack, down to the operating system. They provide greater isolation and control over the environment a software is executed in, at the expense of some customizability. Conda complements container-based approaches. Where flexibility is needed, Conda packages can be used and combined directly. Where the uniformity of containers is required, Conda can be used to build images without having to reproduce the nuanced installation steps that would ordinarily be required to build and install a software within an image. In fact, for each Bioconda package, our build system automatically builds a minimal Docker image containing that package and its dependencies, which is subsequently uploaded and made available via the Biocontainers project (da Veiga Leprevost et al., 2017). As a consequence, every built Bioconda package is available not only for installation via Conda, but also as a container via Docker, Rkt (https://coreos.com/rkt), and Singularity (Kurtzer et al., 2017), such that the desired level of reproducibility can be chosen freely (Grüning et al., 2017).

Discussion

By turning the arduous and error-prone process of installing bioinformatics software, previously repeated endlessly by scientists around the globe, into a concerted community effort, Bioconda frees significant resources to instead be invested into productive research. The new simplicity of deploying even complex software stacks with strictly controlled software versions enables software authors to safely rely on existing methods. Where previously the cost of depending on a third party tool - requiring its installation and maintaining compatibility with new versions - was often higher than the effort to re-implement its methods, authors can now simply specify the tool and version required, incurring only negligible costs even for large requirement sets.

For reproducible data science, it is crucial that software libraries and tools are provided via an easy to use, unified interface, such that they can be easily deployed and sustainably managed. With its ability to maintain isolated software environments, the integration into major workflow management systems and the fact that no administration privileges are needed, the Conda package manager is the ideal tool to ensure sustainable and reproducible software management. With Bioconda, we unlock Conda for the life sciences while coordinating closely with other related projects such as conda-forge and Biocontainers. Bioconda offers a comprehensive resource of thousands of software libraries and tools that is maintained by hundreds of international contributors. Although it is among the youngest, it outperforms all competing projects by far in the number of available packages. With almost six million downloads so far, Bioconda packages have been well received by the community. We invite everybody to participate in reaching the goal of a central, comprehensive, and language agnostic collection of easily installable software by maintaining existing or publishing new software in Bioconda.

Funding

The Bioconda project has received support from Anaconda, Inc., Austin, TX, USA, in the form of expanded storage for Bioconda packages on their hosting service (`https://anaconda.org`). Further, the project has been granted extended build times from Travis CI, GmbH (https://travis-ci.com). The Bioconda community also would like to thank ELIXIR (https://www.elixir-europe.org) for their constant support and donating staff.

Acknowledgements

We thank the participants of various hackathons (e.g., the GalaxyP and IUC contribution fest, ELIXIR BioContainers and NETTAB hackathon) for porting numerous packages to Bioconda.

Contributions

Kyle Beauchamp, Christian Brueffer, Brad Chapman, Ryan Dale, Florian Eggenhofer, Björn Grüning, Johannes Köster, Elmar Pruesse, Martin Raden, Jillian Rowe, Devon Ryan, Ilya Shlyakter, Andreas Sjödin, Christopher Tomkins-Tinch, and Renan Valieris (in alphabetical order) have written the manuscript. Johannes Köster and Ryan Dale have conducted the data analysis. Dan Ariel Sondergaard contributed by supervising student programmers on contributing recipes and maintaining the connection with ELIXIR. All other authors have contributed or maintained recipes.

Online Methods

Security Considerations

Using Bioconda as a service to obtain packages for local installation entails trusting that (a) the provided software itself is not harmful and (b) it has not been modified in a harmful way. Ensuring (a) is up to the user. In contrast, (b) is handled by our workflow. First, source code or binary files defined in recipes are checked for integrity via MD5 or SHA256 hash values. Second, all review and testing steps are enforced via the GitHub interface. This guarantees that all packages have been tested automatically and reviewed by a human being. Third, all changes to the repository of recipes are publicly tracked, and all build and test steps are transparently visible to the user. Finally, the automatic parts of the development workflow are implemented in the open-source software *bioconda-utils* (`https://github.com/bioconda/bioconda-utils`). In the future, we will further explore the possibility to sign packages cryptographically.

Software management with Conda

Via the Conda package manager, installing software from Bioconda becomes very simple. In the following, we describe the basic functionality assuming that the user has access to a Linux or macOS terminal. After installing Conda, the first step is to set up the Bioconda channel via:

```
$ conda config --add channels conda-forge
$ conda config --add channels bioconda
```

Now, all Bioconda packages are visible to the Conda package manager. For example, the software CNVkit (Talevich et al., 2016), can be searched for with

```
$ conda search cnvkit
```

in order to check if and in which versions it is available. It can be installed with:

```
$ conda install cnvkit
```

CNVkit needs various dependencies from Python and R, which would otherwise have to be installed in separate manual steps (Fig. 2a). Furthermore, Conda enables updating and removing all these dependencies via one unified interface. A key value of Conda is the ability to define isolated, shareable software environments. This can happen ad-hoc, or via YAML (`https://yaml.org`) files. For example, the following defines an environment consisting of Salmon (Patro et al., 2017) and DESeq2 (Love et al., 2014):

```
channels:
  - bioconda
  - conda-forge
  - defaults
dependencies:
  - bioconductor-deseq2 =1.16.1
  - salmon =0.8.2
  - r-base =3.4.1
```

Given that the above environment specification is stored in the file `env.yaml`, an environment `my-env` meeting the specified requirements can be created via the command:

```
$ conda env create --name my-env --file env.yaml
```

To use the commands installed in this environment, it must first be "activated" by issuing the following command:

```
$ source activate my-env
```

Within the environment, R, Salmon, and DESeq2 are available in exactly the defined versions. For example, salmon can be executed with:

```
$ salmon --help
```

It is possible to modify an existing environment by using `conda update`, `conda install` and `conda remove`. For example, we could add a particular version of Kallisto (Bray et al., 2016) and update Salmon to the latest available version with:

```
$ conda install kallisto=0.43.1
$ conda update salmon
```

Finally, the environment can be deactivated again with:

```
$ source deactivate
```

How isolated software environments enable reproducible research

With isolated software environments as shown above, it is possible to define an exact version for each package. This increases reproducibility by eliminating differences due to implementation changes. Note that above we also pin an R version, although the latest compatible one would also be automatically installed without mentioning it. To further increase reproducibility, this pattern can be extended to all dependencies of DESeq2 and Salmon and recursively down to basic system libraries like zlib and boost (`https://www.boost.org`). Environments are isolated from the rest of the system, while still allowing interaction with it: e.g., tools inside the environment are preferred over system tools, while system tools that are not available from within the environment can still be used. Conda also supports the automatic creation of environment definitions from already existing environments. This allows to rapidly explore the needed combination of packages before it is finalized into an environment definition. When used with workflow management systems like Galaxy (Afgan et al., 2016), bcbio-nextgen (`https://github.com/chapmanb/bcbio-nextgen`), and Snakemake (Köster and Rahmann, 2012) that interact directly with Conda, a data analysis can be shipped and deployed in a fully reproducible way, from description and automatic execution of every analysis step down to the description and automatic installation of any required software.

Data analysis

The presented figures and numbers have been generated via a fully automated, reproducible Snakemake (Köster and Rahmann, 2012) workflow that is freely available under `https://github.com/bioconda/bioconda-paper`.

References

E Afgan, D Baker, den Beek M van, D Blankenberg, D Bouvier, M Čech, J Chilton, D Clements, N Coraor, C Eberhard, B Grüning, A Guerler, J Hillman-Jackson, Kuster G Von, E Rasche, N Soranzo, N Turaga, J Taylor, A Nekrutenko, and J Goecks. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res*, 44:W3–W10, Jul 2016. doi: 10.1093/nar/gkw343. URL https://doi.org/10.1093/nar/gkw343.

Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, may 2016. doi: 10.1038/533452a. URL https://doi.org/10.1038%2F533452a.

NL Bray, H Pimentel, P Melsted, and L Pachter. Near-optimal probabilistic RNA-seq quantification. *Nat Biotechnol*, 34:525–7, May 2016. doi: 10.1038/nbt.3519. URL https://doi.org/10.1038/nbt.3519.

F da Veiga Leprevost, BA Grüning, Aflitos S Alves, HL Röst, J Uszkoreit, H Barsnes, M Vaudel, P Moreno, L Gatto, J Weber, M Bai, RC Jimenez, T Sachsenberg, J Pfeuffer, Alvarez R Vera, J Griss, AI Nesvizhskii, and Y Perez-Riverol. BioContainers: an open-source and community-driven framework for software standardization. *Bioinformatics*, 33:2580–2582, Aug 2017. doi: 10.1093/bioinformatics/btx192. URL https://doi.org/10.1093/bioinformatics/btx192.

Dawn Field, Bela Tiwari, Tim Booth, Stewart Houten, Dan Swan, Nicolas Bertrand, and Milo Thurston. Open software for biologists: from famine to feast. *Nature Biotechnology*, 24(7):801–803, jul 2006. doi: 10.1038/nbt0706-801. URL https://doi.org/10.1038%2Fnbt0706-801.

Björn Grüning, John Chilton, Johannes Köster, Ryan Dale, Jeremy Goecks, Rolf Backofen, Anton Nekrutenko, and James Taylor. Practical computational reproducibility in the life sciences. oct 2017. doi: 10.1101/200683. URL https://doi.org/10.1101%2F200683.

GM Kurtzer, V Sochat, and MW Bauer. Singularity: Scientific containers for mobility of compute. *PLoS One*, 12:e0177459, 2017. doi: 10.1371/journal.pone.0177459. URL https://doi.org/10.1371/journal.pone.0177459.

J Köster and S Rahmann. Snakemake–a scalable bioinformatics workflow engine. *Bioinformatics*, 28:2520–2, Oct 2012. doi: 10.1093/bioinformatics/bts480. URL https://doi.org/10.1093/bioinformatics/bts480.

MI Love, W Huber, and S Anders. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol*, 15:550, 2014. doi: 10.1186/s13059-014-0550-8. URL https://doi.org/10.1186/s13059-014-0550-8.

J. P. Mesirov. Accessible Reproducible Research. *Science*, 327(5964):415–416, jan 2010. doi: 10.1126/science.1179653. URL https://doi.org/10.1126%2Fscience.1179653.

Marcus R. Munafò, Brian A. Nosek, Dorothy V. M. Bishop, Katherine S. Button, Christopher D. Chambers, Nathalie Percie du Sert, Uri Simonsohn, Eric-Jan Wagenmakers, Jennifer J. Ware, and John P. A. Ioannidis. A manifesto for reproducible science. *Nature Human Behaviour*, 1(1):0021, jan 2017. doi: 10.1038/s41562-016-0021. URL https://doi.org/10.1038%2Fs41562-016-0021.

R Patro, G Duggal, MI Love, RA Irizarry, and C Kingsford. Salmon provides fast and bias-aware quantification of transcript expression. *Nat Methods*, 14:417–419, Apr 2017. doi: 10.1038/nmeth.4197. URL https://doi.org/10.1038/nmeth.4197.

E Talevich, AH Shain, T Botton, and BC Bastian. CNVkit: Genome-Wide Copy Number Detection and Visualization from Targeted DNA Sequencing. *PLoS Comput Biol*, 12:e1004873, Apr 2016. doi: 10.1371/journal.pcbi.1004873. URL https://doi.org/10.1371/journal.pcbi.1004873.

On Evidential Clustering with Partial Supervision

Violaine Antoine[1( )], Kévin Gravouil[1,2], and Nicolas Labroche[3]

[1] Clermont Auvergne University, UMR 6158, LIMOS,
63000 Clermont-Ferrand, France
violaine.antoine@uca.fr
[2] Clermont Auvergne University, INRA, MEDIS, LMGE,
63000 Clermont-Ferrand, France
kevin.gravouil@uca.fr
[3] University of Tours, LIFAT, EA 6300, Blois, France
nicolas.labroche@univ-tours.fr

Abstract. This paper introduces a new semi-supervised evidential clustering algorithm. It considers label constraints and exploits the evidence theory to create a credal partition coherent with the background knowledge. The main characteristics of the new method is its ability to express the uncertainties of partial prior information by assigning each constrained object to a set of labels. It enriches previous existing algorithm that allows the preservation of the uncertainty in the constraint by adding the possibility to favor crisp decision following the inherent structure of the dataset. The advantages of the proposed approach are illustrated using both a synthetic dataset and a real genomics dataset.

Keywords: Evidential clustering · Partial labels
Semi-supervised clustering · Belief function

1 Introduction

Evidential clustering algorithms, such as ECM [1], rely on the theoretical foundation of belief functions and evidence theory [2] and allow to express many types of uncertainty about the assignment of an object to a cluster. It enables to handle crisp single cluster assignment, as well as cluster membership degrees, total ignorance and outliers detection. The credal partition, which is formed with the assignments of all the objects, generalizes other soft partitions such as fuzzy, possibilistic or rough partitions [3].

Clustering is a complex unsupervised task that often requires additional assumptions to determine relevant solutions. The performances of a clustering algorithm can be highly improved by using background knowledge [4]. To this end, several semi-supervised evidential clustering approaches have been proposed [5–7]. In [7], the SECM-pl algorithm integrates prior information in the form of labeled data instances. The particularity of SECM-pl is its ability to

S. Destercke et al. (Eds.): BELIEF 2018, LNAI 11069, pp. 14–21, 2018.
https://doi.org/10.1007/978-3-319-99383-6_3

handle partial knowledge, which corresponds to the uncertainty about the assignment of an object to several classes. This partial knowledge is controlled by the algorithm in such a way that the uncertainty can be preserved.

In this paper, we propose an approach that generalizes SECM-pl, which maintains a high flexibility on the constraints, by favoring a decision making on the constraints. The paper is organized as follows: Sect. 2 recalls the basics concerning the evidence theory and its application in clustering. Section 3 details the novel SECM algorithm and focuses on how labels constraints are expressed and incorporated in ECM. Section 4 presents experimental settings and results. Finally a discussion and future work are presented in Sect. 5.

2 Preliminaries

2.1 Belief Functions

The evidence theory (or belief functions theory) [2,8] is a mathematical framework that enables to reflect the state of partial and unreliable knowledge. Let $\Omega = \{\omega_1, \ldots, \omega_c\}$ be the frame of discernment where ω_i is the true state of the system which will be defined below. The mass function $m : 2^\Omega \rightarrow [0,\ 1]$, also called basic belief assignment (bba), measures the degree of belief that ω_i belongs to a subset $A \subseteq \Omega$. It satisfies $\sum_{A \subseteq \Omega} m(A) = 1$. Any subset A such that $m(A) > 0$ is named a focal set of m. Given a mass function m, the plausibility function $pl : 2^\Omega \rightarrow [0,\ 1]$ is defined by:

$$pl(A) = \sum_{B \cap A \neq \emptyset} m(B), \qquad \forall A \subseteq \Omega. \tag{1}$$

The quantity $pl(A)$ corresponds to the maximal degree of belief that could be given to A. To make a decision, a mass function can be transformed into a pignistic probability distribution $BetP$ [8].

2.2 Evidential C-Means

Evidential clustering algorithms generate for each object $\mathbf{x}_1, \ldots \mathbf{x}_i, \ldots \mathbf{x}_n \in \mathbb{R}^p$ a mass function $\mathbf{m}_i$ on the set $\Omega = \{\omega_1, \ldots, \omega_c\}$ denoting the clusters. The collection $\mathbf{M} = (\mathbf{m}_1, \ldots, \mathbf{m}_n)$ forms the credal partition and allows to represent the uncertainties and imprecisions regarding the class membership of each object. ECM [1] is the credibilistic version of Fuzzy C-Means [9]. It considers for each subset $A_j \subseteq \Omega$ a representation of the subset with a prototype vector $\mathbf{v}_j$ in $\mathbb{R}^p$. The objective function is:

$$J_{ECM}(\mathbf{M}, \mathbf{V}) = \sum_{i=1}^{n} \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} |A_j|^\alpha m_{ij}^\beta d_{ij}^2 + \sum_{i=1}^{n} \rho^2 m_{i\emptyset}^\beta, \tag{2}$$

where $\mathbf{V}$ is the collection of prototypes, $m_{ij} = m_i(A_j)$ corresponds to the bba of the object $\mathbf{x}_i$ for the subset A_j, $m_{i\emptyset}$ denotes the mass of $\mathbf{x}_i$ allocated to the

empty set and d_{ij}^2 represents the squared Euclidean distance between $\mathbf{x}_i$ and the prototype $\mathbf{v}_j$. The last term of the objective function enables to handle the empty set which can be interpreted as a cluster for outliers. The ρ parameter is a fixed coefficient representing the distance between any object and the empty set. The two parameters α and $\beta > 1$ are introduced to penalize the degree of belief assigned to subsets with a high cardinality and to control the fuzziness of the partition. The objective function is subject to

$$\sum_{j/A_j \subseteq \Omega, A_j \neq \emptyset} m_{ik} + m_{i\emptyset} = 1; \quad m_{ij} \geq 0 \quad \forall i = \{1, \dots n\}, \forall j / A_j \subseteq \Omega. \tag{3}$$

2.3 SECM-pl

The main idea of the algorithm [7] is to add a penalty term in the objective function of ECM, in order to take into account a set of already labeled objects. Any mass function which partially or fully respects a constraint on a label ω_k has a high plausibility $pl(\omega_k)$ given to the label. Similarly, an object constrained in several classes, i.e. on the set $A_j \subset \Omega$ is respected with mass functions given a high plausibility $pl(A_j)$. Thus, the following penalty term has been proposed:

$$J_S = \sum_{i=1}^{n} \sum_{A_j \subset \Omega, A_j \neq \emptyset} b_{ij}(1 - Pl_i(A_j)), \tag{4}$$

where $b_{ij} = 1$ if $\mathbf{x}_i$ is constrained on A_j and 0 otherwise.

3 New ECM Algorithm with Partial Supervision

3.1 Modeling the Constraints

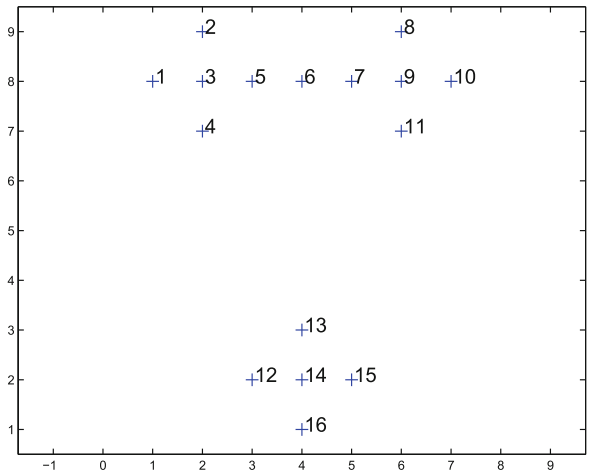
Let us consider a set of partially labeled constraints, i.e. a collection of objects $\mathbf{x}_i$ such that $\mathbf{x}_i \in A_j$, $\forall A_j \neq \emptyset$. If A_j is a singleton, then the object i belongs to a class with certainty. Otherwise, $\mathbf{x}_i$ belongs to a class listed in A_j without knowing which one more precisely. Notice that $\mathbf{x}_i \in \Omega$ corresponds to complete ignorance concerning the class of the object i. Degrees of belief containing the set of clusters A_j or a part of it should be favored as well as mass functions of subsets with a low cardinality. Thus, we define the measure $1 \geq T_{ij} \geq 0$ by the following formula:

$$T_{ij} = T_i(A_j) = \sum_{A_j \cap A_l \neq \emptyset} \frac{|A_j \cap A_l|^{\frac{r}{2}}}{|A_l|^r} m_{il}, \quad \forall i \in \{1 \dots n\}, A_j \subseteq \Omega, \tag{5}$$

where $r \geq 0$ controls a degree of penalization of the subsets. The coefficient $|A_l|^r$ is used to penalize subsets with a high cardinality and $|A_j \cap A_l|^{\frac{r}{2}}$ allows to concentrate efforts on subsets containing mostly elements of A_j. Notice that when $r = 0$, T_{ij} corresponds to the plausibility that the object $\mathbf{x}_i$ belongs to A_j. For the rest of the paper, we set $r = 1$.

3.2 Illustration

The behavior of the new measure T_{ij} is illustrated with the DiamondK3 dataset presented Fig. 1(a). This dataset is composed of 15 objects that should be separated into 3 groups. As it can be observed, points 13 to 16 are well isolated, whereas objects 1 to 11 seem to correspond to two natural clusters connected by the object 6. Let us suppose that some partial knowledge is available: e.g. object 6 is in the cluster ω_1 and object 13 belongs either to ω_1 or to ω_3, but not to ω_2. Thus, we obtain the two following constraints: $\mathbf{x}_6 \in \{\omega_1\}$ and $\mathbf{x}_{13} \in \{\omega_1, \omega_3\}$.



(a)

$m_i(A_j)$	$\emptyset$	0	0	0	0	0
	ω_1	**1**	0	0	0	0
	ω_2	0	0	0	0	**1**
	$\{\omega_1, \omega_2\}$	0	0	**1**	0	0
	ω_3	0	0	0	0	0
	$\{\omega_1, \omega_3\}$	0	**1**	0	0	0
	$\{\omega_2, \omega_3\}$	0	0	0	0	0
	Ω	0	0	0	**1**	0
$T_i(A_j)$	ω_1	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$	0
	$\{\omega_1, \omega_3\}$	1	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	$\frac{1}{3}$	0
$Pl(A_j)$	ω_1	1	1	1	1	0
	$\{\omega_1, \omega_3\}$	1	1	1	1	0

(b)

Fig. 1. DiamondK3 dataset (a) and illustration of the proposed penalty term $T_i(A_j)$ when considering several possible mass functions and compared to penalty term based on plausibility $Pl(A_j)$ for previous SECM-pl [7] (b).

Figure 1(b) presents in each column a set of possible mass functions for an object $\mathbf{x}_i$ coming from the DiamondK3 dataset. First, let us consider that $\mathbf{x}_i = \mathbf{x}_6$ and let us assume that $m_6(\omega_1) = 1$ as shown in the first column of Fig. 1(b). Thus, the constraint is respected and it can be observed that $T_6(\omega_1) = 1$. Inversely, if $m_6(\omega_2) = 1$ as presented in the last column of Fig. 1(b), then the constraint is totally neglected and $T_6(\omega_1) = 0$. Other columns illustrate partial respect of the constraint, since the bba is allocated to subsets containing the label ω_1. The larger the cardinality of the subset, the lower the value of T_{ij}.

Let us assume that $\mathbf{x}_i = \mathbf{x}_{13}$ and let us focus on the value obtained by $T_i(\{\omega_1, \omega_3\})$ for the set of possible mass functions. As it can be observed, $T_{ij} = 0$ when no focal sets contain ω_1 and/or ω_3. Conversely, if there exists a degree of belief not null on a subset including at least one of the classes included in the constraint, then $T_{ij} > 0$. As previously, the larger the cardinality of the subset, the lower the value of T_{ij}. For the same amount of subsets, for example columns 2 and 3 in Fig. 1(b), a higher value is given to subsets containing the most of classes in the constraint, i.e. $\{\omega_1, \omega_3\}$. This is a significant difference with the plausibility measure for which all subsets intersecting with the constraints contribute equally to the final value.

3.3 Objective Function and Optimization

Based on the mass function m_i of an object i, we can quantify the degree to which a partial constraint $\mathbf{x}_i \in A_j$ is respected by computing T_{ij} in Eq. (5). $T_{ij} = 1$ when the belief is given to a cluster in A_j and is 0 when the belief is assigned to none of the clusters included in A_j, i.e. when the constraint is not respected. If we consider now that the bbas have to be found, a natural requirement is to obtain a value of T_{ij} as high as possible if there exists a constraint such that $\mathbf{x}_i \in A_j$. This goal is achieved by minimizing the following objective function:

$$J_{SECM}(M,V) = (1-\gamma)\frac{1}{2^c n} J_{ECM}(M,V) + \gamma \frac{1}{s} \sum_{i=1}^{n} \sum_{A_j \subset \Omega, A_j \neq \emptyset} b_{ij}(1-T_{ij}), \quad (6)$$

such that constraints (3) are respected, s corresponds to the number of constraints, and $b_{ij} = 1$ if $\mathbf{x}_i \in A_j$, i.e if the object i is constrained with A_j and 0 otherwise.

The coefficient γ controls the tradeoff between the objective function of ECM and the constraints. Notice that if $r = 0$ for the computation of T_{ij}, then J_{SECM} is identical to the objective function proposed in [7]. Such setting allows the penalty term to give equal importance to any subset intersecting with the constraints, whereas $r > 0$ favors subsets with low cardinality. As ECM, the credal partitioning is carried out through an iterative optimization of the objective function, with the update of the mass functions and the prototypes. If β is set to 2, then the problem becomes quadratic with linear constraints and can be resolved with classical methods, for instance [10].

4 Experimentations

4.1 Toy Example

To illustrate the behavior of the SECM algorithm, we used the DiamondK3 dataset. First, an execution of ECM is performed with $\alpha = 1$, $\beta = 2$, $\rho^2 = 10^3$ and the final mass functions for the most representative subsets varying with the objects number are presented Fig. 2(a). It can be seen that ECM identifies the 3 clusters by assigning the belief to the 3 singletons. The object 6, which is located between the cluster ω_1 and ω_2, is ambiguous as it can belong to either ω_1 or ω_2. Thus, ECM assigns for $\mathbf{x}_6$ a high mass for the subset $\{\omega_1, \omega_2\}$.

Let us consider now that the following set of constraints are available: $\mathbf{x}_5 \in \{\omega_1\}$, $\mathbf{x}_6 \in \{\omega_2\}$ and $\mathbf{x}_{13} \in \{\omega_1, \omega_2\}$. The SECM algorithm is executed with $\gamma = 0.5$ and the credal partition obtained is presented Fig. 2(b). As it can be observed, constraints are well respected. The object 6, previously ambiguous with the ECM algorithm, is now assigned with certainty to ω_2. Similarly, the object 5 had with ECM its belief divided into $\{\omega_1, \omega_2\}$ and ω_1, whereas now all its belief is given to $\{\omega_1\}$. Finally, the mass function $m_{13}(\omega_3)$ for the object 13, which is already high with ECM, has increased with SECM. It shows that SECM is able to constrained $\mathbf{x}_{13}$ more specifically on ω_3 following the inherent structure of the dataset.

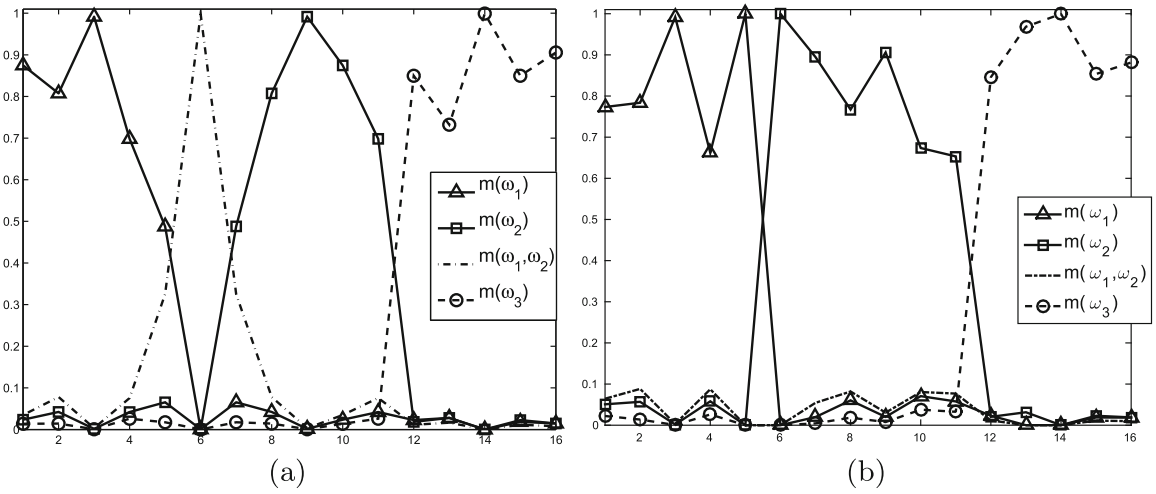


Fig. 2. Credal partitions obtained for DiamondK3 with (a) ECM and (b) SECM such that $\mathbf{x}_6 \in \{\omega_1\}$ and $\mathbf{x}_5 \in \{\omega_2\}$ and $\mathbf{x}_{13} \in \{\omega_1, \omega_3\}$.

4.2 Genomics Application

Dataset: Dozens of thousands microorganism's genomes are available in public databases. We selected three known genomes from the RefSeq database [11], namely *Clostridium acetoburylicum*, *Bacillus cereus* and *Brachyspira hyodysenteriae*, to simulate a small microbial community. DNA sequences were extracted from these genomes then embedded in numerical vectors using normalized tetranucleotide frequencies with a CONCOCT-inspired approach [12]. The final dataset, called tetragen, is composed of 22 attributes and 1188 objects corresponding to DNA sequences. Classes, i.e. the genomes *B. hyodysenteriae*, *C. acetoburylicum* and *B. cereus* contain respectively 288, 383 and 517 instances. In order to obtain the tetragen dataset, the largest DNA sequences were divided in several objects. We took benefit of this process to create label constraints: we assigned two DNA sequences composed of 13 and 21 objects in the subsets $\{B.\ cereus\}$ and $\{B.\ cereus, B.\ hyodysenteriae\}$ respectively. As a consequence, we obtained a dataset composed of 2.9% of constrained objects. Figure 3 presents the class and prior information used for the tetragen dataset.

Experimental Protocol: For both ECM and SECM, we performed 10 executions with random initialization of the centroids and kept the credal partition giving the minimum value for the objective function. To synthesize the information provided by the partitions, we transformed them into hard credal partitions by assigning each object to the subset of classes with the highest mass. Figures 4(a) and (b) illustrates the obtained results. As it can be observed, constraints helped SECM to impact the boundary of ω_3.

In order to compare the methods, partitions obtained with ECM and SECM were transformed into hard partitions by selecting the cluster with the maximal pignistic probability. Then, their agreement with the real partition were measured with the Adjusted Rand Index (ARI) [13] and the Normalized Mutual Information (NMI). Both of them provide a 1 value when the partitions totally

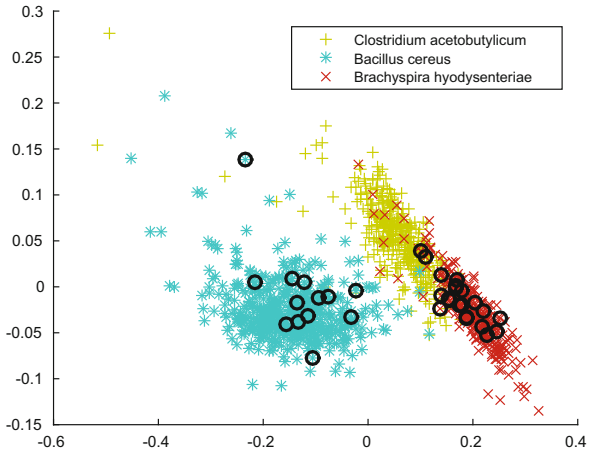


Fig. 3. Real classes (color) and constrained objects (encircled) for the tetragen data set. (Color figure online)

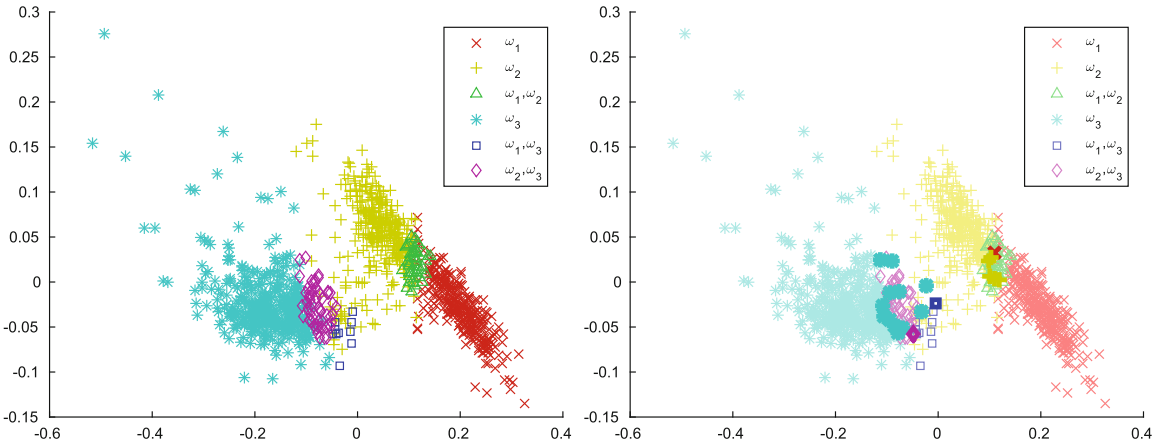


Fig. 4. Hard credal partition obtained with (a) ECM and (b) SECM for tetragen. Colors are lightened in (b) for objects for which the assignment has not changed between the two algorithms. (Color figure online)

match. With ECM, we obtained ARI = 0.75 and NMI = 0.71 whereas SECM gives an ARI = 0.78 and a NMI = 0.73. It shows that a few number of constrained objects, even partially labeled, can lead our clustering algorithm to a better result than ECM.

5 Conclusion

In this paper, a new semi-supervised clustering algorithm called SECM is proposed. It generalizes previous approach [7] based on partial label constraints. The new penalty term can be parameterized to favor either any credal partition for which constraints are still plausible or only credal partitions for which constrained objects have belief on subsets with low cardinalities. A proof of concept is provided and shows the benefits of the new algorithm. Finally, a real test

is performed on genomics data set and shows the necessity of such expressive approaches in real use case.

In the future, extensive tests on real and synthetic datasets should be conducted in order to show the influence of the parameter r and to compare various semi-supervised clustering algorithms. The genomics use case should also be developed as it offers a relevant testbed for partial user knowledge integration. A further work is to scale SECM for larger datasets, in order to apply the algorithm in a real genomics application.

References

1. Masson, M.H., Denœux, T.: ECM: an evidential version of the fuzzy c-means algorithm. Patt. Recogn. **41**(4), 1384–1397 (2008)
2. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
3. Denœux, T., Kanjanatarakul, O.: Beyond fuzzy, possibilistic and rough: an investigation of belief functions in clustering. In: Ferraro, M.B., et al. (eds.) Soft Methods for Data Science. AISC, vol. 456, pp. 157–164. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-42972-4_20
4. Basu, S., Davidson, I., Wagstaff, K.: Constrained Clustering: Advances in Algorithms, Theory, and Applications. CRC Press, Boca Raton (2008)
5. Antoine, V., Quost, B., Masson, M.H., Denœux, T.: CECM: constrained evidential c-means algorithm. Comput. Stat. Data Anal. **56**, 894–914 (2012)
6. Antoine, V., Quost, B., Masson, M.H., Denœux, T.: Evidential clustering with instance-level constraints for proximity data. Soft Comput. **18**(7), 1321–1335 (2014)
7. Antoine, V., Labroche, N., Vu, V.V.: Evidential seed-based semi-supervised clustering. In: Soft Computing and Intelligent Systems (SCIS), Kitakyushu, Japan, pp. 706–711. IEEE, December 2014
8. Smets, P., Kennes, R.: The transferable belief model. Artif. Intell. **66**, 191–234 (1994)
9. Bezdek, J.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenium Press, New York (1981)
10. Ye, Y., Tse, E.: An extension of Karmarkar's projective algorithm for convex quadratic programming. Math. Program. **44**(1), 157–179 (1989)
11. Pruitt, K., Tatusova, T., Maglott, D.: NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Res. **35**(Suppl. 1), D61–D65 (2006)
12. Alneberg, J., et al.: Binning metagenomic contigs by coverage and composition. Nat. Methods **11**(11), 1144 (2014)
13. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**(1), 193–218 (1985)

Liste des figures

Liste des tableaux

Liste des sigles

ACP Analyse en Composantes Principales. 14, 26, 34, 35, 46–49, 51, 52, 71, 90, 134, 135, 143, 189, 233

AMBER *Assessment of Metagenome BinnERs*. 43

ANI *Average Nucleotide Identity*. 41, 68, 69, 125–127, 137, 138, 144, 229

AP *Affinity-Propagation*. 25, 29, 34, 36, 189

BH-tSNE *Barnes-Hut t-distributed stochastic neighbor embedding*. 28, 31, 36, 52, 72, 98, 105, 107, 189

CAMI *Critical Assessment of Metagenome Interpretation*. 42, 43, 142

CLR *Centered Log Ratio*. 28

E-M Éspérance-Maximisation. 25–27, 34, 35, 55, 74, 78, 189

IUPAC *International Union of Pure and Applied Chemistry*. 70

kPLS *kernel Partial Least Square*. 35

MAG *Metagenome-Assembled Genome*. 21, 23, 24, 42, 69, 122, 123, 125, 138, 142, 144, 185, 230

NMF *Non-negative Matrix Factorization*. 25, 34, 189

SC simple copie. 34–36

SCG gène en simple copie. 23

SOM *Self Organizing Map*. 26

SVD *Singular Value Decomposition*. 35, 48

SVM *Support Vector Machine*. 31

VBGMM *Variational Bayesian Gaussian Mixture Model*. 26, 34, 35, 55, 73, 74, 94, 96, 104, 136, 143, 189

Table des matières