Building teaching tools for chemistry using chemoinformatics approaches

Louis Plyer

▶ **To cite this version:**

Louis Plyer. Building teaching tools for chemistry using chemoinformatics approaches. Theoretical and/or physical chemistry. Université de Strasbourg, 2024. English. NNT : 2024STRAF018 . tel-04767336

HAL Id: tel-04767336

https://theses.hal.science/tel-04767336v1

Submitted on 5 Nov 2024

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

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

UNIVERSITE DE STRASBOURG

ÉCOLE DOCTORALE DES SCIENCES CHIMIQUES

Chimie de la matière complexe – UMR 7140

THÈSE présentée par :

Louis Plyer

soutenue le : **03 octobre 2024**

pour obtenir le grade de : **Docteur de l'Université de Strasbourg**

Discipline/ Spécialité : Chimie informatique et théorique

Construction d'outils pédagogiques pour la Chimie par des approches Chémoinformatique

THÈSE dirigée par :

M. MARCOU Gilles	Maître de conférences, Université de Strasbourg

RAPPORTEURS :

M. LEPAILLEUR Alban	Maître de conférences, Université de Caen Normandie
M. SVOZIL Daniel	Professeur, University of Chemistry and Technology Prague

AUTRES MEMBRES DU JURY :

Mme M. CIVERA	Maître de conférences, Università di Milano
M. J. AIRES DE SOUSA	Professeur, Universidade NOVA de Lisboa
Mme E. KELLENBERGER	Professeure, Université de Strasbourg

INVITÉS :

Mme S. KENNEL	Maître de conférences, Université de Strasbourg
Mme C. PERVES	Ingénieure développement informatique, Université de Strasbourg

Acknowledgment

In this section, I would like to thank all of the people who helped me to achieve this thesis, both personally and professionally.

Firstly, I would like to thank my supervisor, Dr. Gilles Marcou for his supervision and guidance during my Ph. D, and for allowing me to complete this thesis.

I am also grateful to Prof. Alexandre Varnek and Dr. Olga Klimchuk for their constant support in all aspects of my presence in the laboratory. I want to specially thank Dr. Fanny Bonachera for her emotional and technical support, and for her door which has remained open all these years. I also want to thank Dr. Dragos Horvath for his help and advice on the genetic algorithms. Then, I would like to thank all of the members of the Laboratory of Chemoinformatics at the University of Strasbourg, present and past, and especially Dr. William Bort, Dr. Yuliana Zabolotna, Dr. Shamkhal Baybekov, Pierre Llompart, Erik Yeghyan, Regina and Karina Pikalyova and all of the others. I also want to thank Céline Perves for her help concerning Moodle development.

I want to thank the friends I made during my master's degree, for the unforgettable memories, discussions, and nights we had together. To Roman, Julia, Pablo, Tagir, Alexandra, Antoine, Marie and Marie.

I also want to thank all of the friends I've met on my chemistry journey, Cyprien, Camille, Alianna, Nina, Julien, Sajanthan, Frank, Paula, Lucie, Saad, Oussama, and all of the others.

I would like to thank my friends, who helped me enjoy these years. To Tom, Marc, Laure, Eloi, Camille, Marilou, Louise, Wuji, Emilien, Stan, Lodin, Anais, Thomas, Capucine, Tomy and all the others, the list being far too long.

Lastly, I would like to thank my family for their presence, Papa, Maman and Lili, thank you for your support during these years. Elise, Milo, and Hauru, I couldn't thank you enough for your patience and support during the most stressful situations, and for filling my heart. I also want to thank Ethel for her kindness and presence.

To all of these friends and to my family, I love you all and thank you again for being here!

Contents

1 Résumé en français

1.1 Introduction

Dans le contexte actuel marqué par l'essor des technologies de l'information et de la communication, l'enseignement à distance s'impose comme une composante essentielle de l'offre éducative. Cette mutation du paysage éducatif, accélérée par des circonstances imprévues telles que la crise sanitaire mondiale, souligne le besoin urgent d'outils pédagogiques adaptatifs et polyvalents. Ces outils doivent supporter une variété de formats pédagogiques, allant des cours magistraux traditionnels en passant par les cours inversés, à la préparation de travaux dirigés (TD) et travaux pratiques (TP), facilitant ainsi une approche plus flexible et inclusive de l'apprentissage.

L'introduction de systèmes de notation automatique représente une innovation majeure, permettant non seulement d'accroître significativement le volume de questions posées pour évaluer et soutenir l'apprentissage des étudiants, mais également de cibler efficacement le soutien pédagogique vers ceux qui en ont le plus besoin.

De plus, l’enseignement à distance repose naturellement sur des outils numériques pour faire face aux nombreux défis qui y sont associés ; entre autres, la personnalisation de l'enseignement et la gestion des retours faits aux étudiants. Par exemple, les questions posées pour un contrôle de connaissance peuvent être adressées à un étudiant en particulier et des éditeurs de texte enrichis facilitent l’approfondissement des discussions entre enseignant et étudiant. Les outils numériques dans les mains des enseignants soutiennent donc la qualité des enseignements et le niveau d’exigence. De surcroît , ils sont source de résilience car ils ont moins besoin d’être dimensionnés précisément au nombre d’étudiants, ce qui a permis l’émergence de nouvelles formes de cours tels que les MOOCs [1]. Et enfin, comme moyen technique pour l’enseignement à distance, ils assurent la continuité pédagogique et l'équité d'accès à l'éducation, même face à des évènements imprévus pour l’étudiant (e.g. accident) ou pour une promotion (e.g. pandémie).

Dans ce contexte, cette thèse présente le développement de différents outils intégrés dans le projet ChemMoodle. ChemMoodle regroupe un ensemble de plugins Moodle [2] innovants conçus pour l'enseignement et l'apprentissage de la chimie à distance. À travers l'introduction de plugins de type Atto et TinyMCE (section 2.1), de types de question spécifiques (Section 2.2, 2.3) et de ChemEngineering (2.4), un outil dédié au génie chimique, cette recherche explore les possibilités offertes par des technologies de l'éducation pour enrichir l'expérience d'apprentissage des étudiants en sciences chimiques. Pour finir, le générateur de banque de questions (Section 2.5) propose une méthode novatrice pour la génération automatique de ressources pédagogiques personnalisées, répondant ainsi aux besoins spécifiques des apprenants et des enseignants. La conclusion (Section 3) résume les principales contributions de cette thèse.

1.2 ChemMoodle

1.2.1 Insertion d'image de structures : MolStructure

Un premier plugin permettant d'insérer des dessins chimiques (structures ou réactions) dans n'importe quel type de questions et dans n'importe quelle zone de texte de Moodle a été développé (Figure 1-1).

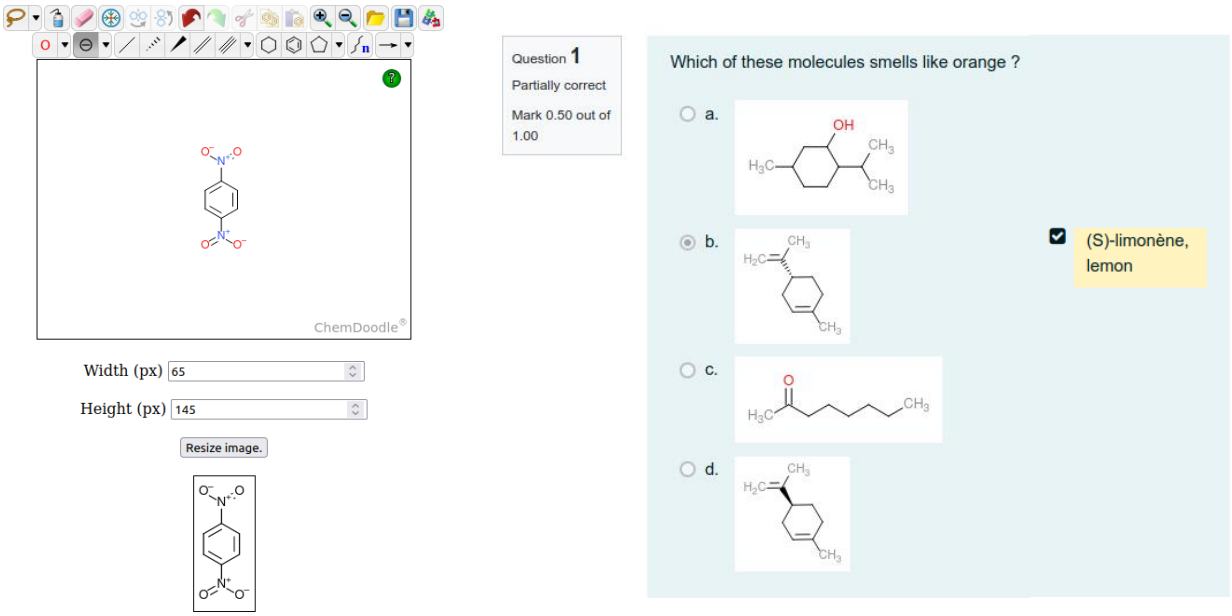

Figure 1-1. Interface d'édition (à gauche) et exemple de question à choix multiples utilisant le plugin atto (à droite).

Moodle est une plateforme d'apprentissage en ligne - ou Système de Gestion de l'Apprentissage (Learning Management System, LMS) - gratuite et open source, fonctionnant sous la forme d'un code source mère, autour duquel gravite une série de plugin implémentés, suivis et vérifiés par la communauté.

Les premières versions développées dans ce projet étaient au format d'un plugin Atto. Atto est un module JavaScript produit pour Moodle pour servir d'éditeur de texte standard. Il est aujourd'hui remplacé progressivement pour TinyMCE (Tiny Moxiecode Content Editor), un nouveau standard de module JavaScript utilisé comme éditeur de texte en ligne. La migration d'Atto à TinyMCE est dû à l'obsolescence du Framework YUI (Yahoo ! User Interface) sur lequel est construit l'éditeur Atto. Pour accompagner cette évolution de Moodle, un second plugin a été développé, fonctionnant avec l'éditeur TinyMCE.

Contrairement aux autres plugins du même type existant, il est totalement open source, ce qui permet à n'importe quel utilisateur de l'utiliser sans avoir à posséder une licence pour un logiciel tiers, comme MarvinJS [3] de Chemaxon [4] par exemple. L'éditeur MolStructure inclus également le support de représentations 3D de molécules et il peut aussi afficher des informations spectrales (masse, RMN, spectroscopie), en fonction des besoins du rédacteur.

Ce plugin est intégré à programme de maintenance logicielle et a déjà bénéficié de mises à jour.

1.2.2 Similarité moléculaire : MolSimilarity

Dans le domaine de l'éducation à distance, l'évaluation des dessins de structures chimiques représente un défi majeur. Les plateformes pédagogiques en ligne les plus évoluées reposent sur une comparaison exacte des réponses des étudiants et de la solution attendue. Mais cette approche ne permet pas une évaluation précise et nuancée, surtout lorsqu'il s'agit de questions impliquant des dessins de structures chimiques compliquées.

Pour aborder cette problématique nous avons développé un système innovant de questions auto-corrigées pour la chimie, intégrée à la plateforme Moodle. Il prend la forme d'un plugin de type « question type » et il prend en compte la complexité

induite par des questions sur les structures chimiques. En effet, la réponse se présente sous la forme d'un graphe. La correction doit donc être insensible à des variations légitimes (e.g. la numérotation des atomes). De plus les erreurs commises peuvent être plus ou moins sérieuses, conduisant à différencier l'évaluation pour des structures parfois très similaires à celle qui était attendue. Il peut donc arriver que l'enseignant n'ai pas intérêt à sanctionner de la même façon toutes les erreurs qui peuvent figurer dans la réponse proposée par l'étudiant. C'est pourquoi notre approche vise à fournir une évaluation fine (i.e. configurable) et douce (i.e. progressive), en prenant en considération la similarité entre les dessins de structures chimiques proposés par les étudiants et les structures attendues.

La flexibilité et le caractère progressif de la notation souple permet d'encourager les étudiants à adopter une approche d'apprentissage plus interactive. Il permet en particulier de mettre en œuvre des scénarios d'auto-apprentissage, en laissant l'étudiant revenir sur ses erreurs pour améliorer son résultat [5].

L'enseignant est également en mesure de proposer des solutions alternatives et des châssis moléculaires que l'étudiant doit modifier. Ceci permet de focaliser l'évaluation sur des éléments spécifiques de l'enseignement tel qu'il apparaît dans un graphe moléculaire (par exemple, le traitement correct de la stéréochimie).

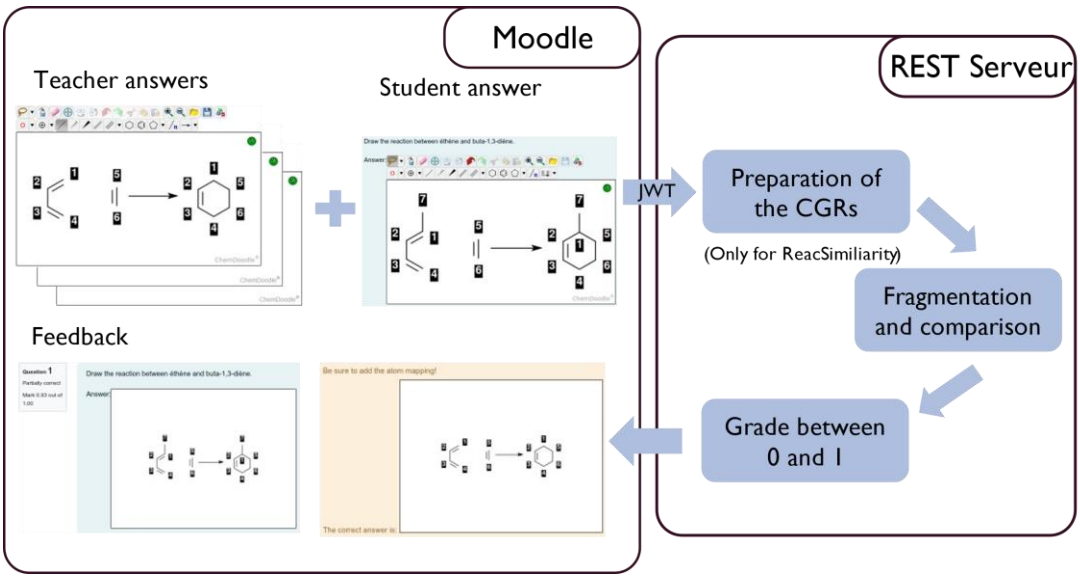

Figure 1-2. Processus de traitement d'une question quizz MolSimilarity

Une caractéristique importante de notre système est sa flexibilité. Les utilisateurs académiques peuvent facilement configurer le plugin en fonction de leurs

besoins spécifiques. Ils peuvent ajuster les paramètres d'intégration et de mesure de similarité pour s'adapter à différents contextes d'enseignement et types de questions.

Ce plugin utilise le moteur Chemdoodle [6] pour dessiner les structures chimiques et communique avec un serveur REST pour calculer le score de similarité en utilisant des descripteurs moléculaires ISIDA [7]. Le processus de traitement informatique de ces questions est résumé dans la Figure 1-2. Le module propose deux interfaces de rédaction : une pour l'enseignant et l'autre pour l'étudiant. Ces interfaces communiquent avec un serveur chargé de l'évaluation nuancée. Les résultats des calculs du serveur sont traités par Moodle pour produire une note et un retour commenté à l'étudiant. L'enseignant peut insérer plusieurs réponses considérées comme correctes, et la réponse de l'étudiant sera comparée à chacune d'entre elles.

En résumé, cette recherche propose une solution innovante pour évaluer les dessins de structures chimiques dans le cadre de l'enseignement à distance. Notre approche, intégrée à la plateforme Moodle, offre une évaluation plus précise et plus adaptée à la complexité des questions sur les structures chimiques, contribuant ainsi à améliorer l'expérience d'apprentissage des étudiants en chimie.

1.2.3 Similarité réactionnelle : ReacSimilarity

Comme pour les dessins de structures chimiques, des méthodes d'évaluation adaptées aux questions portant sur les réactions chimiques doivent aussi être crées. Dans cette optique, nous avons développé une nouvelle approche qui offre une solution aux limitations des méthodes de notation binaire traditionnelles. Cette méthode innovante vise à évaluer la compréhension des étudiants envers les réactions chimiques en prenant en compte une certaine tolérance dans les réponses.

Notre méthode repose sur une évaluation de similarité par paires de réactions chimiques. Pour ce faire, chaque réaction est encodée sous forme d'un graphe condensé de réaction (Condensed Graph of Reaction, CGR) [8–11]. Le CGR résulte de la superposition d'atomes de réactifs et de produits portant les mêmes identifiants (voir Figure 1-3), ce qui nécessite l'équivalence entre atomes (Atom to Atom Mapping, AAM)

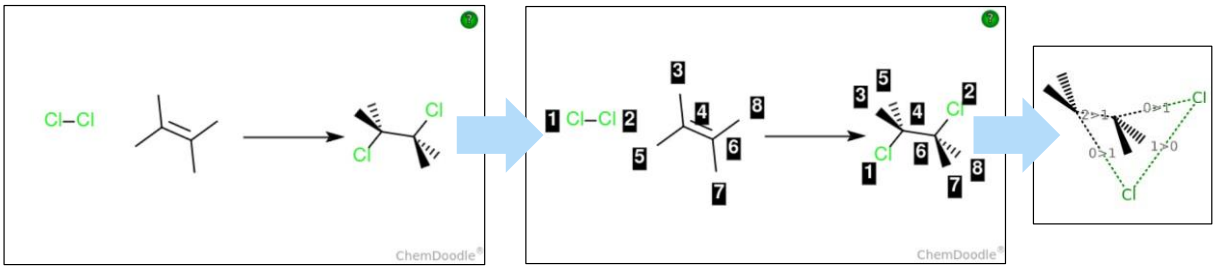

Figure 1-3. Processus de mise en correspondance des atomes : Réaction non appariée à gauche, appariement des atomes de la réaction au milieu et CGR associé à droite ("0>1" : création d'une liaison simple ; "1>0" : rupture d'une liaison simple).

Pour le dessin et la visualisation des réactions chimiques, notre méthode utilise le moteur Chemdoodle, qui a été modifié afin de mieux prendre en compte l'AAM. Ce moteur permet aux utilisateurs de créer facilement des représentations graphiques précises des réactions, facilitant ainsi la compréhension visuelle des concepts chimiques.

Enfin, notre méthode communique avec un serveur REST spécialement conçu pour calculer le score de similarité entre les réactions. Ce score est calculé grâce à l'utilisation de descripteurs de fragments ISIDA, générés à partir du graphe pseudo-moléculaire 2D du CGR.

En résumé, notre approche propose une méthode novatrice pour évaluer les questions portant sur les réactions chimiques dans le contexte de l'enseignement à distance. Elle combine des techniques avancées de représentation des réactions à des Système de Gestion de l'Apprentissage connus, offrant ainsi une solution facile d'utilisation et adaptable aux besoins des enseignants et des étudiants.

1.2.4 ChemEngineering

Lors de l'apprentissage de la chimie, un étudiant est amené à étudier des éléments de génie chimique, et donc, est amené à réaliser des schémas de procédés. Un schéma de procédé en génie chimique consiste à représenter les équipements de l'industrie chimique et leur relation afin de décrire un processus (ex : transformation chimique, distillation, etc.). Le schéma a pour but de donner une vision d'ensemble des équipements nécessaires au processus ainsi que de décrire les flux de matière.

Habituellement, les outils permettant le dessin de schémas de procédés disponibles sur le marché sont payants et non libres de droit, et aucun n'est présent sur des LMS. Pour répondre à cette problématique, nous développons une nouvelle solution : un plugin open source permettant l'insertion de schémas de procédés dans le contenu pédagogique, intégré à la plateforme Moodle.

Le développement de ce plugin a nécessité une approche collaborative. Nous avons rassemblé une équipe multidisciplinaire comprenant des experts en génie chimique, des développeurs web et des spécialistes de l'enseignement à distance. Ensemble, nous avons travaillé à l'identification des besoins des utilisateurs, à la conception de l'interface utilisateur et au développement des fonctionnalités clés.

À la suite de ce travail, nous avons élaboré un livre blanc. La mise en œuvre de ce document pour créer cet outil a été confié à l'association Ikigai Games for Citizens (l'association à but non lucratif, dont l'objectif est de développer des « jeux sérieux », i.e. des jeux vidéo à visé pédagogique). Notre équipe supervise les travaux (fonctionnalités, interface graphique) et intervient dans l'intégration de l'éditeur dans Moodle.

Ce plugin représente la première solution open source de ce type, développée entièrement en JavaScript, ce qui le rend compatible avec une large gamme de navigateurs web. En utilisant ce plugin, les enseignants en génie chimique peuvent facilement créer et intégrer des schémas de procédés dans leurs cours en ligne, offrant ainsi une expérience d'apprentissage interactive et immersive aux étudiants.

L'aspect open source de ce plugin est également un élément essentiel de sa conception. En publiant le code source sur GitHub, une plateforme de développement collaboratif, nous encourageons la communauté à se l'approprier et contribuer à son amélioration continue.

En effet, ce travail est géré sous la forme de deux bibliothèques hébergées sur GitHub, une pour le code de l'éditeur de schéma de procédé, et l'autre pour le plugin Moodle. Les utilisateurs sont encouragés à contribuer par leurs suggestions, des corrections de bugs et des fonctionnalités supplémentaires, ce qui permet d'assurer que le plugin reste à jour et réponde à de plus nombreux besoins des enseignants et des étudiants en génie chimique.

1.2.5 Question bank generator

La création de banques de questions pertinentes et variées représente un travail conséquent pour les enseignants. Traditionnellement, cette tâche repose sur une approche séquentielle dans laquelle chaque question est rédigée individuellement, y compris quand elles sont des variations autour d'un même thème. Cette méthodologie est naturelle quoique fastidieuse et peu gagner en efficacité.

Aussi, nous proposons de développer un outil innovant pour rédiger des questions sous la forme d'un concept qui est ensuite décliné sur des exemples chimiques concrets. Ces exemples doivent être choisis en fonction du contexte chimique.

Nous proposons d'exprimer ce contexte sous la forme de cartes de l'espace chimique, grâce à la technologie des cartes topologiques génératives (Generative Topographic Mapping, GTM) [12]. L'utilisateur peut alors choisir quelle zone d'un espace chimique utiliser pour créer des questions. L'objectif visé est de faciliter le processus de création de banques de questions par les enseignants, notamment dans les cas où il est nécessaire de générer des questions portant sur des types de molécules ou de réactions similaires. Ce dispositif vise à réduire la charge de travail associée à la conception manuelle de chaque question, en permettant une génération rapide de questions contenant des structures moléculaires similaires à une structure de référence donnée.

La GTM est une technique de réduction de dimensionnalité, permettant de passer d'un espace des composés chimiques à un espace latent en 2D, tout en préservant la relation de distance dans cet espace latent et la similarité entre les composés chimiques. Cette méthode permet ainsi de cartographier l'espace chimique, en transformant des données complexes en une représentation graphique 2D, plus simple et plus interprétable.

Dans ce segment du projet l'objectif est de préparer un outil pour aider les enseignants à identifier les zones les plus pertinentes pour générer des questions, de trois manières différentes :

La première, avec un ensemble de cartes déjà préparées et annotées, où les enseignants peuvent facilement cibler des zones de l'espace chimique en fonction de leurs besoins pédagogiques et des sujets qu'ils souhaitent aborder.

La seconde, où les enseignants peuvent saisir une structure chimique qui va être projetée sur la carte et utilisée comme point de départ pour générer des questions sur des analogues chimiques.

Enfin, la troisième où les enseignants peuvent apporter leurs propres jeux de données et utiliser la carte de leur espace chimique personnalisé. Cette nouvelle carte sera disponible pour l'enseignant aux côtés de celles proposées par défaut. Cette carte pourra être utilisée comme dans les deux scénarios précédents.

Le problème est que la GTM nécessite de choisir des valeurs pour des paramètres libres de la méthode, ce qui impose un savoir-faire pour générer ces cartes personnalisées. Pour répondre à ce défi, un nouvel algorithme génétique (Genetic Algorithm, GA) [13] a été implémenté, pour produire sans intervention experte des cartes de bonne qualité. Pour cette raison, de nouveaux outils logiciels ont été développé afin d'être simple à installer et à utiliser sur les architectures informatiques les plus courantes (Linux, Windows, Mac). Ce nouveau GA a été testé et validé en l'utilisant pour déterminer les méta-paramètres dans une variété de problèmes rencontrés au sein du laboratoire dans plusieurs thèmes de recherche.

En pratique, l'objet d'une carte est de proposer des structures chimiques analogues à celles qui sont situées dans une région délimitée par l'utilisateur. Les enseignants peuvent alors alimenter leur concept de question avec ces structures. La carte génère donc des structures chimiques et l'interface de l'enseignant génère des questions qui sont stockées dans des banques de question pour alimenter un LMS tel que Moodle.

1.3 Conclusion

Cette thèse explore l'importance croissante de l'enseignement à distance, en particulier à la lumière de la récente crise sanitaire mondiale mais aussi pour toucher des populations isolées ou dans l'incapacité de se déplacer sur un site d'apprentissage.

Dans ce contexte, cette thèse présente le développement d'outils intégrés dans le projet ChemMoodle, spécifiquement conçus pour l'enseignement et l'apprentissage de la chimie à distance. Ces outils, allant de l'insertion d'images de structures chimiques à la création automatisée de banques de questions en passant par l'évaluation automatique de la similarité moléculaire et réactionnelle, visent à enrichir l'expérience d'apprentissage des étudiants en sciences.

Cette thèse démontre comment ces outils numériques répondent aux évolutions de l'enseignement supérieur, en proposant des solutions innovantes et personnalisées pour soutenir l'apprentissage à distance.

Les perspectives de ces travaux se trouvent en particulier dans la diffusion de ceux-ci à un plus large public, notamment le secondaire. Par ailleurs, ce projet a émergé des résultats de recherche fondamentales en Chémoinformatique. Mais ces innovations pédagogiques contribuent en retour aux recherches fondamentales et à l'innovation, par exemple à travers le module ChemEngineering ou la diffusion auprès d'une très large audience des algorithmes génétiques les plus avancées. En ce sens, ce projet illustre le dialogue vertueux entre recherche et enseignement.

2 Introduction

2.1 General introduction

In an era marked by the rapid advancement of information and communication technologies, the educational offer is undergoing a profound transformation. Distance learning has become a pivotal component of the educational system, driven by both the need for modernization and unexpected global events such as the COVID-19 pandemic. Indeed, as universities closed, students and teachers had to turn to online courses. This shift underscores the urgent requirement for adaptive and versatile teaching tools that can support a variety of educational formats, including traditional lectures, flipped classrooms, and the preparation of practical work sessions. Furthermore, studies demonstrated that completing homework activities significantly impacts individual students' academic performance [14, 15], and those conducted online are viewed by students as useful learning tool [16]. Moreover, it was demonstrated that there is little to no difference in pass rates between online learners and traditional in-person students [17].

The integration of automated grading systems is a groundbreaking advancement in education [18]. These systems not only allow for a significant increase in the number of questions that can be posed to assess and support student learning but also enable precise targeting of educational assistance to those who need it most. Automated grading can swiftly identify student weaknesses and provide timely feedback, enhancing the overall learning experience.

Distance learning [19], by its nature, relies heavily on digital tools (e-learning) to address various challenges, including personalized instruction and effective student feedback management. For instance, tailored questions for assessments can be directed to individual students, and advanced text editors can facilitate deeper interactions between teachers and students. Additionally, these tools offer flexibility as they can be easily scaled according to the number of students, which has led to innovative course formats like MOOCs [1, 20, 21]. Finally, they ensure continuity of teaching and equity of access to education [22], even in the face of unforeseen events for the student or for a promotion.

E-learning can also help to address the limitations imposed by the increasing ratio of students to teaching staff, a reality that is becoming increasingly common in higher education institutions. In the last few years, there has been a significant decrease in the national budget per student, while the number of teaching staff recruited in universities has fallen and the number of students has increased year after year [23, 24]. The government does not adjust its higher education spending to account for student demographics and inflation. Instead, budget increases most often correspond to an augmentation in workload, not always related to teaching.

In the field of chemistry, the need for e-learning tools is particularly pronounced [25], as chemistry education relies heavily on the understanding and manipulation of chemical structures, which are traditionally conveyed through complex diagrams and hands-on laboratory experiments. As a result, a few solutions were developed, and tools proposed.

The SOCOT (Structure-Based Organic Chemistry Online Tutorials) platform [26] represents an early example of using a chemical structure sketcher in online tutorials with automated correction. Students' chemical structure drawings are converted into a canonical SMILES (Simplified Molecular Input Line Entry Specification) string [27]. These strings are then assessed by comparing them to an anticipated response, also in SMILES format. Other solutions leverage smartphones as the only hardware requirement [28]. The method involves students drawing chemical structures using JSME editor [29]. SMILES are generated, to be copied and pasted into ChemDrawJS [30] by the student, to generate an InChI code [31]. This code is to be copied by the student to a web-based classroom response system, Socrative [32]. It acts as concise, one-sentence answers that is instantly evaluated. Other learning tools were created to help students learn organic chemistry, as *Nomenclature101.com* [33], a platform containing quizzes and material to learn about nomenclature. OpenOChem is another platform allowing a teacher to ask chemistry related questions and containing practice problems and activities [34].

However, the current online tools may fall short in evaluating and transmitting these essential skills. Therefore, a need for new e-learning tools is there and needs to be addressed.

In this context, this thesis presents the development of various tools integrated into the ChemMoodle project. First, main cheminformatic concepts used are introduced (2.2). Then, the ChemMoodle project is introduced (3). It encompasses a set of innovative Moodle plugins designed for remote chemistry teaching and learning. By introducing Atto and TinyMCE plugins (3.1), specific question types (3.2, 3.3), and ChemEngineering (3.4), a tool dedicated to chemical engineering, this research explores the potential of educational technologies to enhance the learning experience of students in chemical sciences. Finally, the question bank generator (4) offers an innovative method for the automatic generation of personalized educational resources, thus meeting the specific needs of learners and educators. The conclusion (5) summarizes the main contributions of this thesis.

2.2 Introduction to the Chemical space, GTM

The following section introduces key cheminformatics concepts that are integral to the tools and methods developed for chemical education.

2.2.1 Molecular descriptors ISIDA Fragments

A good representation of the chemical data is needed in every chemoinformatic work to extract the underlying knowledge. Indeed, chemical data can be represented in multiple ways.

Molecular graphs are a popular representation of chemical data where the nodes are atoms, labeled with the bond type, and edges bonds, labeled with the bond type. This 2D representation captures most of the underlying information of a molecule: its connectivity and topology.

However, these graphs are often converted into a vector of molecular descriptors D to ease the representation and numerical encoding of structural information, allowing usage by chemoinformatic tools. There are 3 main types of molecular structure representation [35]:

1D molecular descriptors are calculated from the chemical formula, such as molecular weight or atom count.

2D molecular descriptors are calculated from the molecular graph, such as topological indices, or 2D fingerprints.

3D molecular descriptors are calculated considering the 3D conformation of the molecule, capturing the information about the spatial and geometry properties of the molecules.

We use 2D molecular descriptors, as the 2D representation of the molecules was sufficient in our work, and the speed of calculation of such descriptors is faster than 3D ones. To be more specific, 2D descriptors are expressing graph invariants of chemical structures - in general the Lewis structures. For instance topological indices that can be derived from a matrix representation: the *Wiener Index* [36], the *Zagreb indices* [37], or the Randić *branching index* [38]. An alternative concept is 2D fingerprints. For instance, hashed fingerprints (for instance the Chemical Hashed Fingerprints, CHFP [39]) are generated by enumerating all fragments corresponding to a specific definition, up to a certain size and recording them into a fixed size bit string by the mean of a *hash function.* Alternatively, the bit positions in the fingerprint can be

assigned a role in advance to record for the presence or absence of specific features, as in MACCS keys [40].

In this work, we use In SIlico design and Data Analysis (ISIDA) Substructure Molecular Fragments (SMF) [7, 10, 41] and Fuzzy Pharmacophore Triplets [42], which are part of the ISIDA software package [43].

ISIDA SMF descriptors depict the molecular structure by counting the occurrence of subgraphs within the molecule. The algorithmic definition of a subgraph corresponds to a particular type of descriptor, and the number of times a subgraph appears in the molecule determines the value of that descriptor (Figure 2-1).

Four kinds of fragmentation are possible within the ISIDA software:

- Atom Count: Substructural fragments of length 1, accounting for the occurrence of atoms in the molecular structure.
- Atom centered fragments: Substructural descriptors composed of the central atom, and its neighbors. Also called augmented atoms (depending on the number of neighbors).
- Sequences: Linear sequences of atoms and/or bonds. Shortest possible path between each pair of atoms.
- Triplets: All possible combinations of 3 atoms with the corresponding topological distance indicated.

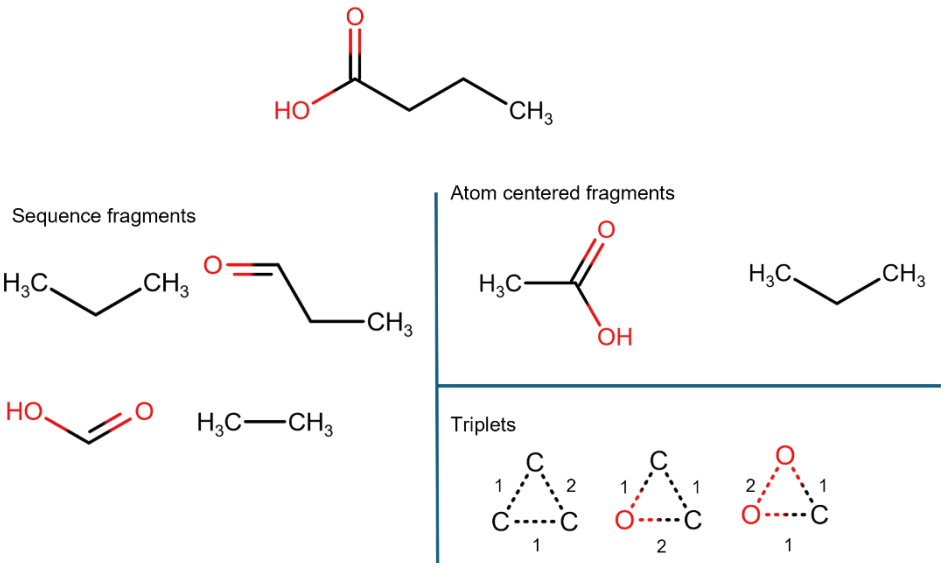

Figure 2-1. Example of fragments of the butanoic acid using ISIDA fragments.

Depending on the definition of the molecular descriptor, bond types (B), atom types (A) or both (AB) can be recorded in the fragment. The length of the descriptor is

also configurable. The descriptors can incorporate graph annotations: the formal charge, chiral labels, being member of a cycle or custom annotation. The definition of the fragments can be exhaustive or focused on shortest topological path only.

ISIDA SMF are calculated on the molecular graph, where its nodes can be colored by different means, such as pharmacophore types, electrostatic potentials, or Benson atoms.

2.2.2 Chemical/Descriptor space

The primary task of Chemoinformatics is the exploration of the Chemical Space (CS). Molecules are represented by molecular descriptors vectors, which embed them in a highly dimensional space (Figure 2-2). The embedding space has as many dimensions as the molecular descriptors vectors.

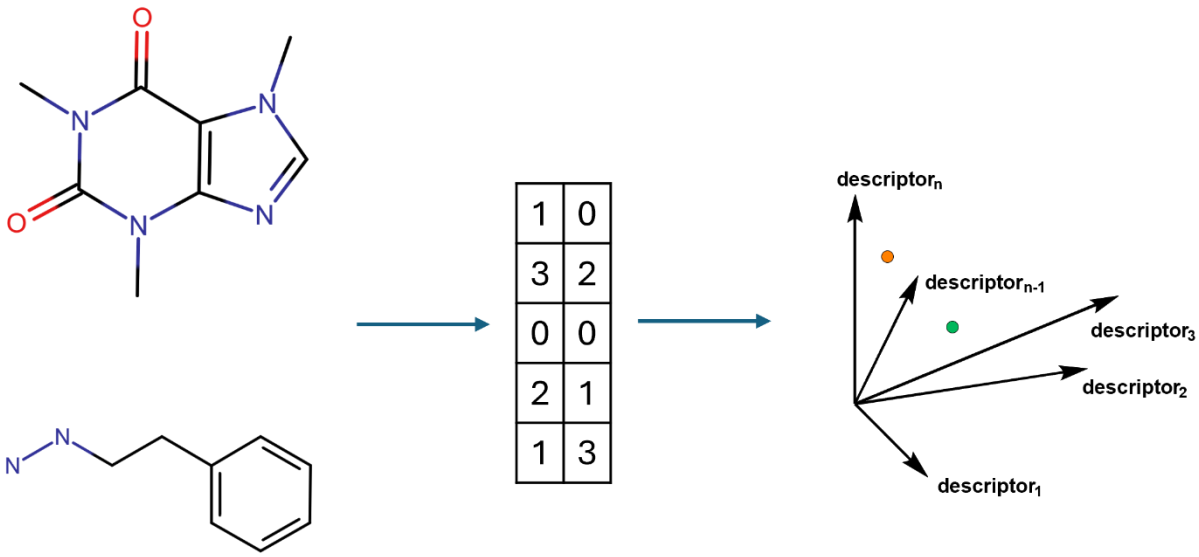

Figure 2-2. Depiction of the process of representing molecules in the Chemical space.

Therefore, in Chemography [44], which is the part of chemoinformatics dedicated to CS navigation, it is crucial to identify the set of descriptors that accurately represent the data set. Indeed, before applying any data reduction technique or any modeling attempts, it is essential to find the descriptor space (DS) that best represents the data, and respect the neighborhood principle [45]. Without an appropriate DS, even a modellable data set cannot be effectively visualized or used to build strong machine learning models [46]. The Chemical space is the central concept to model an assembly of chemical structures. This is used, for instance, for Quantitative Structure-Activity Relationship (QSAR) and Quantitative Structure-Property Relationship (QSPR) models, to prioritize compounds in virtual screening and for data visualization.

The role of QSAR/QSPR is to uncover the relationship between a compound's property or activity P and its representation D, by the mean of descriptors : P=f(D) [47].

The QSPR methodology involves searching for a mathematical model able to correlate a molecule's activity or property with its structure, as described by molecular descriptors. Numerous methods exist and can be used to build these models, such as SVM (2.2.5) [48], random forests [49], GTM (Generative Topographic Mapping, 2.2.4)

[12], k-Nearest Neighbors [50], deep neural networks (Deep QSAR) [51], or Naive Bayesian classifiers [52].

These models are based on the principle that a molecule's property is a smooth function of its structure, meaning small changes in the descriptors values should lead to small changes of properties [45, 53].

Numerous methods have been developed for CS visualization and exploration. The primary challenge in visualizing high-dimensional data lies in representing it in human-readable dimensions with no information loss or focusing on the most relevant information. It often relies on various dimensionality reduction methods starting from the high-dimensional molecular descriptor space to a human readable one. Instances of such techniques are Principal Component Analysis (PCA) [54], Self-Organizing Maps (SOM) [55], GTM [12] or t-Distributed Stochastic Neighbor Embedding (t-SNE) [56].

PCA is a linear dimensionality reduction technique that transforms data into a new coordinate system using a set of uncorrelated variables or principal components. These components are chosen to explain as much variance as possible and are ranked in descending order, with PC1 explaining the most variance.

SOM is an artificial neural network organized on a bidimensional manifold. The training is unsupervised producing a low-dimensional representation of the input space. The training is based on a winner-takes-all policy so similar data points are attached to one artificial neuron. As a result, the artificial neurons have an image in the input space located in the center of the tessellation cell containing the data points it represents. It is a valuable tool to visualize complex data and can even be used for prediction tasks if well-parametrized.

t-SNE is another nonlinear dimensionality reduction technique. It focuses on the similarities between pairs of compounds in the input space that are mapped to distances in a 2D (or 3D) space. The input distances are mixtures of gaussian radial basis functions, and the output distances are regularized multiplicative inverse of the squared Euclidean distances in the output space. Input and output similarities are normalized in order to be used as probability distributions. The output images of the input data points are then adjusted to minimize the Kullback-Leibler divergence of the resulting input probability distribution from the output probability distribution

GTM aims to extend the concept of SOM using fuzzy logic. The distribution of the molecules in the input space is modelled using a manifold centered normal distribution. As a result, a molecule is no longer represented by a single artificial neuron but by a population of responses proportional to the proximity of the molecule to elements of the manifold, the nodes of the map. To each molecule corresponds a vector of value of residence at each node on the map, which are called responsibilities. New compounds can be projected onto an existing map without retraining. The training can be operated on representative subsets of a training dataset which is computationally effective. More details are given later (2.2.4).

2.2.3 Evaluation metrics.

2.2.3.1 Classification

In classification tasks and particularly in multi-class scenarios, various metrics are used to evaluate a model's performance: Sensitivity (also called Recall or True Positive Rate (TPR), (2-1)), Specificity (True Negative Rate (TNR), (2-2)), Accuracy (2-3), and Balanced Accuracy (BA, (2-4)) [57].

Sensitivity quantifies the model's ability to identify positive instances. Specificity assesses the model's capability to identify negative ones. They range from 0 to 1, with higher values indicating better performance in identifying positive and negative instances. Accuracy measures the proportion of correctly predicted out of all instances and ranges from 0 to 1, where 1 indicates perfect classification and 0 indicates no correct predictions. Balanced Accuracy is the arithmetic mean of sensitivity and specificity. Therefore, it's a metric that provides a more comprehensive and equitable measure of a model's effectiveness, particularly in datasets with uneven class distribution.

The equations below define TP_i as true positive, TN_i as true negative, FP_i as false positive, and FN_i as false negative, for each class i. against all other classes. k is the number of classes.

$$Sensitivity(TPR) = \frac{TP_i}{TP_i + FN_i} \tag{2-1}$$

$$Specificity(TNR) = \frac{TN_i}{TN_i + FP_i} \tag{2-2}$$

$$Accuracy = \frac{1}{k}\sum_{i=1}^{k} \frac{TP_i}{TP_i + TN_i + FP_i + FN_i} \tag{2-3}$$

$$Balanced\ Accuracy\ (BA) = \frac{1}{k}\sum_{i=1}^{k} \frac{Sensitivity_i + Specificity_i}{2} \tag{2-4}$$

2.2.3.2 Regression

In regression tasks, it is standard to measure a Coefficient of Determination (R^2) and Root Mean Square Deviation (RMSD) to evaluate model performance. R^2 (2-7) is the metric most commonly used to estimate the descriptive quality of a model [57]. These metrics help quantify the variance explained by the model and the variance that remains unexplained.

The Root Mean Square Error (RMSE, (2-8)), quantifies the average magnitude of the errors between predicted and observed values. It is expressed in the same unit as the dependent variable, making it easy to interpret in the context of the problem. The formulas are given below, where x_i, y_i are the predicted and observed values, $\bar{y}$ is the mean of the observed values, and n is the number of data:

$$SST = \sum_i (y_i - \bar{y})^2 \tag{2-5}$$

$$SSR = \sum_i (y_i - x_i)^2 \tag{2-6}$$

$$R^2 = 1 - \frac{SSR}{SST} \tag{2-7}$$

$$RMSE = \sqrt{\frac{1}{n} SSR} \tag{2-8}$$

2.2.4 Generative Topographic Mapping

Generative Topographic Mapping (GTM) [12] effectively maps the chemical space, converting complex data into a simpler and more interpretable 2D graphical representation, analogous to a geographic map. Specifically, it is a nonlinear dimensionality reduction technique projecting from the chemical space onto a 2D latent space. It can be understood as a probabilistic extension of SOM. The distance relationships in this latent space and the similarity between chemical compounds in the input space are preserved in the probability measure of the GTM, achieving an intuitive representation.

The map results from a probability model of the dataset. The GTM fits a normal distribution centered on a manifold, maximizing the likelihood of the training dataset. The manifold is a finite 2D surface defined by a linear combination of Gaussian Radial Basis Functions (RBFs, Figure 2-3). The optimization procedure adjusts the position of the manifold in the input space and the width of the normal probability distribution function. The training dataset is subsampled in a frameset to save computational time. The frameset is the actual dataset used to train the GTM: it is of smaller size compared to the initial training set while representative of the training dataset.

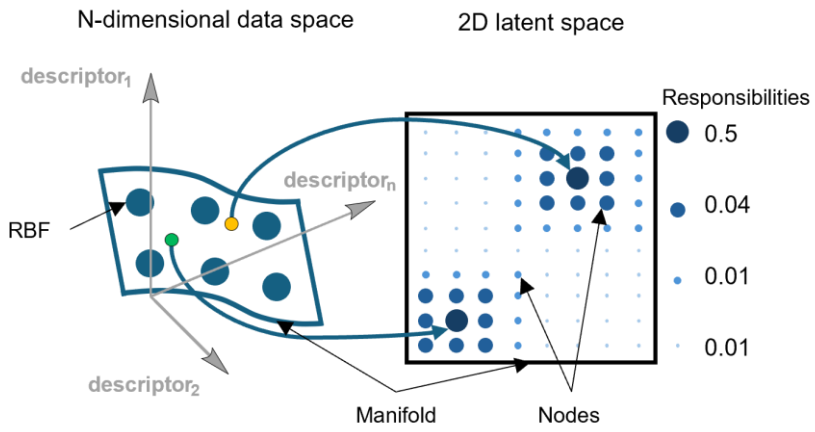

Figure 2-3. Concept of the GTM. Molecules are projected from the chemical space to a 2D latent space, with value of residency to each node, called responsibilities. The sum of all responsibilities for one molecule is equal to 1.

The manifold is fitted to align more closely with the frameset items by maximizing the Log-Likelihood (LLh) value. It represents the probability of a compound as explained by the manifold-centered probability distribution. The probability calculation is discretized over a sampling of the manifold, the nodes of the manifold. All nodes of the manifold do not contribute equally to the likelihood: those nodes closer

to the compound contribute more. The contribution of each node is termed the responsibility. Nodes are used to sample the manifold at defined positions and can be seen as grid points.

The responsibility vector defines the degree to which each compound is associated with each node. The components of a responsibility vector are summing to 1. The responsibility vectors are the central ingredient to generate GTM-based QSAR and QSPR models. The models are used to color the map, producing a classification or regression *landscape*. This coloration is achieved by assigning each node of the manifold a property value, calculated as the responsibility-weighted average of the properties of all compounds associated with that node.

The number of RBFs is the main parameter of the GTM since it controls the number of degrees of freedom of the model. The width of the RBF and the regularization coefficient are two other parameters controlling the expressivity and the overfitting of the GTM model. The number of nodes is a neutral parameter: it has almost no impact on the quality of the model: a large number of nodes slows down the calculation but produces high-resolution maps, while a low number of nodes reduces computational cost but decreases the details in the generated map. However, the number of nodes can always be increased a posteriori.

GTM has been successfully applied in chemoinformatics. It has been used for representing big-data libraries using incremental GTM [58], as for Gaspar et al. [59], where it was applied to a database of over 2 million compounds, integrating datasets from 36 chemical suppliers and the NCI collection. It was also proved that even the largest libraries may be represented by only a small frameset, in a study where parallel GTM was introduced [60]. Universal GTM was introduced, to gather in a single map information about multiple ligands [61].

GTM was also utilized for understanding compound activities by scaffold analysis [62, 63]. It has been used to diversify existing chemical libraries by combining hierarchical GTM [64] and Maximum Common Substructure (MCS) comparison to create *AutoZoom* [65]. Lately, DNA-encoded libraries were mapped and analyzed by the mean of GTM [66], and later, the chemical library space was defined [67] and analyzed using meta-GTM [68].

2.2.5 Support Vector Machines

Support Vector Machine (SVM) is a supervised learning algorithm initially designed for binary classification scenarios. First introduced in 1965 by V. Vapnik [69], more efficient implementations were proposed since 1995 [48]. C-SVM, for classification, operates by constructing models that classify samples into distinct classes using support vectors, which define a hyperplane. The optimal hyperplane is defined based on examples of the training set termed support vectors. The training of the model implements the concept of soft margin. It is a balance between a criterion of the largest margin, representing the maximum distance between the hyperplane the nearest data points from each class, and the number of classification errors. The hyperparameter cost (*C*) determines the trade-off between maximizing the margin and minimizing classification errors. A high *C* value penalizes misclassifications more heavily at the cost of a smaller margin, making it prone to overfitting. A low *C* value allows for a wider margin with more tolerance for errors at the cost of training accuracy.

By identifying the hyperplane with the maximum margin, SVM aims to enhance generalization to unseen data and minimize classification errors. The SVM can be expressed using kernel functions. These functions reformulate the initial problem in a feature space, a formal vector space that never has to be explicit. The kernel formulation of SVM has no limits in its expressivity and hence is a method of choice to solve non-linearly separable problems (2-9) in the initial n-dimensional space (Figure 2-4). Common kernel functions are gaussian radial basis function (RBF, (2-10)), polynomial (2-11), and sigmoid kernels (2-9).

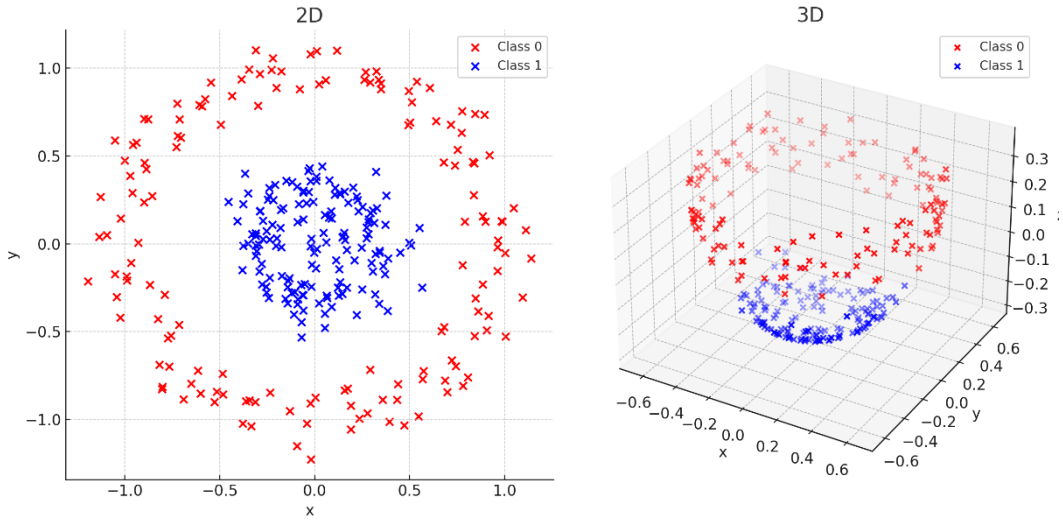

Figure 2-4. Use of a Kernel function, allowing for linear separation of the two classes.

The kernel often introduces a supplementary free parameter to choose, termed gamma (γ). It determines the resolution of the kernel, translating how far the influence of a single support vector reaches, with a small gamma value implying a far-reaching influence and a high gamma value indicating a close-reaching one. A systematic bias can be set and is conventionally noted $coeff_0$.

$$K(x_i, x_j) = x_i, x_j \tag{2-9}$$

$$K(x_i, x_j) = \exp\left(-\gamma \left\| x_i, x_j \right\|^2\right) \tag{2-10}$$

$$K(x_i, x_j) = (\gamma \langle x_i, x_j \rangle + coeff_0)^d \tag{2-11}$$

$$K(x_i, x_j) = \tanh\left(\gamma \langle x_i, x_j \rangle + coeff_0\right) \tag{2-12}$$

Where:

$x_i$and x_j are the feature vectors of the *i*-th and *j*-th data points.

$\left\| x_i, x_j \right\|$ is the Euclidean distance between x_i and x_j.

$\langle x_i, x_j \rangle$ is the dot product (inner product) of x_i and x_j.

γ is the parameter controlling the influence of a single training example.

$coeff_0$is a constant term.

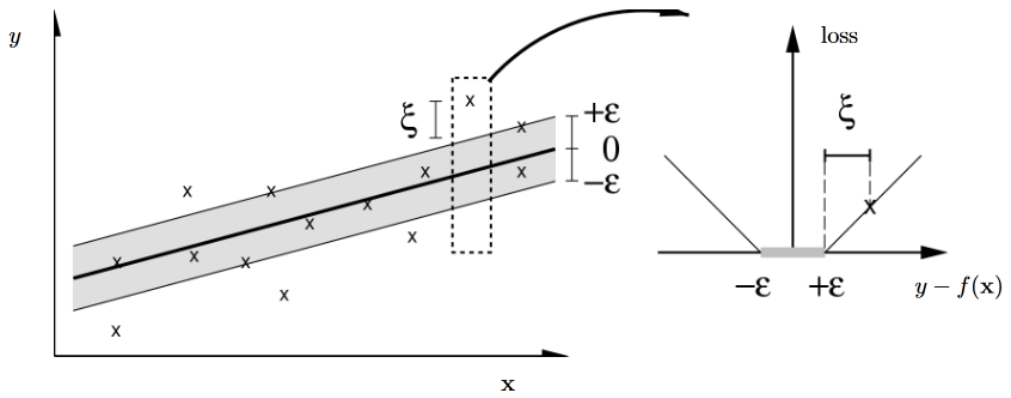

Figure 2-5. The soft margin ε insensitive tube, and penalty ξ. Taken from Schölkopf and Smola, 2002 .

Support Vector Regression (SVR) [70] extends the principles of SVM to regression problems. ϵ-SVR aims to predict continuous target values. This is achieved by defining a cost function that is insensitive to errors in range $[0, \epsilon]$. The solution to the problem is defined as a linear combination of training set instances termed the support vectors. Errors within this $[0, \epsilon]$ margin are not penalized, while errors exceeding this margin are penalized by a penalty ξ that increases linearly with the error (Figure 2-5, [71]). The hyperparameter *C* controls the trade-off between the complexity of the model -as measured by the number of support vectors- and the extent of permissible error.

2.3 Critical review of Genetic Algorithm

Genetic Algorithms (GAs) are a metaheuristic, a class of optimization algorithms that falls within the broader category of evolutionary algorithms (EA) [72], inspired by the principles of natural selection and genetics (Figure 2-6). GAs imitates the process of evolution to search for optimal solutions to complex problems. They have been widely applied across various domains, including engineering [73], economics [74], biology [75], medicine [76], and chemoinformatics [77].

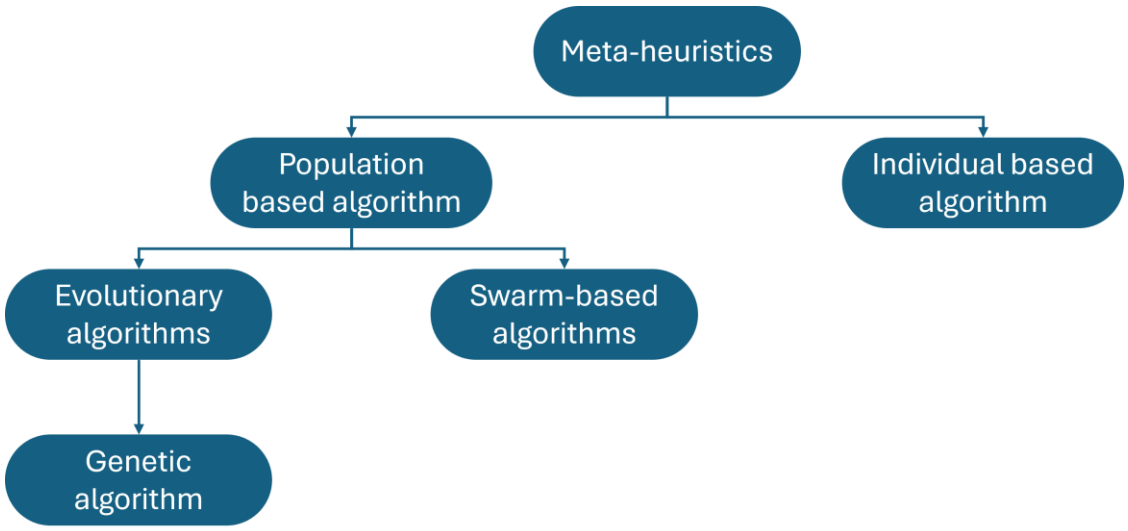

Figure 2-6. Classification of Meta-heuristic algorithm.

At the essence of a genetic algorithm is a population of potential solutions, often referred to as individuals or *chromosomes*. Each chromosome represents a candidate solution to the optimization problem and is typically encoded as a string of binary digits or real-valued numbers. A chromosome is of N dimensions, each dimension representing a parameter to optimize, and called a *gene* (Figure 2-7).

Each chromosome is associated to a score or *fitness score* (FS), measuring the efficacy of the solution encoded by the chromosome. The population undergoes iterative generations, where new individuals are created through processes such as crossover (recombination of genetic material) and mutation (random modification in genetic material). The process goes on until one stopping criteria is reached [78], which can be a maximum number of generations, maximum number of chromosomes created, convergence analysis [79], number of generations without amelioration of best individual, value of fitness function reached

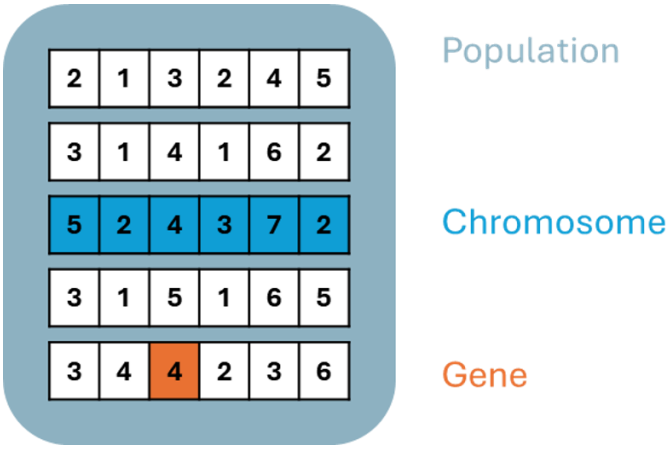

Figure 2-7. A GA is based on three nested concepts. The population is a set of solution under consideration for a given optimization problem. Individual solutions are termed chromosomes. Each chromosome is composed of individual values being part of a solution to the optimization problem.

Genetic algorithms offer several advantages, including their ability to explore large search spaces, handle complex, non-linear optimization problems, and find globally optimal or near-optimal solutions. The are almost guaranteed for instance to find the optimal solution to complex non-linear problems with an arbitrary number of local optimums, degenerated or not. However, they also have limitations, such as computational overhead, sensitivity to inner parameter settings, and the potential for premature convergence to suboptimal solutions [80]. For instance, the guarantee mentioned above may be met only after infinite time. To address these limitations, strategies need to be developed that optimize computational efficiency [81], fine-tune parameter settings, and implement mechanisms to prevent premature convergence [82], by preserving the diversity in the population [83]. Additionally, the trade-off between exploration and exploitation of each solutions/chromosome was addressed in the literature, with diverse strategies including selection methods [84–86] or adaptive values of crossover and mutation rates [87, 88].

In Chemoinformatics, genetic algorithms have emerged as powerful tools for solving various problems in drug discovery, molecular modeling, and chemical optimization. GAs can be used for lead optimization and de novo drug design, by modifying ligand structures to improve binding affinities [89]. They can also be used to optimize ligands in a receptor's pocket [90]. They can generate diverse sets of molecular structures by reducing costly evaluations and enhancing molecular property sampling, as demonstrated by JANUS [91]. GAs combined with machine learning were shown to effectively design antimicrobial peptides by generating diverse sequences with targeted properties [92].

They are also widely used for parameter selections in QSAR studies, as presented in Horvath et al. work [77], which was applied in numerous studies, as for parameter optimization on GTM for the creation of multi-target competent maps [61] or for the parameter optimization on QSAR models estimating acute toxicity [93]. Furthermore, GAs have been used for QSAR study, selecting significant descriptors to identify key features in VEGFR-2 inhibitors [94].

GAs are also used for molecular docking and ligand-receptor binding prediction, as with AutoDock, which implements a Lamarckian GA [95], or with GOLD which searches the space of available binding modes [96, 97].

GAs are employed in pharmacophore modeling by generating and refining pharmacophore hypotheses to create overlays of multiple ligands with GAPE and GALAHAD [98, 99].

To summarize, genetic algorithms are used in various fields, including economics, engineering, and chemistry. In the chemoinformatic field, they offer a flexible and efficient approach for molecular optimization, de novo design, docking, and structure-activity relationship analysis.

3 ChemMoodle

In this chapter, we introduce a suite of tools developed as part of the ChemMoodle project, aimed at facilitating the learning of chemical skills within the Moodle learning platform. In this project, we are offering three new Moodle modules that can be installed independently.

The first module is an Atto plugin called "MolStructure" This module adds a button to the Moodle text editor toolbar, allowing users to access a chemical sketcher. This feature enables instructors and content creators to insert chemical structures or reaction drawings into questions or learning resources. The plugin has been updated to a TinyMCE plugin.

The second module, "MolSimilarity", introduces a graded scoring system to evaluate students' responses when they draw chemical structures automatically. This method assigns a score to the response based on the graph similarity between the student's answer and the correct solution.

The third module, "ReacSimilarity", incorporates the concept of CGR. When a response to a question involves a chemical transformation, it is represented by a pseudo-molecular graph (the CGR) where modified bonds and atoms are labeled. This representation allows for a graded scoring system similar to the MolSimilarity plugin.

ChemMoodle is implemented as a series of Moodle plugins, utilizing the ChemDoodle engine for drawing chemical structures. The graded scoring engine is an open-source REST server provided with the plugins, which can be installed and managed locally.

Finally, “ChemEngineering” is an open-source process flow diagram sketcher under development. It will allow to integrate these processes into Moodle courses, enhancing the learning experience in chemical engineering.

Documentation aimed at teachers for the creation of questions for the two question type plugins may be found in Appendix 1: Manual of the question type .

3.1 MolStructure

A plugin has been developed that allows users to insert chemical drawings (structures or reactions) into any type of question or text area in Moodle (Figure 3-1) [100].

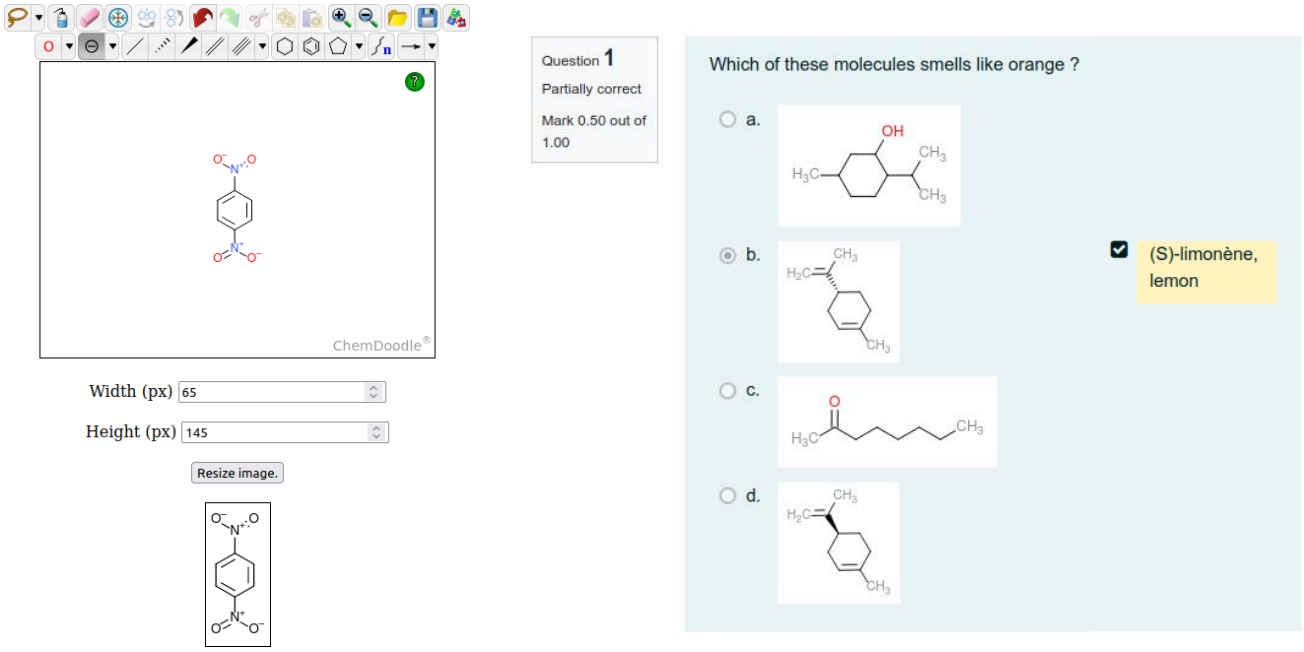

Figure 3-1. Editing interface (left), and example of a multi choice question making use of the atto plugin (right).

The plugin was initially developed as an Atto plugin. Atto is a JavaScript module created for Moodle, serving as the standard text editor. However, it is being gradually replaced by TinyMCE (Tiny Moxiecode Content Editor), a new standard JavaScript module used for online text editing. The migration from Atto to TinyMCE is due to the obsolescence of the YUI (Yahoo! User Interface) framework [101] on which the Atto editor is built. The obsolescence of YUI can be attributed to its lack of ongoing updates and diminished popularity, particularly as newer frameworks, such as React [102] and Angular [103], offer enhanced functionality, security, and alignment with evolving web development practices and browser standards. TinyMCE is a popular online text editor used in many web applications. It offers a "What You See Is What You Get" (WYSIWYG) editing experience, allowing users to format text, insert images, and manage complex content easily within a web browser. It is open-source, platform-independent, and highly extensible, with numerous plugins available to enhance its functionality. To align with this evolution of Moodle, a second plugin has been developed to work with the TinyMCE editor [104] in collaboration with Céline PERVES.

The plugin is developed using the ChemDoodle [6] open-source solution as a basis. The code has been re-worked in depth to add functionalities, and since the original material contains functionalities operated by third party closed source services. It is also much more open to contributions. For this reason, in contrast to other similar existing plugins, this one is entirely open-source, allowing any user to utilize it without needing a license for third-party software, such as MarvinJS [3] from Chemaxon [4]. MarvinJS is a chemical editor developed by Chemaxon for drawing, viewing, and characterizing chemical structures, often used in scientific and educational settings. While MarvinJS is highly functional, it is a commercial product requiring licensing, which limits its accessibility for many applications.

The *MolStructure* editor also supports 3D representations of molecules and can display spectral information (mass, NMR, spectroscopy) according to the user's needs. The plugin is integrated into a software maintenance program supported by the University of Strasbourg and the Laboratoire de Chémoinformatique (UMR 7140) and has already benefited from an update.

This development represents a significant step forward in integrating specialized scientific tools into online learning platforms. The move from Atto to TinyMCE aligns with current trends in Moodle software development, emphasizing the importance of maintaining up-to-date frameworks. By opting for an open-source model, this plugin aligns with the Moodle community's preference for accessible and transparent tools. The functionality for 3D molecular representation and spectral information showcases the plugin's capacity to answer to advanced scientific needs, thereby enhancing the utility of Moodle as a platform for scientific education.

Moreover, the inclusion of this plugin in a software maintenance program shows a commitment to ongoing improvements and updates, which is crucial for the long-term sustainability and relevance of such tools in a rapidly evolving technological landscape. This commitment benefits both educators and students by providing valuable resources and contributes to the broader field of online learning with specialized, high-quality educational tools.

At this date, the *MolStructure* Atto plugin has been downloaded 394 times in the last year and is being used on 79 sites. (released in August 2022). As we can see on Figure 3-2, the number of sites using this plugin is increasing each month.

The *MolStructure* TinyMCE plugin has been downloaded 88 times and is being used on 29 sites (released in April 2024).

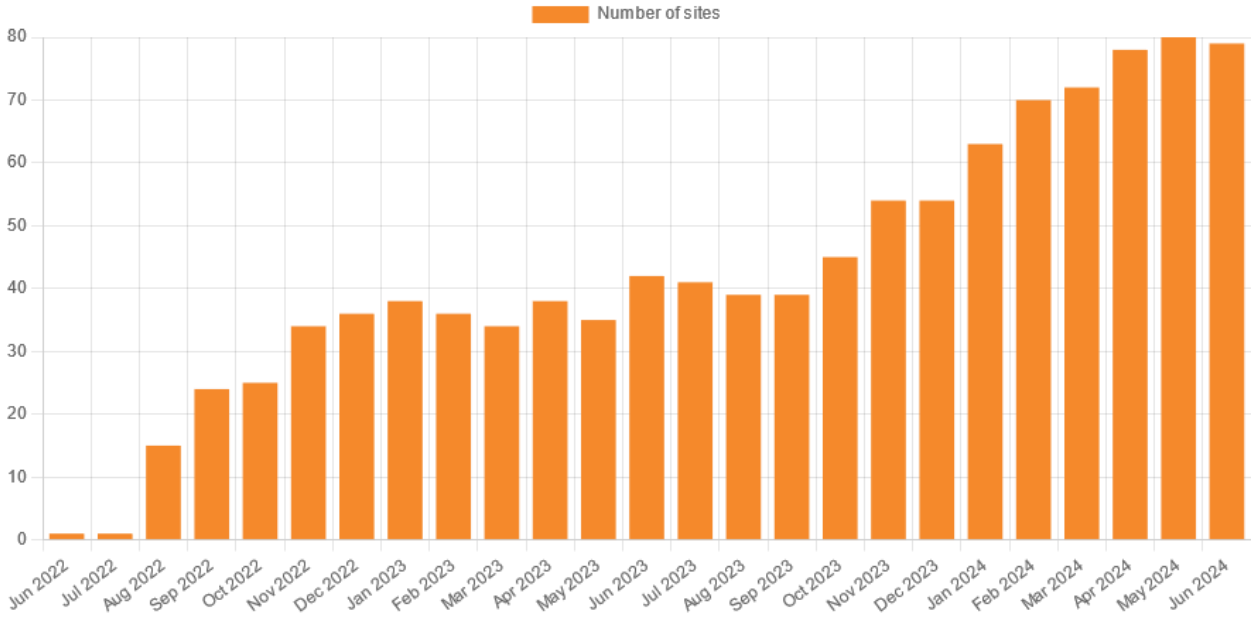

Figure 3-2. Number of sites using the MolStructure Atto plugin from its submission on the Moodle platform up to June 2024 (data from Moodle plugin repository).

3.2 MolSimilarity

3.2.1 Introduction

In the field of distance education, evaluating drawings of chemical structures presents a major challenge. The most advanced online educational platforms rely on an exact comparison of students' responses with the expected solution [26, 28], SMILES [27], or InChi [31] code. However, this approach does not allow for precise and nuanced evaluation, especially when it comes to questions involving complex chemical structure drawings.

To address this issue, we have developed an innovative system of auto-correcting questions for chemistry, integrated into the Moodle platform. It takes the form of a "question type" plugin and accounts for the complexity introduced by questions about chemical structures. Indeed, the response is presented as a graph. Therefore, the correction must be insensitive to legitimate variations (e.g., atom numbering). Additionally, the errors made can vary in severity, leading to differentiated evaluation for structures that may be very similar to the expected one. It may happen that the teacher does not wish to penalize all errors in the student's response equally. This is why our approach aims to provide fine-tuned (i.e., configurable) and gentle (i.e., progressive) evaluation, considering the similarity between the chemical structure drawings proposed by the students and the expected structures.

The flexibility and progressive nature of the flexible grading system encourage students to adopt a more interactive learning approach [105]. It particularly allows for the implementation of self-learning scenarios, enabling students to revisit their mistakes to improve their results.

The teacher is also able to propose alternative solutions and molecular frameworks that the student must modify. This allows the evaluation to focus on specific elements of the teaching as they appear in a molecular graph (for example, the correct treatment of stereochemistry).

An important feature of our system is its flexibility. Academic users can easily configure the plugin according to their specific needs. They can adjust integration and similarity measurement parameters to suit different teaching contexts and types of questions.

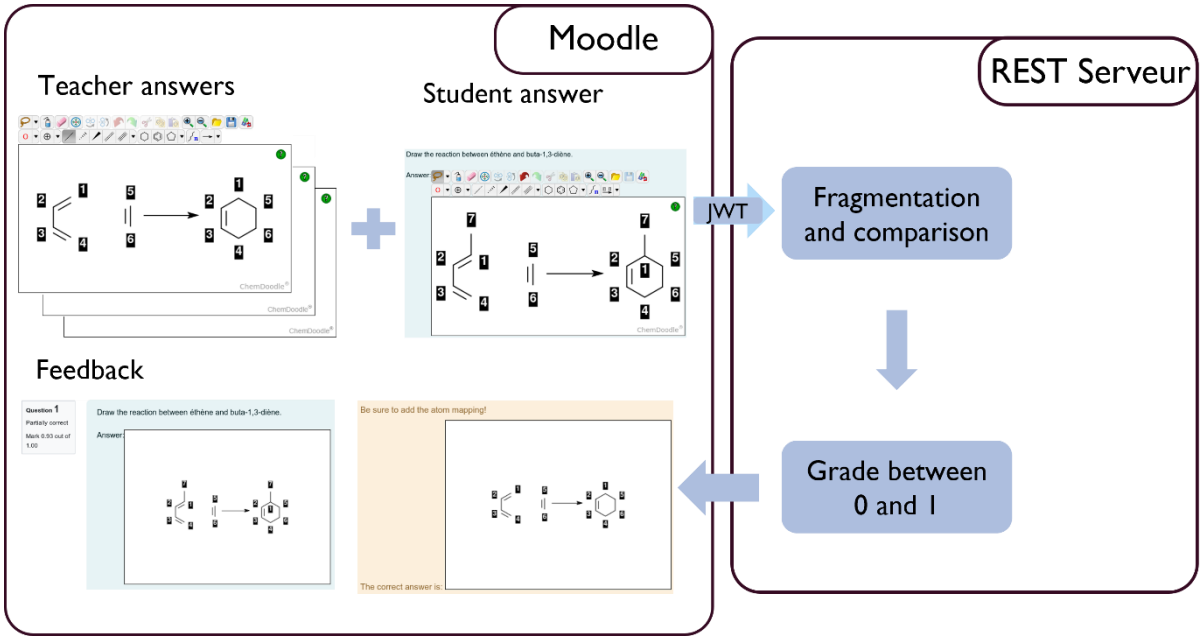

Figure 3-3. MolSimilarity quiz question handling process.

This plugin uses the Chemdoodle engine [6] to draw chemical structures and communicates with a REST server written in Pascal object [106] to calculate the similarity score using ISIDA molecular descriptors [7]. The computational processing of these questions is summarized in Figure 3-3. The module offers two writing interfaces: one for the teacher and one for the student. These interfaces communicate with a server responsible for nuanced evaluation. The server's calculation results are processed by Moodle to produce a grade and feedback for the student. The teacher can insert multiple answers considered correct, and the student's response will be compared to each of them.

Plyer *et al. Journal of Cheminformatics* (2022) 14:72
https://doi.org/10.1186/s13321-022-00645-0

Journal of Cheminformatics

SOFTWARE

Open Access

Implementation of a soft grading system for chemistry in a Moodle plugin

Louis Plyer[1], Gilles Marcou[2*], Céline Perves[3], Rachel Schurhammer[1] and Alexandre Varnek[2]

Abstract

We report a novel approach for grading chemical structure drawings for remote teaching, integrated into the Moodle platform. Typically, existing online platforms use a binary grading system, which often fails to give a nuanced evaluation of the answers given by the students. Therefore, such platforms are unevenly adapted to different disciplines. This is particularly true in the case of chemical structures, where most questions simply cannot be evaluated on a true/false basis. Specifically, a strict comparison of candidate and expected chemical structures is not sufficient when some tolerance is deemed acceptable. To overcome this limitation, we have developed a grading workflow based on the pairwise similarity score of two considered chemical structures. This workflow is implemented as a Moodle plugin, using the Chemdoodle engine for drawing structures and communicating with a REST server to compute the similarity score using molecular descriptors. The plugin (https://github.com/Laboratoire-de-Chemoinformatique/moodle-qtype_molsimilarity) is easily adaptable to any academic user; both embedding and similarity measures can be configured.

Keywords: Educational chemistry, Softgrading, Moodle, Plugin

Introduction

Several solutions have been proposed in the past few years for the remote teaching of chemistry. One of the first tools implying using a chemical structure sketcher for organic chemistry online tutorials with automated correction was described by O'Sullivan and Hargarden [1]. The drawing prepared by the student is exported to a canonical SMILES (Simplified Molecular Input Line Entry Specification) [2] string, followed by the evaluation based on its comparison with an expected answer in SMILES format. Such a solution is realized in the SOCOT platform maintained by the University of Cork and the Dublin Institute of Technology. A similar solution was developed by Flynn et al. [3] for learning nomenclature in chemistry; it is accessible on the *nomenclature101.com* web service hosted by the University of Ottawa. Morsh and Lewis [4] described how the teacher and the students exchange chemistry questions and answers at the University of Illinois–Springfield and at Saint-Louis University, using a touchpad.

OpenOChem [5] is another tool accessible from several Learning Management Systems (LMS). Therefore, it leaves room for Learning Tools Integration (LTI). Unfortunately, the solutions described in [4, 5] cannot be integrated with Moodle [6] or Scenari [7]—Scenari is a popular LMS in France. Moreover, these solutions are based on the ChemAxon web services [8], which are free for academic organizations as long as the company maintains this policy.

Most existing online platforms use a binary grading system, implying a strict comparison of the two canonical SMILES. However, as noticed by Richards-Babb et al. [9], questions whose solutions are based on a limited number of choices are often ineffective for self-assessment. According to their estimations, about a third of students simply try different suggested choices instead of turning

*Correspondence: g.marcou@unistra.fr

[2] Laboratory of Chemoinformatics–UMR7140, University of Strasbourg, Strasbourg, France
Full list of author information is available at the end of the article

© The Author(s) 2022. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

to the course or remediation materials when they do not know the answer. It should also be noted that some questions need smooth grading. A typical example concerns the demand to prepare a chemical structure corresponding to a given SMILES string. The binary assessment results in a grade equal to zero in case of any minor error, whereas the smooth assessment distinguishes the level of students as a function of the number of mistakes in the answer.

Here, we describe a novel software tool able to perform a smooth grading of chemical structures using the Moodle platform. In order to make the tool accessible to any educational institution, we follow the Moodle philosophy [10], so the plugin and all its components are free and open source. Unlike already existing plugins, our tool doesn't transform chemical structures into canonical SMILES because the latter can hardly be applied in certain chemistry case studies [11]. Instead, InChI strings and ISIDA fragment descriptors were used for chemical structure encoding, whereas a pairwise Tanimoto similarity score for teacher/student structures was used for a smooth evaluation of students.

Development

A workflow used by the developed plugin is shown in Fig. 1. In this implementation, we use an in-house correction algorithm hosted on a REST server. It computes the similarity between the student's and teacher's chemical structures. A REST server is, in our opinion, the most relevant technology in this context. It can be managed as suited by the end user–for instance, it can be installed on the same server as Moodle, encapsulated in a virtual machine or a different machine. The server uses little computing power. It doesn't store any data and communicates exclusively with the Moodle server. Data is exchanged using the JSON (JavaScript Object Notation) [12] format, which is a standard in web applications and server transfers. The user interface is built using Chemdoodle Web Component [13], an open source JavaScript library providing a sketcher to draw chemical structures. It can export structures in both SDF (Structure Data File) and Chemdoodle JSON Format. The sketcher can also be used to import a MOL file instead of drawing the molecule. Some services of the sketcher, such as the support of other molecular file formats, have been disabled because they required connections to a foreign server. A pairwise Tanimoto similarity was computed using the ISIDA fragment descriptors [14, 15] generated with the help of the ISIDA Fragmentor2021 tool. ISIDA descriptors represent counts of subgraphs (fragments) of a molecular graph with defined topology and size, contrasting with fingerprint representations, in which a feature appears either present or absent.

A fragmentation scheme, or embedding, is defined by a set of parameters stored in the configuration file; thus, the administrator can tune the parameters if needed.

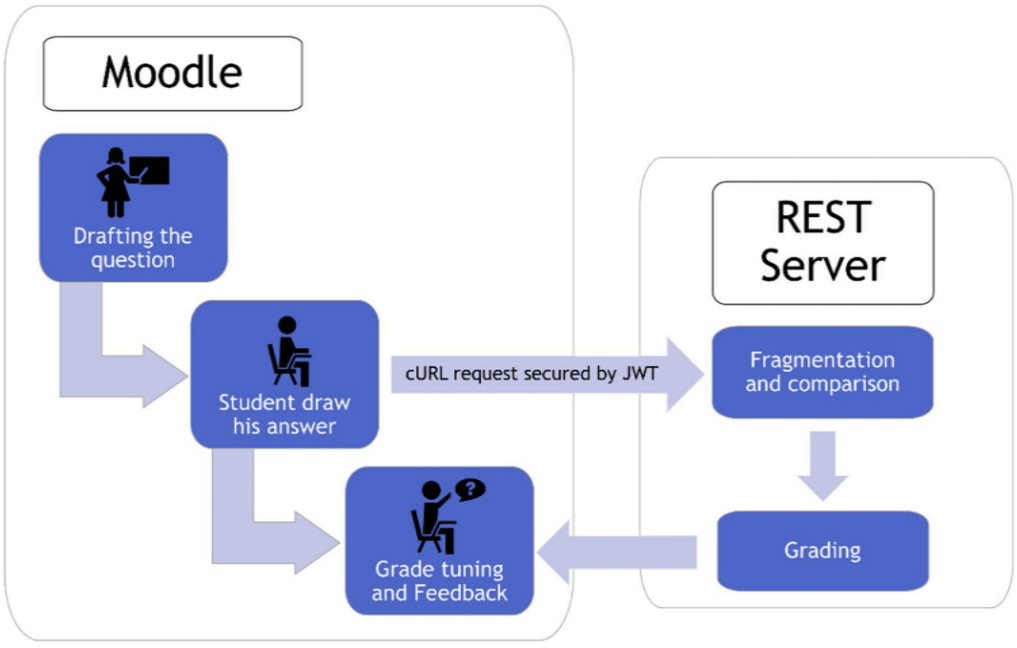

Fig. 1 Workflow of the plugin. The teacher drafts a question, which is proposed to the student via the Moodle interface. The student's and expected answers are sent to a server for comparison and soft grading. The grade is returned to Moodle for evaluation and feedback to the student

The fragments used to compute the molecular similarity are enumerated from the chemical structure of the teacher's answer; they are not pre-defined. The default fragmentation scheme supports organic chemical structures, eventually containing inorganic elements. It takes into account lone pairs, radicals, and formal charges. The grading is sensitive to the presence or absence of explicit hydrogens in the structures–the management of the hydrogens being a part of the evaluation.

The pairwise stereochemistry comparison is based on the InChI [16] strings generated with the help of the InChI v. 1.06 program [17]. The two compared structures without stereo labels must be identical. For them, the algorithm compares the information in related stereolayers; for example, *[/t number_of_atom, stereo_label /m chirality_label*], where *stereo_label* = "+" and "−" and *chirality_label* = "0" and "1".

The communication security between the Moodle server and the REST server is based on JWT (JSON Web Token) [18], an industry standard to secure requests between two entities. They contain three different parts: the header, the payload, and the signature. The header specifies the type of algorithm used to encrypt the signature and the type of token. The payload contains data, including the time at which the token has been issued. Both the header and the payload will be Base64Url encoded [19]. The signature is created using both the header, the payload, and a secret shared between Moodle and the REST API. Therefore, an attacker is not able to change the message without knowing the secret, as a given signature matches only one set of header, payload, and secret.

Implementation

The plugin implementation involves three main steps: (i) formulating a question, (ii) answering a question, and (iii) displaying a teacher's feedback. In this plugin, the teacher inputs from 1 to N deemed correct answers, allowing for several alternative structures (the plugin does not allow the teacher to define "inexact" answers). For instance, for the question "what is the structure of glucose?" both furanose and pyranose forms of glucose can be accepted as answers.

Thus, for each answer, the teacher needs to draw the expected structure using the Chemdoodle Web Component Ketcher, then click on the '*Insert given structure as answer/update the answer with the structure*' button in order to insert this structure (Fig. 2, top). In this case, Chemdoodle JSON is used to encode a given chemical structure in MOL format. Both the Molfile and the Chemdoodle JSON will be saved as an answer in JSON format to the Moodle database.

Apart from chemical structures, the teacher can prepare instructions and feedback for the students. Two kinds of feedback are possible: "general" grade-unrelated feedback and "specific" feedback, displayed if a grade is inferior to 1, aiming to help students improve their answers. Upon taking a test, a student follows the teacher's instructions in order to prepare the required chemical structure (Fig. 2, bottom).

Once submitted, the answer is processed using the same procedure as for the teacher's question (see above). Then, the answers of both the teacher and the student are sent through a cURL [20] request to the correction REST API, written in the Pascal Object language [21]. Connections to the REST API are authenticated using the JWT standard. If the REST server does not respond, the Moodle administrator is notified, and the student's answer is saved and marked as *Requires grading*. If a request is sent to the correction REST API and the authentication is not validated, the Moodle administrator is noticed that someone attempts to access to the correction REST API and receives related IP address.

Once the request is authenticated, the grade g_{rest} based on Tanimoto similarity between the student's and teacher's structures is computed on the REST server. Every chemical structure is encoded using the ISIDA molecular descriptors; by default, fragmentation IAB(2–4)FC_UR is used. It stands for sequences of 2–4 atoms and bonds, taking into account formal charges, lone pairs and radicals. It also includes atom count. If there are several structures prepared by the teacher, the highest Tanimoto score and the corresponding pair of student/teacher structures are kept for the upcoming steps. If the stereochemistry analysis is not requested, the grade g_{rest} is sent back to Moodle. Otherwise, InChI [16] strings are used to compare the teacher's and student's structures containing stereo-centers (the stereo centers can be either R/S or Z/E). The g_{rest} value is computed as the proportion of correctly drawn stereo centers ("*#CorrectStereoCenter*") over the total number of stereo centers in the chemical structure ("*#TotalStereoCenter*"), and sent back to Moodle:

$$g_{rest} = \begin{cases} \dfrac{\#Correct\ Stereo\ Center}{\#Total\ Stereo\ Center}, & if\ \ similarity\ score\ = 1 \\ 0, & if\ similarity\ score\ \neq 1 \end{cases} \quad (1)$$

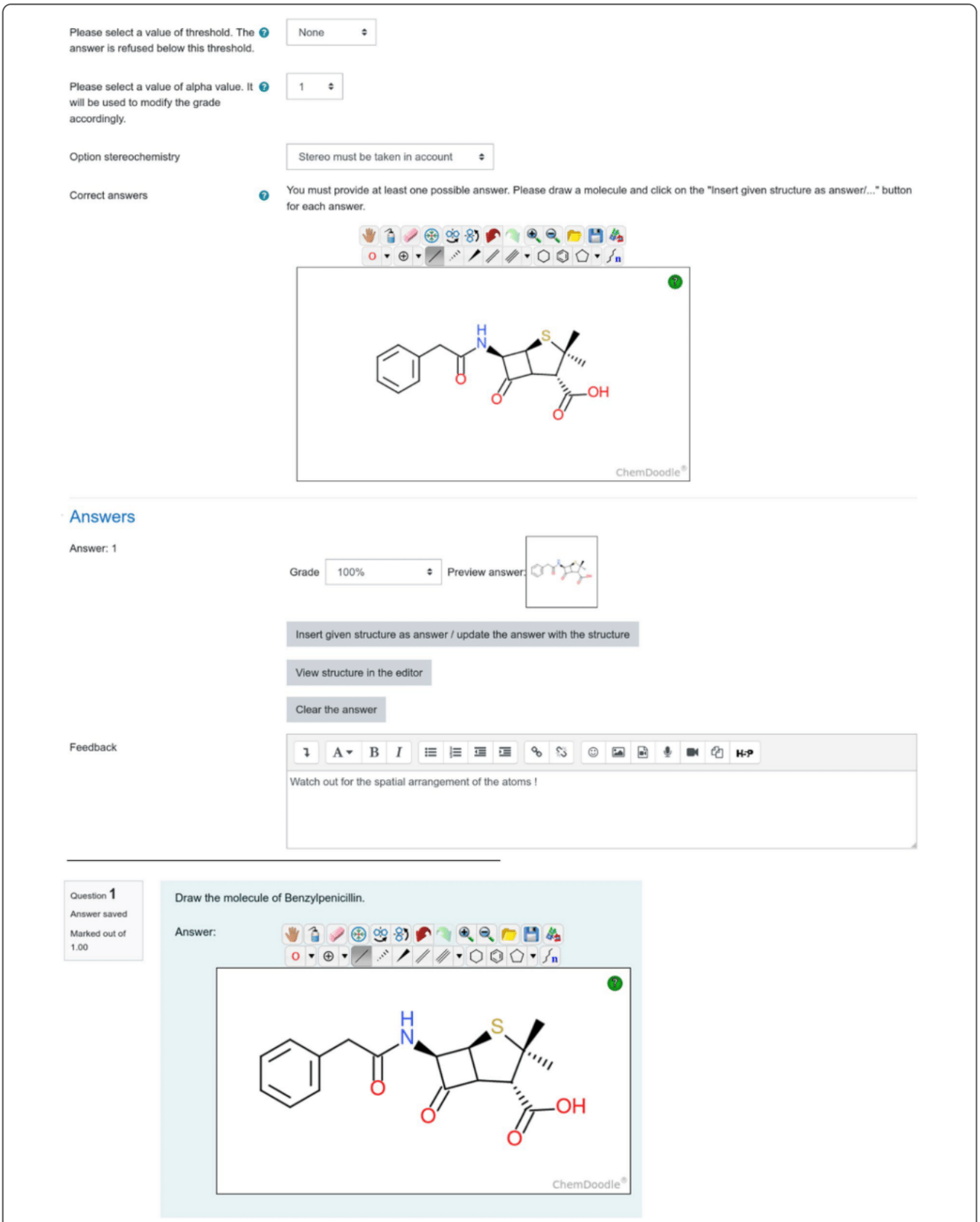

Fig. 2 (Top) Drafting of the question by the teacher. The "Insert given structure as answer / update the answer with the structure" button inserts the current drawing as answer. The "View structure in the editor" button loads the data of the given answer to the sketcher. The "Clear the answer" button removes the answer. (bottom) Interface for the student to answer the question. The teacher's instructions are displayed above the sketcher

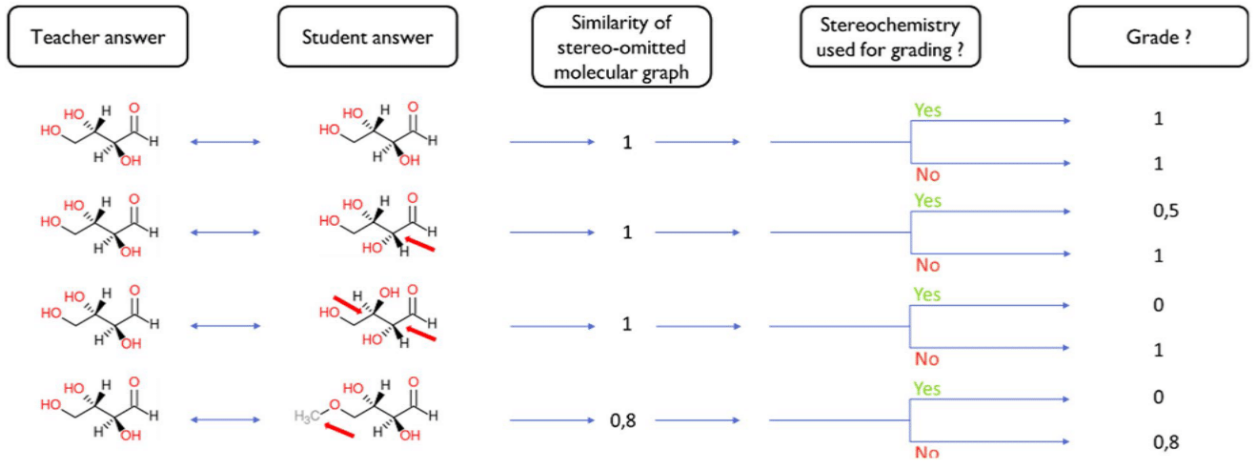

Fig. 3 Illustration of the grading methodology. The red arrows are pointing at the differences between the teacher's and the student's answers. The chemical structures are compared by similarity first, to propose a grade. If the stereochemistry is evaluated, the proportion of correctly drawn centers is used to compute the grade

Notice that the stereocenter comparison becomes impossible if the structures (without stereo labels) are not identical. For this reason, if the similarity score is not equal to 1, a g_{rest} of 0 is returned to Moodle. Typical examples illustrating grading methodology including/excluding stereochemistry analysis are demonstrated in Fig. 3. For instance, if the student confuses an alcohol function with an ether, the Tanimoto similarity score student/teacher structures is 0.8. Therefore, the final grade is either zero, if the stereochemistry is required, or 0.8, otherwise.

Once the grade computed by the REST server is returned to the Moodle server, the final grade g is calculated according to formula (2) where t and α are user-defined parameters. Both parameters (α and t) can be set at the level of the question editor (Fig. 2, top) and can differ from question to question.

$$g = \begin{cases} (g_{rest})^{\alpha}, & if\ (g_{rest})^{\alpha} \geq t \\ 0, & other\ wise \end{cases} \quad (2)$$

The parameter α modulates the teacher's exigency: more lenient (α < 1) or more severe (α > 1). The t parameter is a threshold under which the grade is set to 0, to avoid attributing points to unacceptable answers.

Finally, the general feedback containing the expected answer is shown to the student, accompanied by the specific feedback if g < 1 (Fig. 4).

Question examples

In this section, we describe several typical examples which can be realized with the developed plugin.

Example 1. Drawing a Lewis structure

Both lone pairs, radicals and explicit/implicit hydrogens are considered. Since the correction is not binary, it allows students to be awarded some of the points even if some structural details are missed. For example, when asked for the Lewis structure of Nitrosyl Fluoride, forgetting one of the lone pairs on the Fluorine atom would result in a grade of 0.9/1.

Example 2. Identification of the major product of a reaction

The soft grading system better assesses the student's understanding of the regioselectivity, because the minor products is often similar with the major one. For example, if asked for the major product of 2,3-Dimethyl-2-butanol dehydration by H2SO4, the incorrect result would get a grade of 0.68.

Example 3. Drawing a given configuration (R/S, E/Z) of a molecule

If a compound has multiple stereo-centers but some stereo-centers were not found by the student, the soft grading system can be particularly useful. For instance, the question "what is the structure of glucose?" requires the

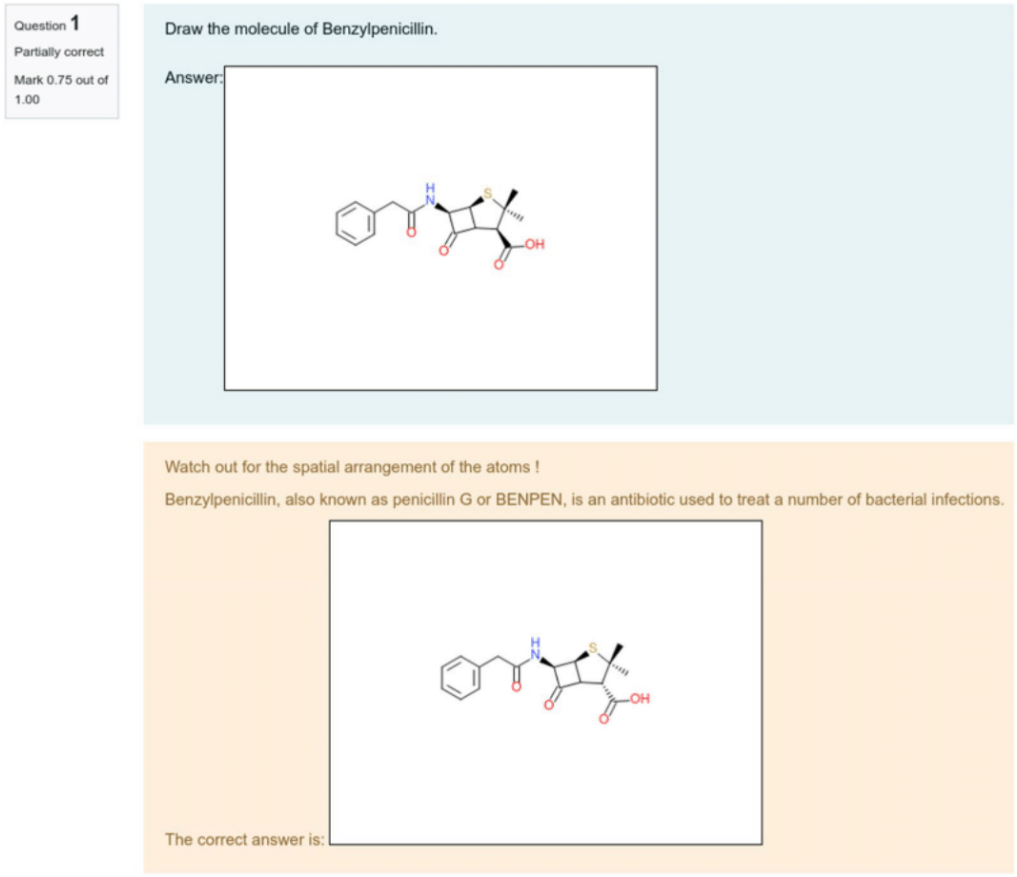

Fig. 4 Interface of the feedback shown to the student once her/his answer has been corrected. Above is the answer of the student, below the expected answer with the teacher's feedback. The mark of this question is displayed on the top left corner

student to consider stereochemistry. In this case, three structures are legitimate answers: open, furanose, and pyranose forms of glucose. Therefore, the teacher should prepare related structures by adding them as expected answers (see Fig. 5). Moreover, for the open form, the structures with both explicit and implicit hydrogen on the aldehyde group on the aldehyde need to be anticipated (Fig. 5, first line). Finally, the teacher needs to consider the stereo-orientation of the methoxy-substituent of the furanose and pyranose forms. In such a way, all 8 alternative structures of glucose (Fig. 5) must be considered as the correct answer. Let's suggest that the student prepare the structure shown in Fig. 6, top. Compared to the closest teacher's structure, s/he has correctly drawn 3 out of 4 stereo-centers. Thus, according to formula (2), his grade is 0.75. Both grade and teacher's feedback are displayed after examination by the algorithm (Fig. 6, bottom).

Comments

It should be noted that the soft grading is "global", therefore it is presently not possible to give more importance to a given substructure.

Conclusion

A new open source Moodle plugin for the assessment of chemical structures has been developed. It significantly extends an arsenal of chemical questions requiring students to draw chemical structures. A soft grading algorithm was implemented in order to reasonably assess the students' skills. The tool is provided with the REST API server that can be used in any institution. It is highly secure with an authentication method needed to access the API, and it allows the teacher to ask chemical questions where the student has to draw her/his answer thanks to the implemented soft grading algorithm. This

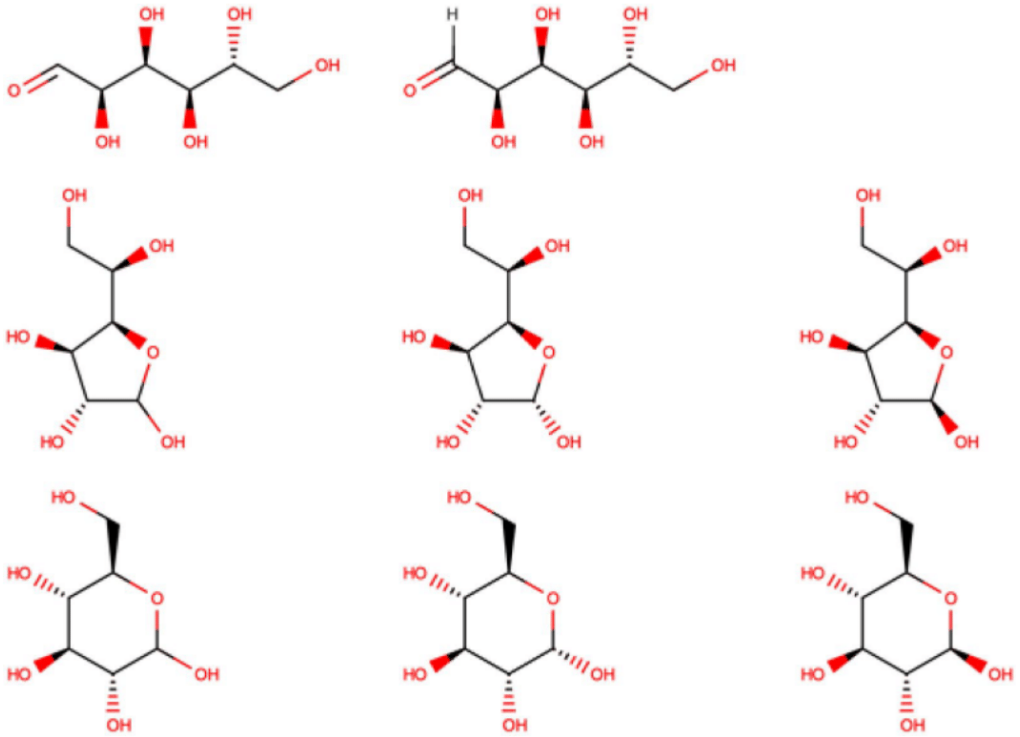

Fig. 5 Eight possible answers expected by the teacher: 2 open structures (with and without explicit hydrogen); 3 furanoses and 3 pyranoses for the α-, β- and undefined isomers

plugin only needs to be installed by the Moodle administrator of the institution, following the same procedure as any other Moodle plugin. It appears as a specific type of question when preparing a test. This work could be enhanced by the addition of several features, such as the possibility to consider wrong answers to give specific feedback to the students (for example, when requested to draw the major product of a reaction, it is desirable to consider the minor product as a wrong answer to provide some specific explanation to the student). The creation of a dedicated tool to automate the editing of questions using a set of chemical structures could be a useful addition–by generating questions in XML format that can be imported into Moodle, for instance. Other options to tune the soft grading are also desirable. For instance, it could be possible to let the teacher build the grade as a weighted sum of the structural similarity and the stereochemistry score. Another improvement could be to let the teacher decide if the module should standardize the protonation of the chemical structures. This work will be improved by the addition of a new layer which enables the asynchronous execution of the evaluation server, on dedicated Docker containers, managed by a RabbitMQ system (Queue message system) [22]. It is fully available from the git of the project: https://github.com/Laboratoire-de-Chemoinformatique/moodle-qtype_molsimilarity.

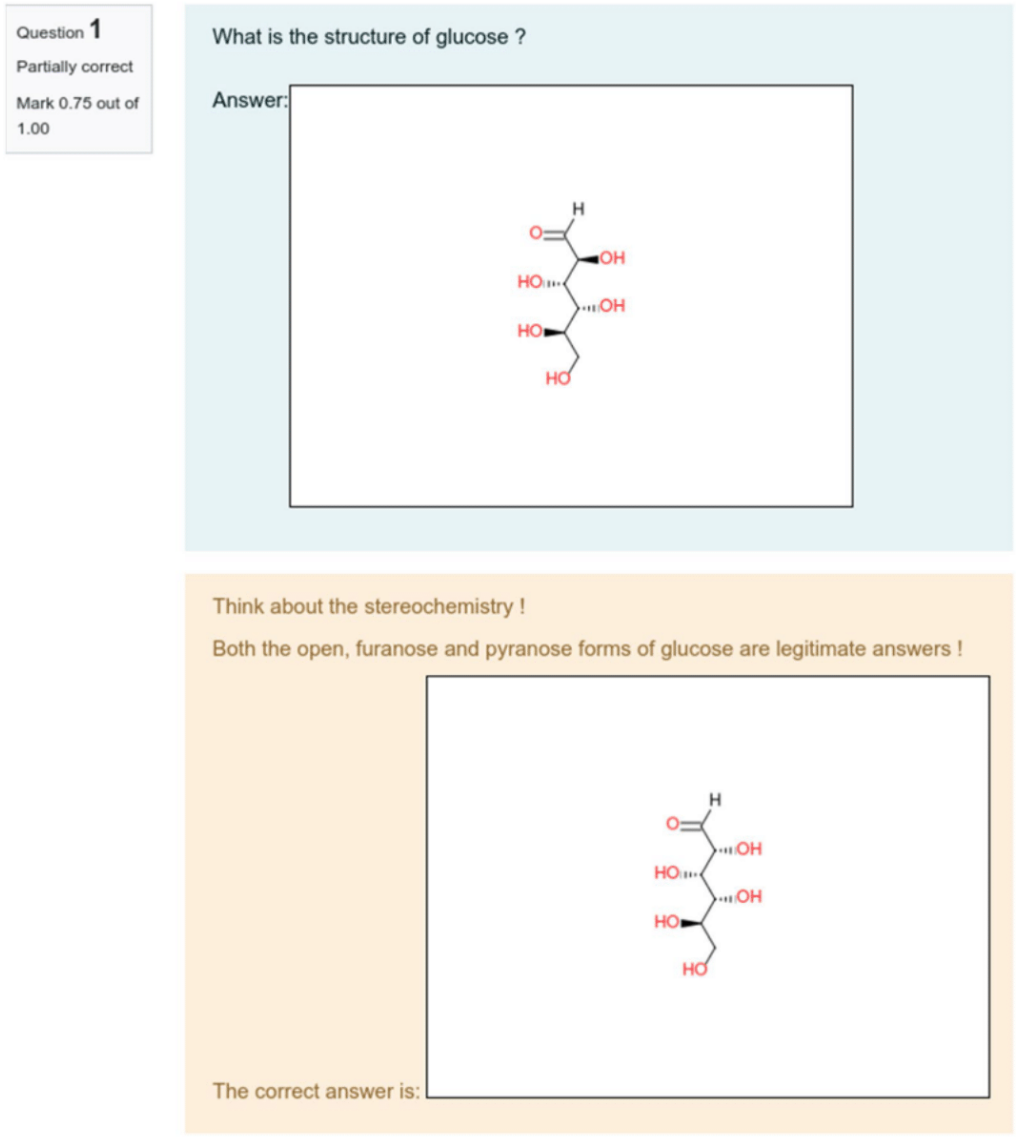

Fig. 6 (top) The open form of glucose with explicit hydrogen on the carbonyl fragment and erroneous stereochemistry chosen by the student; (bottom) specific feedback and correct structure

Acknowledgements
Not applicable

Author contributions
LP, GM and AV are the main contributors to the manuscript and figures. GM and RS contributed to concept and design of the project. LP, GM, CP contributed to code and software development. GM, RS and AV contributed to fund management and fund raising. CP contributed to expertise with LMS, specifically on Moodle. AV, RS and GM contributed to pedagogical expertise. All authors have contributed the final manuscript. All authors read and approved the final manuscript.

Funding
This work has benefited from a state aid managed by the National Research Agency under the program "Investissements d'avenir" with the reference ANR-20-NCUN-0004 DEPHY.

Availability of data and materials
Software is available on the web page of the project: https://github.com/Laboratoire-de-Chemoinformatique/moodle-qtype_molsimilarity. No additional data has been used.
Project name: molsimilarity.
Project home page: Git: https://github.com/Laboratoire-de-Chemoinformatique/moodle-qtype_molsimilarity
Operating system(s): Linux, Mac & Windows.
Programming language: PhP, JavaScript, Pascal Object.
Other requirements: Moodle 3.9.
License: GNU GPL v3 or later, IUPAC/InChI-Trust Licence No.1.0.

Declarations

Ethics approval and consent to participate
Not applicable.

Competing interests
No competing interest to declare.

Author details
[1]Faculté de Chimie, University of Strasbourg, Strasbourg, France. [2]Laboratory of Chemoinformatics–UMR7140, University of Strasbourg, Strasbourg, France. [3]Direction du Numérique (DNUM), University of Strasbourg, Strasbourg, France.

Received: 21 March 2022 Accepted: 17 September 2022
Published online: 26 October 2022

References

1. O'Sullivan TP, Hargaden GC (2014) using structure-based organic chemistry online tutorials with automated correction for student practice and review. J Chem Educ 91:1851–1854. https://doi.org/10.1021/ed500140n
2. Weininger D (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Model 28:31–36. https://doi.org/10.1021/ci00057a005
3. Flynn AB, Caron J, Laroche J et al (2014) Nomenclature101.com: a free, student-driven organic chemistry nomenclature learning tool. J Chem Educ 91:1855–1859. https://doi.org/10.1021/ed500353a
4. Morsch LA, Lewis M (2015) engaging organic chemistry students using chemdraw for iPad. J Chem Educ 92:1402–1405. https://doi.org/10.1021/acs.jchemed.5b00054
5. LeBlond C, Bucholtz E, Muzyka J (2019) OpenOChem: An LMS agnostic chemistry quizzing platform. In: DivCHED CCCE: Committee on Computers in Chemical Education. http://confchem.ccce.divched.org/2019CCCENLP3. Accessed 13 Dec 2021
6. Moodle home page. https://moodle.org. Accessed 9 Dec 2021
7. Scenari home page. https://scenari.org/index.html. Accessed 10 Dec 2021
8. ChemAxon main page. https://chemaxon.com/. Accessed 10 Dec 2021
9. Richards-Babb M, Curtis R, Georgieva Z, Penn JH (2015) Student perceptions of online homework use for formative assessment of learning in organic chemistry. J Chem Educ 92:1813–1819. https://doi.org/10.1021/acs.jchemed.5b00294
10. Moodle (2014) Using learning communities to create an open source course management system. In: Dougiamas. https://dougiamas.com/archives/edmedia2003/. Accessed 14, Dec 2021
11. O'Boyle NM (2012) Towards a universal smiles representation—a standard method to generate canonical smiles based on the InChI. J Cheminform 4:22. https://doi.org/10.1186/1758-2946-4-22
12. Bray T (2014) The JavaScript Object Notation (JSON) data interchange format. Internet Requests for Comments RFC7159. https://doi.org/10.17487/rfc7159
13. Burger MC (2015) ChemDoodle web components: HTML5 toolkit for chemical graphics, interfaces, and informatics. J Cheminform 7:35. https://doi.org/10.1186/s13321-015-0085-3
14. Ruggiu F, Marcou G, Varnek A, Horvath D (2010) ISIDA property-labelled fragment descriptors. Mol Inf 29:855–868. https://doi.org/10.1002/minf.201000099
15. Varnek A, Fourches D, Horvath D et al (2008) ISIDA—platform for virtual screening based on fragment and pharmacophoric descriptors. Curr Comput Aided Drug Des 4:191–198. https://doi.org/10.2174/157340908785747465
16. Heller SR, McNaught A, Pletnev I et al (2015) InChI, the IUPAC international chemical identifier. J Cheminform 7:23. https://doi.org/10.1186/s13321-015-0068-4
17. Download page of the InChI Trust. https://www.inchi-trust.org/downloads/. Accessed 11, Dec 2022
18. Jones M, Bradley J, Sakimura N (2015) JSON Web Token (JWT). Internet Req Comments RFC 7519. https://doi.org/10.17487/RFC7519
19. Josefsson S (2006) The Base16, Base32, and Base64 Data Encodings. Internet Req Comments RFC4648. https://doi.org/10.17487/rfc4648
20. cURL website. In: cURL://. https://curl.se/. Accessed 11 Jan 2022
21. Alekseev E, Chesnokova O, Kucher T (2021) Free Pascal and Lazarus—a textbook on programming. ALT Linux library, Moskow
22. Wood A (2016) Rabbit Mq for Starters. California, USA

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

3.2.2 Summary

In this work, an open-source Moodle plugin has been developed and published to be used by the community. It is innovative in the sense that it proposes a new kind of grading for chemical drawing questions in Moodle, using a soft grading implemented in a REST API server developed for the occasion.

In a later release, the possibility of using a scaffold to be imputed for the student has been added and is presented in the following chapter (*ReacSimilarity*).

MolSimilarity plugin has been downloaded 891 times in the last year and is being used on 45 sites. (released in February 2023). As we can see on Figure 3-4, the number of sites using this plugin is increasing continuously.

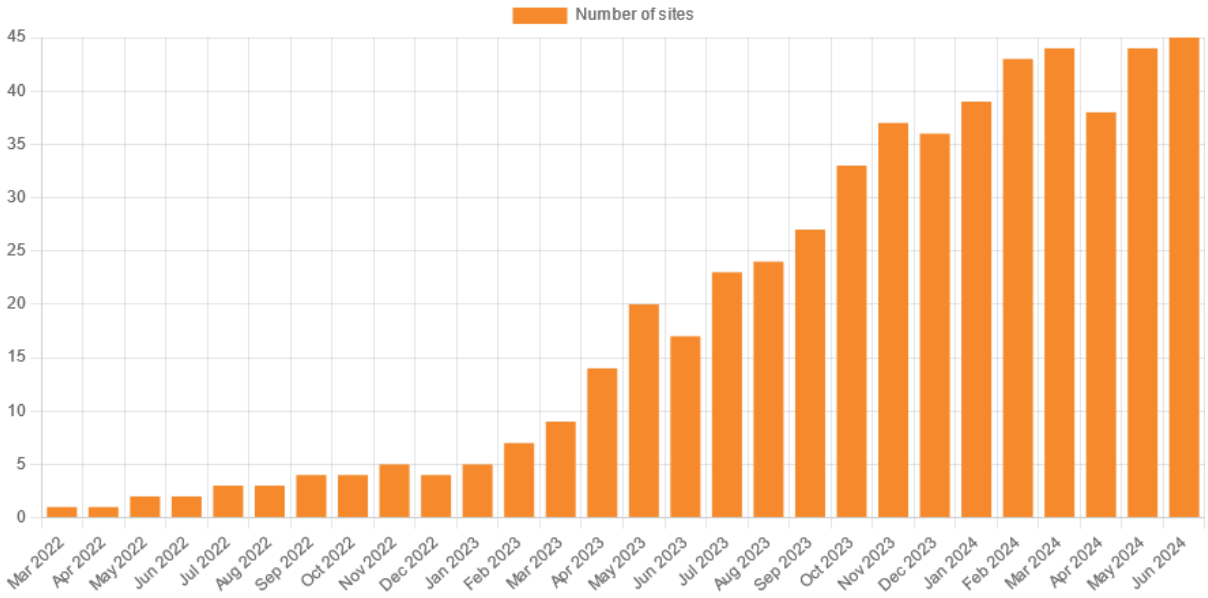

Figure 3-4. Number of sites using the MolSimilarity plugin from its submission on the Moodle platform up to June 2024 (data from Moodle plugin repository).

3.3 ReacSimilarity

3.3.1 Introduction

Following the development of the *MolSimilarity* plugin, which employs molecular similarity to soft grade chemical related questions in the Moodle platform, we introduced the ReacSimilarity plugin to adapt this concept for soft grading of reactions. For easy representation and grading of reactions, we chose to apply the concept of Condensed Graphs of Reactions (CGR) [8–10, 107].

CGRs encode a chemical reaction, encapsulating both reactants and products, within a single molecular graph. This comprehensive representation captures all structural and dynamic changes during a reaction, including bond formations, dissociations, and atom modification. These atoms and bonds that are formed or modified during the reactions are called dynamic atoms and bonds. Thanks to this representation, CGRs may be used in various tasks, from data visualization [108] to deep learning [109], reaction modelling [110–115] or activity cliffs [116].

To allow the use of CGRs in our working environment, we have implemented an algorithm to construct CGRs from Reaction Data File (RDF). This algorithm was designed to construct CGRs from RDF that contain Atom-to-Atom Mapping (AAM). The mapping is of the utmost importance as mentioned in the following article, as it provides a correspondence between atoms in the reactants and products, needed for accurate CGR construction.

The implementation involves several steps. Unconnected graphs are created for both the reactant and products. Then, the AAM is used to compare these two unconnected graphs. Based on this comparison, the algorithm identifies and creates the dynamic atoms and bonds that constitute the reaction center, hence constructing the CGR (Figure 3-5).

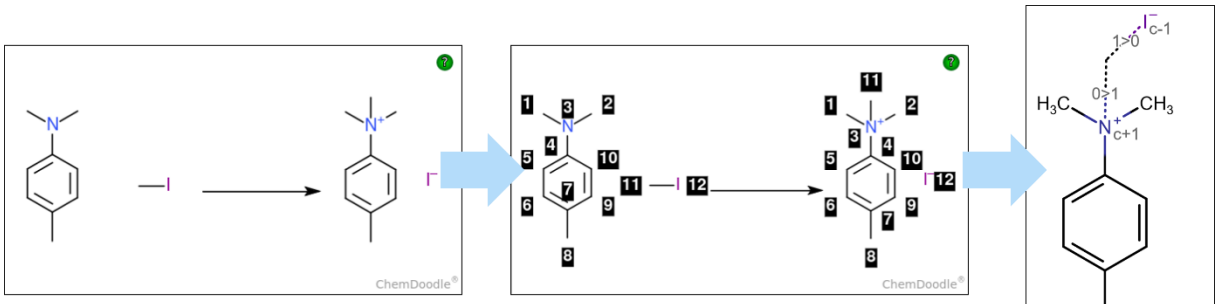

Figure 3-5. Steps involved for CGR creation. Mapping of the reaction and creation of the dynamic atoms and bonds.

Plyer *et al. Journal of Cheminformatics* (2024) 16:90
https://doi.org/10.1186/s13321-024-00889-y

Journal of Cheminformatics

SOFTWARE Open Access

Implementation of a soft grading system for chemistry in a Moodle plugin: reaction handling

Louis Plyer[1], Gilles Marcou[2*], Céline Perves[3], Fanny Bonachera[2] and Alexander Varnek[2]

Abstract

Here, we present a new method for evaluating questions on chemical reactions in the context of remote education. This method can be used when binary grading is not sufficient as some tolerance may be acceptable. In order to determine a grade, the developed workflow uses the pairwise similarity assessment of two considered reactions, each encoded by a single molecular graph with the help of the Condensed Graph of Reaction (CGR) approach. This workflow is part of the ChemMoodle project and is implemented as a Moodle Plugin. It uses the Chemdoodle engine for reaction drawing and visualization and communicates with a REST server calculating the similarity score using ISIDA fragment descriptors. The plugin is open-source, accessible in GitHub (https://github.com/Laboratoire-de-Chemoinformatique/moodle-qtype_reacsimilarity) and on the Moodle plugin store (https://moodle.org/plugins/qtype_reacsimilarity?lang=en). Both similarity measures and fragmentation can be configured.

Scientific contribution

This work introduces an open-source method for evaluating chemical reaction questions within Moodle using the CGR approach. Our contribution provides a nuanced grading mechanism that accommodates acceptable tolerances in reaction assessments, enhancing the accuracy and flexibility of the grading process.

Keywords Educational chemistry, Softgrading, Moodle, Plugin, Chemical reactions

Introduction

Numerous studies have demonstrated that the completion of homework activities exerts a substantial influence on the academic achievement of individual students [1, 2]. Besides exerting an impact on the grades of the students, homework activities, particularly those conducted online, are viewed by students as valuable learning tools [3]. Thus, Vijay S. Vyas and Scott A. Reid concluded that "a combination of active learning pedagogy, core concepts curricula, and incorporation of low-stakes assessments is a strategy capable of moving the needle to improve DFW rates in second-term general chemistry" [4] where DFW rates are the % of D and F grades and withdrawals in a given class. This emphasizes the utility for low-stakes assessments and the need of technical tools to implement them. The importance of distance learning in Chemistry has increased considerably in response to the Covid 19-health crisis [5]. Yet, distance learning has been a long-time concern for the modernization of pedagogy. It is exemplified in the "Charte de l'Enseignement à Distance" [6] of the University of Strasbourg. This implies the development of questionnaires for which correction is automated.

*Correspondence:
Gilles Marcou
g.marcou@unistra.fr
[1] Faculté de Chimie, University of Strasbourg, Strasbourg, France
[2] Laboratory of Chemoinformatics-UMR7140, University of Strasbourg, Strasbourg, France
[3] Direction du Numérique (DNUM), University of Strasbourg, Strasbourg, France

© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

These questionnaires are used during the registration of new students in order to identify knowledge gaps and skill deficiencies [7]. They also serve as an instrument of formative assessment in the preparation and consolidation of knowledge, enabling learners to identify their strengths and weaknesses, benefiting from instructors' feedback, and adjusting their learning strategies [8].

Hence, the existence of tools facilitating online homework assignments in the domain of chemistry is natural and several authors have proposed operational solutions. Thus, Korsakova et al. [9] described a "Chemist Bot" helping Russian students to prepare for chemistry exams. It is designed as a conversational bot proposing remediation articles and text-based quiz questions with immediate feedback. In general, students who used the Chemist Bot performed better on the United State Exam (USE) of the Russian Federation.

O'Sullivan and Hargarden reported one of the earliest instances of utilizing a chemical structure sketcher in online tutorials with automated correction [10]. The students' drawings are converted into a canonical SMILES (Simplified Molecular Input Line Entry Specification) [11] string, which are then assessed by comparing them to an anticipated response also in SMILES format. This approach is implemented in the SOCOT platform, which is overseen by the University of Cork and the Dublin Institute of Technology. Similarly, Otálvaro has proposed a method [12] that involves a sequential procedure for students to respond to chemical questions utilizing the JSME editor [13] on their mobile devices. The students are required to generate the SMILES notation of their answer, which they have to copy and paste into ChemDrawJS [14] to generate an InChI [15] code. Ultimately, the InChI code must be pasted into Socrative [16], the learning management system (LMS) / web-based classroom response system (WBCRS) used in that contribution.

Earlier, we proposed the implementation of a soft grading system for chemistry in the Moodle platform [17] able to automatically evaluate the candidates whose answers contain a chemical structure drawing. This method proposes a mark for the answer proportional to the graph similarity between the answer and the solution. Moodle is the most widely used free LMS [18].

In contrast to "multiple-choice" (closed-ended) questions, this method allows for "constructed-response" (open-ended) questions. This change of methodology makes the self-assessment more effective, as it insists on reasoning instead of trails and errors efforts. [19]. Indeed, as estimated by Richard-Babb et al. [20] about closed-ended questions, approximately one-third of students resort to attempting different suggested options instead of referring to course materials and reasoning when faced with difficult questions.

Liu et al. [21] discussed the advantages and disadvantages of both open-ended and closed-ended question types. Closed-ended questions have low variability in their grading as opposed to the variability originating from the teachers correcting open-ended questions. Open-ended questions typically require more time to score. Yet open-ended question enables for a direct assessment of the students' knowledge without any help/clues: it enables the teachers to identify possible students' misconceptions and inconsistencies. The analysis of the answers to open-ended questions is valuable for the creation of more effective teaching and remediation actions.

The ChemMoodle plugins combining *Reacsimilarity* (described here), *Molsimilarity* [17], and *MolStructure* [22] tools allow the teachers to benefit from strong points of both open-ended and closed-ended strategies mentioned by Liu et al. [21]: easy, fast, objective, and reproducible scoring while retaining the ability to highlight and understand the possible misconceptions from the students.

A few tools allowing teachers to ask questions about chemical reactions were reported in the literature [10, 23]. As an answer, the students are supposed either to draw a single molecule [10] or to write a reaction equation [23]. In the last accessible versions of the plugin [24] (before its replacement by the OpenOChem platform [23]), the correction of the reaction equation was done by comparison of expected and students' SMILES of the reaction. The exact fit between expected and students answers was used for the automated assessment process in either case.

In this work, we propose a new approach based on the Condensed Graph of Reaction (CGR) [25–28] method. This technique allows to compare the students' answers concerning chemical reaction equation to the expected answer from the teachers in a way analogous to the Molsimilarity plugin. To our knowledge, it is the only completely open-source chemical question type plugin to work with reactions, while guaranteeing privacy—the plugin does not trigger any undesired communication over the internet. In the following, we discuss implementation details and how user feedback has been taken into account. Feedbacks were gathered during 5 hackathons organized by the University of Strasbourg between September 2023 and May 2024 to present the tools to end users, mostly teachers. The plugin has been released on the Moodle store in November 2023. As of May 2024, four sites are using it.

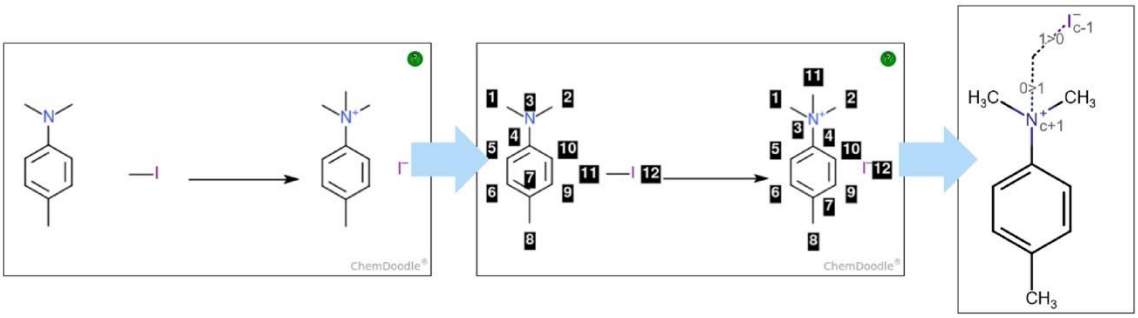

Fig. 1 Atom mapping process: Unmapped reaction on the left, atom mapping of the reaction in the middle and associated CGR on the right ("0 > 1": create a single bond; "1 > 0": break a single bond)

CGR/atom mapping

A CGR [25–28] allows to encode a chemical reaction by a single molecular graph. A CGR is described by both conventional chemical bonds (single, double, triple, aromatic) and so-called dynamic bonds and dynamic atoms characterizing chemical transformations. Thus, dynamic bonds describe breaking, formation, and changing bond order whereas dynamic atoms describe the change of their oxidation number as a result of the chemical reaction. CGR results from superimposing reactants' and products' atoms bearing the same identifiers. Earlier [25–28], the CGR approach was used in the building of machine-learning models for the rate constant of SN2 reactions [28], reaction similarity searches [29], protective groups' reactivity assessment [30], and reaction condition prediction [31]. In conjunction with an autoencoder, it has also been used for an AI-based generation of chemical reactions [32].

The superposition of atoms of reactants and products to construct the CGR requires atom-to-atom mapping (AAM) (Fig. 1 shows the process for the reaction with CAS Number 31-031-CAS-23647760). It consists in assigning to each reactant's atom a unique number and the same unique number to the corresponding atom in the products. AAM is a valuable tool for the classification of chemical reactions and elucidation of the reactions' mechanisms. AAM can be associated to a pattern matching exercise and is usually performed algorithmically [33]. However, these patterns are ambiguous. A proper formulation of the electronic reorganization over the reactants disambiguates the matched patterns, leading to a specific AAM. To enable students to gain and test their understanding of chemical reactions, they are asked to manually number each atom in the reactants and products.

To shed light on reaction mechanisms by describing electrons movement, the teachers can use curved arrow notations ("mechanistic" arrows). Houchlei et al. [34] have raised the point that students understanding a mechanism, with the help of mechanistic arrows, are more efficient when faced to a new mechanism compared to students learning by heart.

Quiz questions involving mechanistic arrows are covered by the OpenOChem [23] tools, which is a different and complementary approach compared to the *Reac-similarity* module. In *ReacSimilarity*, the correct mapping results from a correct understanding of the electron movements. We believe that by performing AAM manually, the students can effectively track the movement of electrons, capture reorganization processes, and challenge their understanding of chemical reactivity [35]. In such a way, this is a valuable didactic exercise, providing students with a practical experience of elucidation of reaction mechanisms at the atomic level. To define one AAM, the user selects the "Reaction mapping" arrow in the arrow toolset, then click on a reactant atom and drags to the corresponding product atom. The exercise can be tedious for a student if they have to start from scratch. However, by using "reaction templates", the teacher can guide the students to specific parts of a reaction to be evaluated. For instance, the teacher may map some of the atom pairs in advance. This reduces the amount of input needed from the students, allowing them to answer faster. For a given reaction, alternative correct AAM are possible. All of them correspond to the same CGR.

ISIDA fragment descriptors

Once the CGRs are built from the reactions drawn by both students and teachers, they are encoded by the ISIDA fragment descriptors [36] generated for the 2D molecular graph to compute the grades, similarly to the *Molsimilarity* module. There are 3 types of ISIDA descriptors: (i) sequences of atoms and bonds or atoms only, (ii) atom-centered fragments, and (iii)

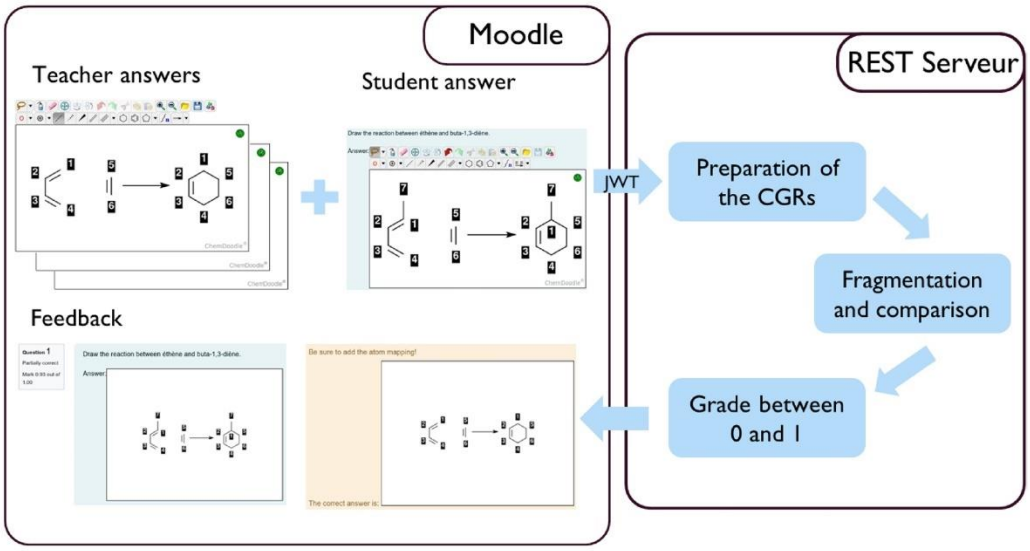

Fig. 2 Workflow of the plugin. The teacher prepares a question and the student an answer (left panel "Moodle"). Both are sent to a server (right panel "REST Server") where they are interpreted and compared on a 0 (completely different) to 1 (complete match) scale. This estimate of the grade is sent back to the student and teacher for feedback and assessment purposes (left panel "Moodle")

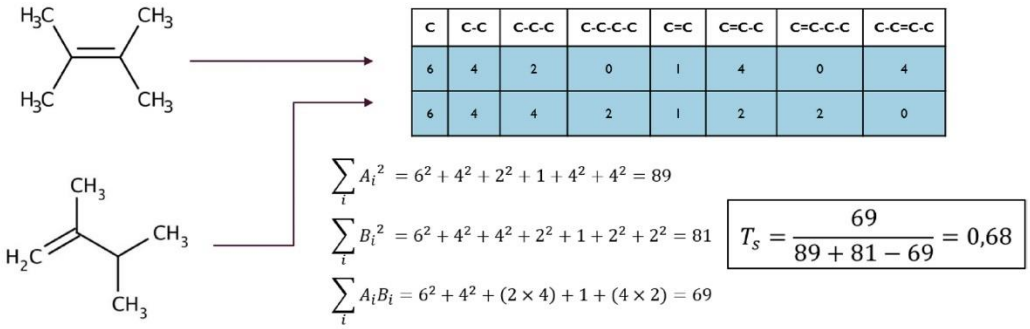

Fig. 3 Tanimoto similarity calculation, with ISIDA descriptors. The formula used is Formula 1

pharmacophoric triplets. In such a way, a molecular graph can be encoded by a vector consisting of the fragments' counts.

Implementation

The workflow of the *Reacsimilarity* plugin is given in Fig. 2. Although it is similar to the *Molsimilarity* module [17], some differences should be pointed out. The user interface on the Moodle side uses Chemdoodle Web Component [37], and exports both the RXN (reaction data file) and Chemdoodle JSON format inside a JSON (JavaScript Object Notation) [38] to the Moodle database. The Chemdoodle code was amended to append the atom-to-atom mapping to the RXN, and to be able to show AAM in the feedback process. The communication between Moodle and the REST server is secured by the JSON Web Token (JWT) [39] standard. The exchange of data between Moodle and the Rest server is based on the JSON format.

A REST server is used for the assessment of similarity between the students' and teachers' structures. First, the reactant and product molecules are aromatized with the help of the Indigo library [40]. The REST server transforms the reaction to CGR using ISIDA software package [41]. ISIDA fragments descriptors and pairwise Tanimoto similarity are then computed by the correction server. The ISIDA descriptors are computed using the ISIDA Fragmentor2022 tool [36] (Fig. 3). The whole process

is implemented as a REST API and hosted on a REST server. The latter can be launched from the same server as Moodle and can be encapsulated in a virtual machine or in a docker container managed by a RabbitMQ system [42]. This server communicates exclusively with the Moodle server.

The fragmentation scheme of the CGRs is defined by default but can be modified by the administrator of the correction server by amending a set of parameters stored in the configuration file in XML format. By modifying these parameters, the user can change the size of the descriptors, the sensitivity to bond types, atom types, or both. The documentation of the plugin is describing which configuration file to edit and how. The molecular descriptors used for the correction of a given question are computed on the fly using the chemical structures input of both the students and the teachers. The ISIDA molecular descriptors (ISIDA fragments) support the encoding of radicals, lone pairs, and formal charges. Presence or absence of explicit hydrogens have an impact on the encoding of the answers and the solution to a quiz question. Therefore, teachers must provide instructions on how they expect the drawings to be done and provide alternative answers with and without hydrogens, when deemed needed. These alternative answers can be added a posteriori if necessary. Although the chemistry sketcher represents implicit hydrogens, those cannot be mapped, preventing any possible confusion.

As mentioned before, the students perform the AAM needed for CGR construction. The teachers can provide students with a starting point using the "reaction template", a partial drawing of the reaction that may include a partial atom-to-atom mapping. This partial AAM is useful to guide the evaluation to focus on a specific part of the reaction only. This mapping procedure is not time-consuming. Any mistakes in AAM results in the creation of wrong dynamic bonds. Notice that an error on the reaction center has more impact than an error on other parts of the molecules. Indeed, an error on the reaction center would modify the type of dynamic atoms and bonds, and therefore modify molecular fragments. As consequence, the CGR fragments descriptor vectors of the correct and erroneous answers are orthogonal. On the other hand, a modification out of the reaction center may still lead to CGR fragments that are in common between the correction and the answer. This is exemplified in Fig. 4 where an error on the reaction center of the Diels Alder reaction between ethylene and buta-1,3-diene (wrong AAM, Fig. 4a) yields a score of 0.57, while an incorrect nucleophile (penta-1,3-diene instead of buta-1,3-diene, Fig. 4b) yields a score of 0.93.

Stereochemistry analysis is performed with the help of the InChI [15] strings generated with the InChI v.1.06 program. We chose to use InChI as the stereochemistry information is located in defined layers that are technically simple to compare. The InChI strings are computed for reactants and products of both students' and teachers' answers. Then, the correction server compares the information in the InChI stereo-layers [/t "stereo labels" on atoms, /m and /s complementary "chirality label"], where "stereo label" = "+" and "−" and "chirality label" = "0" and "1".

Results

For a given question, a teacher prepares from 1 to N reaction equations that are considered correct. It allows to accommodate for chemical ambiguities (mesomeres, tautomers, etc.), see Fig. 5 for an example of the reaction between Pentane-2,4-dione and Ammonia. It also adds some degrees of freedom to the teachers concerning the questions that may be asked.

A teacher can incorporate an initial "reaction template" representing a part of reaction equation (e.g., reactants or products only) or full reaction equation with or without AAM. The student is supposed to finalize reaction equation and to add the mapping. This allows teacher to focus the test on more specific skills and knowledge, such as reaction mechanisms questions. This feature has also been integrated into the most recent version of the Molsimilarity plugin.

For each of these answers, the teachers will draw the reaction using the Chemdoodle Web sketcher, map the reaction (Fig. 1, middle) and click on the "Insert given reaction as answer / update the answer with the reaction". To map the reaction, the user must use the "Reaction Mapping" tool, then click and drag from one atom of the reactant to one atom of the products. Both Chemdoodle JSON format and RXN files will be stored in JSON format [38] in the Moodle database. In addition to chemical reactions, the teachers have the option to provide instructions and feedback for the students.

There are two types of feedback available: "general" feedback that is not related to grades, and "specific" feedback that is displayed when a grade is below 1. The specific feedback aims to assist students in improving their responses.

When taking a test, students follow the instructions and draw the required chemical reaction. Upon test completion, the students' answers are processed, following the same procedure as the teachers' answers. Then, the students' and teachers' answers are sent to the REST API through a cURL [43] request and authenticated via JWT standard. The similarity is then computed by the REST API. The REST API is written in Pascal Object language [44], enabling support of a large number of computer systems at binary code and trivial recompilation from the source—for those interested.

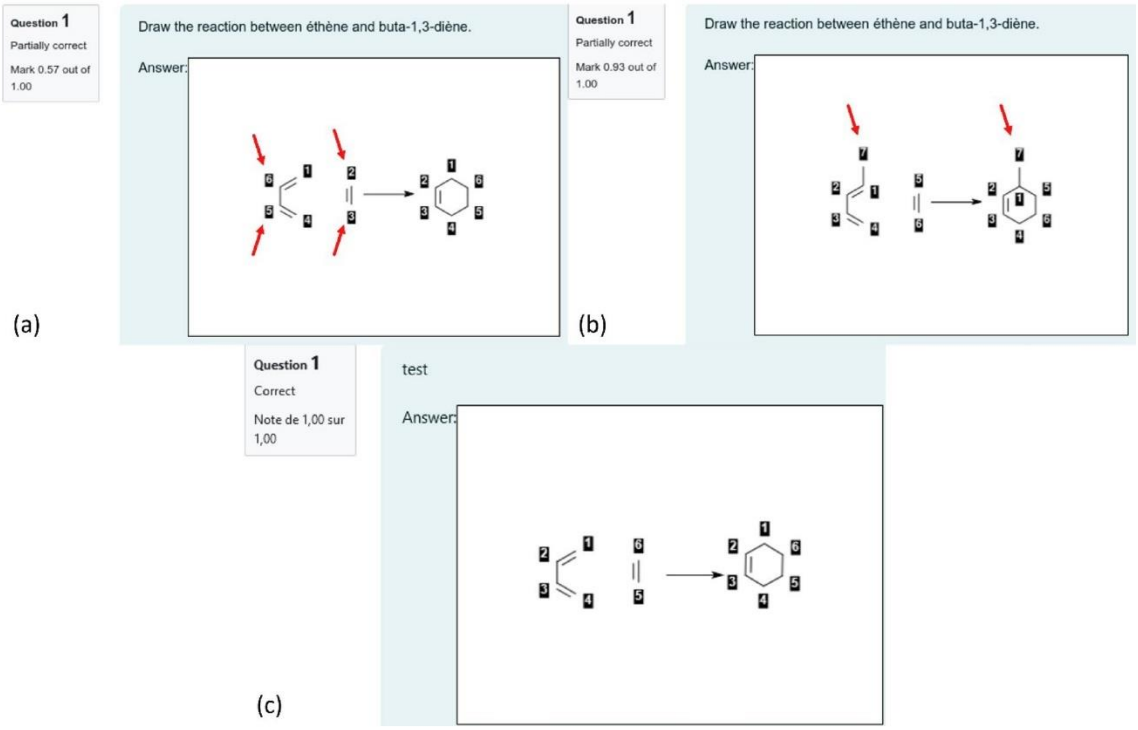

Fig. 4 *Top*: Two erroneous students answers concerning the Diels Alder reaction between ethylene and buta-1,3-diène related to (**a**) wrong mapping of atoms in reaction center which provides a score of 0.57 and (**b**) addition of the Me group to butadiene providing a score of 0.93. The red arrows emphasize on the errors made by the students. **c** Expected answer

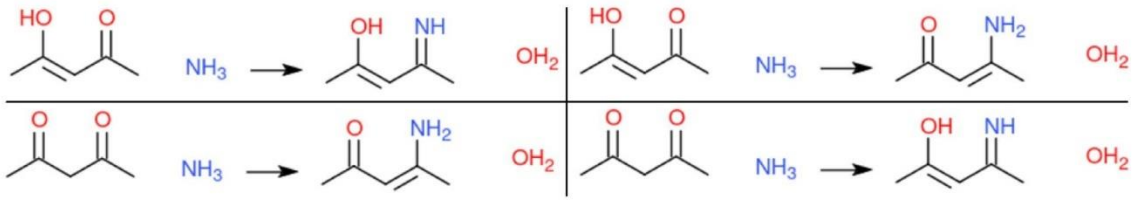

Fig. 5 Alternative correct answers for the Pentane-2,4-dione with ammonia reaction, taking in account the tautomeric structures

If the server fails to respond to the request, a warning message is sent to the Moodle administrator. In this case, the students' answers will be saved and flagged as "needing correction". If an unauthenticated request is made, the Moodle administrator will be warned too, and the IP address of the attacker will be linked in the corresponding warning.

As mentioned above, to compute the grade, reactions are aromatized using the Indigo library [40]. The CGR of the reactants and products from the students and teachers are generated using AAM. Then, both the students' and teachers' CGRs are encoded using the ISIDA molecular descriptors. Fragmentation IAB (2–4) FC_UR is used, standing for sequences of 2–4 atoms and bonds, considering formal charges, lone pairs, and radicals. The configuration of the fragmentation is stored in an XML file that can be edited by the Moodle administrator.

As a function of the type of teacher's question, several scenarios of grading are considered (Fig. 6).

(a) Reaction template is not given (e.g., "Prepare equation for Diels–Alder reaction between butadiene

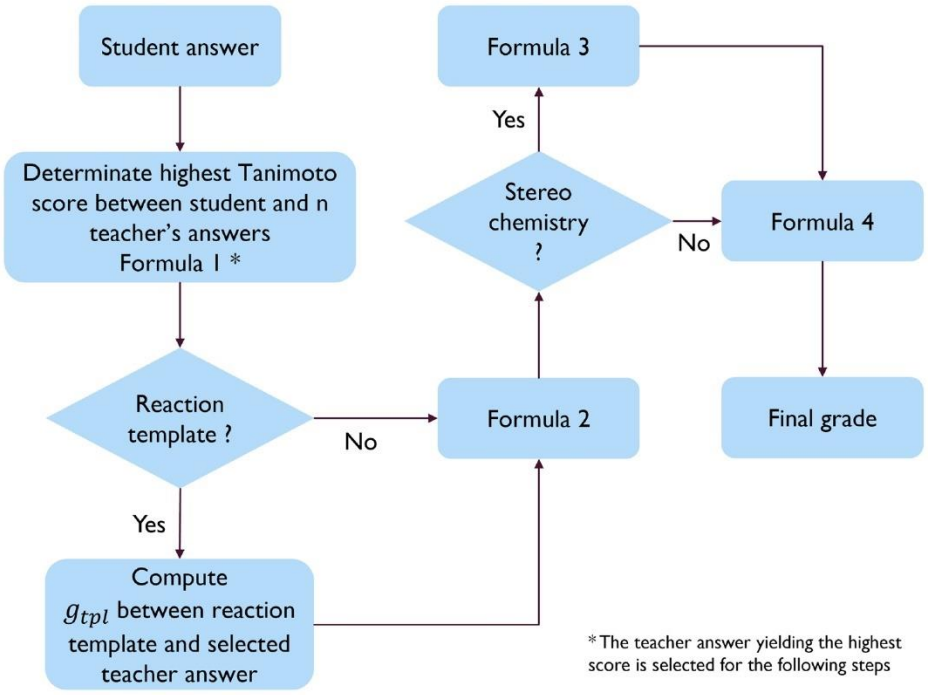

Fig. 6 Decision network summarizing how the grade is computed according to different user cases scenario

and ethylene"). In this case, the grade g_{sim} is the maximal Tanimoto similarity between the students' answers and the 1 to N teachers' answers (Formula 1).

$$T_s = g_{sim} = \frac{A \bigcap B}{A \bigcup B} = \frac{\sum_i A_i B_i}{\sum_i {A_i}^2 + \sum_i {B_i}^2 - \sum_i A_i B_i} \tag{1}$$

(b) If several alternative answers are possible (see Fig. 5), a maximal similarity score is taken as g_{sim}. If a teacher employs a "reaction template", the algorithm will assess the similarity between the template and the full reaction equation of teacher's answers that yielded the maximal Tanimoto similarity (g_{tpl}) even if a student gives no answer at all. In order to take the "reaction template" into account, a final score is modified according to formula (2) where g_{sim} is a similarity score related to the student's answer.

$$g_{rest} = \begin{cases} \frac{(g_{sim} - g_{tpl})}{(1 - g_{tpl})}, & if\,''reaction\,template'' \\ g_{sim}, & if\,no''reaction\,template'' \end{cases} \tag{2}$$

(c) If stereochemistry is not requested by the teachers, the computed grade g_{rest} is sent back to Moodle. Otherwise, the InChI [15] strings are used to compare stereo centers (R/S or Z/E) of each reactant and product from the students and teachers reactions. The grade g_{rest} will be computed as the proportion of correctly drawn stereo centers ("#CorrectStereoCenter") over the total number of stereo centers in the reaction ("#TotalStereoCenter"), and sent back to Moodle (Formula 3), in the same way as for the *Molsimilarity* module.

$$g_{rest} = \begin{cases} \frac{\#\,Correct\,Stereo\,Center}{\#\,Total\,Stereo\,Center}, & if\,similarity\,score, g_{sim} = 1 \\ 0, & if\,similarity\,score, g_{sim} \neq 1 \end{cases} \tag{3}$$

It should be noted that application of InChI strings to compare stereo implies that compared chemical structures may only differ on the stereochemistry. Indeed, a change in any part of a chemical function may alter its priority level to assess the stereochemistry label. Because of this, if the similarity score g_{sim} is not equal to 1 and the stereochemistry is required in the grading process, a g_{rest} of 0 is returned to Moodle.

Once the stereochemistry assessment is performed, the score g_{rest} is sent back to Moodle, where the final grade g is calculated according to Formula 4. This formula introduces the user-defined parameters t and α. These parameters allow to modulate the softness of the grading: for small values of α, more errors will be tolerated while large values will deteriorate the grade for any deviation from the expected answer. If the computed grade is below the cutoff parameter t, no points are given to the question. By

default, t is equal to 0 and has a range going from 0 to 1, and α is equal to 1 and has a range going from 0.1 to 10. Both parameters are set by the teachers while preparing the question and each question can have different values of these parameters. Finally, the general feedback is displayed to the students, containing the expected answer, as well as the specific feedback if $g < 1$.

$$g = \begin{cases} (g_{rest})^{\alpha}, if\ (g_{rest})^{\alpha} \geq t \\ 0, \ otherwise \end{cases} \quad (4)$$

Equivalent AAM (for instance, by permutation of the atom numbering) are accepted as correct. For this reason, the expected answer displaying the solution proposed by the teacher and the student answer may appear visually different while being equivalent and maximally graded. This display can be confusing to some students.

Conclusion

The development of automated tools for online chemistry homework assignments has become increasingly important, especially in the context of distance learning and the need for continuous assessment.

This article proposes a new approach that expands the scope of automated correction from individual molecules to entire chemical reactions using the concept of Condensed Graph of Reaction (CGR). By utilizing CGRs, atom mapping, and substructural fragments, the proposed tool enables the correction and assessment of students answers at a holistic level, capturing the complexity of chemical reactions.

The incorporation of AAM in the students' answers not only eases the grading process but also serves as an educational exercise, allowing students to gain a deeper understanding of chemical reactions at the atomic level.

The implementation of the tool involves a user-friendly interface and utilizes Chemdoodle Web Component to draw reactions. The grading process involves creating the CGRs, encoding them using ISIDA fragment descriptors, and computing pairwise Tanimoto similarity between students' and teachers' answers.

This work could be improved by adding several features. It should be possible in a next update, to make the use of AAM optional. The question could then be to draw a chemical reaction whatever the AAM is. The stereochemistry correction information could be used to build specific feedback for the students. This feedback would categorize answers as identical to the correct answer, a constitutional isomer, a diastereomer, an enantiomer, or introducing ambiguity, offering much richer insights to students. The University of Strasbourg is organizing hackathon and is sharing experience through meetings such as the Moodle Moot. Indeed, the use of a new tool needs some training. Specific documentation designed for teachers will also be shared with the community.

The proposed tool offers several advantages, including the ability to handle multiple correct answers, the option to provide initial reaction templates, and the generation of both general and specific feedback for students. By combining the strengths of open-ended and closed-ended question types, the tool provides objective, efficient, and knowledge-based assessment.

Author contributions
LP, GM, and AV are the main contributors to the manuscript and figures. GM contributed to concept and design of the project. LP, GM, CP contributed to code and software development. GM and AV contributed to fund management and fund raising. CP contributed to expertise with LMS, specifically on Moodle. FB contributed to expertise with the ISIDA software. AV and GM contributed to pedagogical expertise. All authors have contributed the final manuscript. All authors read and approved the final manuscript.

Funding
This work has been funded thanks to the "Programme Investissements d'Avenir", IdEx Formation 2022—Structuration—ref project 33.

Data availability
No datasets were generated or analysed during the current study.

Materials availability
All material is provided free of charge under GNU GPL v3 licence. All material is available either on the Moodle store (https://moodle.org/plugins/qtype_reacsimilarity) or the Git of the project (https://github.com/Laboratoire-de-Chemoinformatique/moodle-qtype_reacsimilarity).

Declarations

Competing interests
The authors declare no competing interests.

Received: 17 April 2024 Accepted: 21 July 2024
Published online: 01 August 2024

References

1. Eichler JF, Peeples J (2013) Online homework put to the test: a report on the impact of two online learning systems on student performance in general chemistry. J Chem Educ 90:1137–1143. https://doi.org/10.1021/ed3006264
2. Freasier B, Collins G, Newitt P (2003) A web-based interactive homework quiz and tutorial package to motivate undergraduate chemistry students and improve learning. J Chem Educ 80:1344. https://doi.org/10.1021/ed080p1344
3. Richards-Babb M, Jackson JK (2011) Gendered responses to online homework use in general chemistry. Chem Educ Res Pract 12:409–419. https://doi.org/10.1039/C0RP90014A
4. Vyas VS, Reid SA (2023) What moves the needle on DFW rates and student success in general chemistry? A quarter-century perspective. J Chem Educ 100:1547–1556. https://doi.org/10.1021/acs.jchemed.2c01121
5. Dietrich N, Kentheswaran K, Ahmadi A et al (2020) Attempts, successes, and failures of distance learning in the time of COVID-19. J Chem Educ 97:2448–2457. https://doi.org/10.1021/acs.jchemed.0c00717
6. Charte de l'ÉàD—EAD—Enseignement à distance—Université de Strasbourg. https://ead.unistra.fr/communaute-ead/charte-de-lead. Accessed 18 Aug 2023

7. Müller MT, Togni A, Thilgen C (2021) Evaluation of the chemistry knowledge of students entering the ETH Zurich with a Moodle Quiz. Chimia 75:89. https://doi.org/10.2533/chimia.2021.89
8. Successes and Challenges: Online Teaching and Learning of Chemistry in Higher Education in China in the Time of COVID-19. https://doi.org/10.1021/acs.jchemed.0c00671. Accessed 3 Aug 2023
9. Korsakova E, Sokolovskaya O, Minakova D et al (2022) Chemist bot as a helpful personal online training tool for the final chemistry examination. J Chem Educ 99:1110–1117. https://doi.org/10.1021/acs.jchemed.1c00789
10. O'Sullivan TP, Hargaden GC (2014) Using structure-based organic chemistry online tutorials with automated correction for student practice and review. J Chem Educ 91:1851–1854. https://doi.org/10.1021/ed500140n
11. Weininger D (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 28:31–36. https://doi.org/10.1021/ci00057a005
12. Otálvaro F (2022) Merging drawing-based questions with automatic assessment in organic chemistry using smartphones. J Chem Educ 99:3044–3048. https://doi.org/10.1021/acs.jchemed.2c00278
13. Bienfait B, Ertl P (2013) JSME: a free molecule editor in JavaScript. J Cheminform 5:24. https://doi.org/10.1186/1758-2946-5-24
14. ChemDraw JS Sample Page. https://chemdrawdirect.perkinelmer.cloud/js/sample/index.html. Accessed 25 Apr 2023
15. Heller SR, McNaught A, Pletnev I et al (2015) InChI, the IUPAC International Chemical Identifier. J Cheminform 7:23. https://doi.org/10.1186/s13321-015-0068-4
16. Socrative home page. https://www.socrative.com/. Accessed 25 Apr 2023
17. Plyer L, Marcou G, Perves C et al (2022) Implementation of a soft grading system for chemistry in a Moodle plugin. J Cheminform 14:72. https://doi.org/10.1186/s13321-022-00645-0
18. Sanchez L, Penarreta J, Soria Poma X (2024) Learning management systems for higher education: a brief comparison. Discov Educ 3:58. https://doi.org/10.1007/s44217-024-00143-5
19. Campbell ML (2015) Multiple-choice exams and guessing: results from a one-year study of general chemistry tests designed to discourage guessing. J Chem Educ 92:1194–1200. https://doi.org/10.1021/ed500465q
20. Richards-Babb M, Curtis R, Georgieva Z, Penn JH (2015) Student perceptions of online homework use for formative assessment of learning in organic chemistry. J Chem Educ 92:1813–1819. https://doi.org/10.1021/acs.jchemed.5b00294
21. Liu OL, Lee H-S, Linn MC (2011) An investigation of explanation multiple-choice items in science assessment. Educ Assess 16:164–184. https://doi.org/10.1080/10627197.2011.611702
22. (2023) Moodle plugins directory: Chemical substances (Atto). https://moodle.org/plugins/atto_molstructure. Accessed 7 Feb 2024
23. LeBlond C, Bucholtz E, Muzyka J (2019) OpenOChem: An LMS Agnostic Chemistry Quizzing Platform. In: DivCHED CCCE: Committee on Computers in Chemical Education. http://confchem.ccce.divched.org/2019CCCENLP3. Accessed 13 Dec 2021
24. (2016) Moodle plugins directory: Name to Structure or Reaction (MarvinJS). https://moodle.org/plugins/qtype_easyonamejs. Accessed 18 Aug 2023
25. Fujita S (1986) Description of organic reactions based on imaginary transition structures. 1. Introduction of new concepts. J Chem Inf Comput Sci 26:205–212. https://doi.org/10.1021/ci00052a009
26. Fujita S (1987) Description of organic reactions based on imaginary transition structures. 6. Classification and enumeration of two-string reactions with one common node. J Chem Inf Comput Sci 27:99–104. https://doi.org/10.1021/ci00055a002
27. Varnek A, Fourches D, Hoonakker F, Solovev VP (2005) Substructural fragments: an universal language to encode reactions, molecular and supramolecular structures. J Comput Aided Mol Des 19:693–703. https://doi.org/10.1007/s10822-005-9008-0
28. Hoonakker F, Lachiche N, Varnek A, Wagner A (2011) A representation to apply usual data mining techniques to chemical reactions—illustration on the rate constant of S N 2 reactions in water. Int J Artif Intell Tools 20:253–270. https://doi.org/10.1142/S0218213011000140
29. de Luca A, Horvath D, Marcou G et al (2012) Mining chemical reactions using neighborhood behavior and condensed graphs of reactions approaches. J Chem Inf Model 52:2325–2338. https://doi.org/10.1021/ci300149n
30. Lin AI, Madzhidov TI, Klimchuk O et al (2016) Automatized assessment of protective group reactivity: a step toward big reaction data analysis. J Chem Inf Model 56:2140–2148. https://doi.org/10.1021/acs.jcim.6b00319
31. Afonina VA, Mazitov DA, Nurmukhametova A et al (2021) Prediction of optimal conditions of hydrogenation reaction using the likelihood ranking approach. IJMS 23:248. https://doi.org/10.3390/ijms23010248
32. Bort W, Baskin II, Gimadiev T, et al Discovery of Novel Chemical Reactions by Deep Generative Recurrent Neural Network. 20
33. Lin A, Dyubankova N, Madzhidov TI et al (2022) Atom-to-atom mapping: a benchmarking study of popular mapping algorithms and consensus strategies. Mol Inf 41:2100138. https://doi.org/10.1002/minf.202100138
34. Houchlei SK, Bloch RR, Cooper MM (2021) Mechanisms, models, and explanations: analyzing the mechanistic paths students take to reach a product for familiar and unfamiliar organic reactions. J Chem Educ 98:2751–2764. https://doi.org/10.1021/acs.jchemed.1c00099
35. Clayden J, Greeves N, Warren S (2012) Organic chemistry, 2nd edn. Oxford University Press Inc, New York
36. Ruggiu F, Marcou G, Varnek A, Horvath D (2010) ISIDA property-labelled fragment descriptors. Mol Inf 29:855–868. https://doi.org/10.1002/minf.201000099
37. Burger MC (2015) ChemDoodle Web Components: HTML5 toolkit for chemical graphics, interfaces, and informatics. J Cheminform 7:35. https://doi.org/10.1186/s13321-015-0085-3
38. Bray T (2014) The JavaScript Object Notation (JSON) Data Interchange Format. Internet Requests for Comments RFC7159. https://doi.org/10.17487/rfc7159
39. Jones M, Bradley J, Sakimura N (2015) JSON Web Token (JWT). Internet Requests for Comments RFC 7519. https://doi.org/10.17487/RFC7519
40. Pavlov D, Rybalkin M, Karulin B et al (2011) Indigo: universal cheminformatics API. J Cheminform. https://doi.org/10.1186/1758-2946-3-S1-P4
41. Varnek A, Fourches D, Horvath D et al (2008) ISIDA—platform for virtual screening based on fragment and pharmacophoric descriptors. CAD 4:191–198. https://doi.org/10.2174/157340908785747465
42. Wood A (2016) Rabbit Mq for Starters, CreateSpace Independent Publishing Platform
43. cURL website. In: cURL://. https://curl.se/. Accessed 11 Jan 2022
44. Alekseev E, Chesnokova O, Kucher T (2010) Free Pascal and Lazarus - A textbook on programming. ALT Linux library, Moskow

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

3.3.2 Summary

In this work, a new open-source question type Moodle plugin has been developed and published. It is innovative in the sense that it proposes a new kind of grading for chemical reaction drawing questions in Moodle, using soft grading implemented in a REST API server developed for the occasion.

We also developed new units in the ISIDA software, making possible the construction of CGRs from the input of the students, allowing for fast, easy, and comprehensive correction of their answers. As for the *MolStructure* plugin and for the same reason, *ReacSimilarity* plugin has been updated to maintain up-to-date frameworks.

At this date, the *ReacSimilarity* plugin has been downloaded 36 times in the last year and is being used on 11 sites. (released in March 2024). As we can see on Figure 3-4, the number of sites using this plugin is increasing continuously.

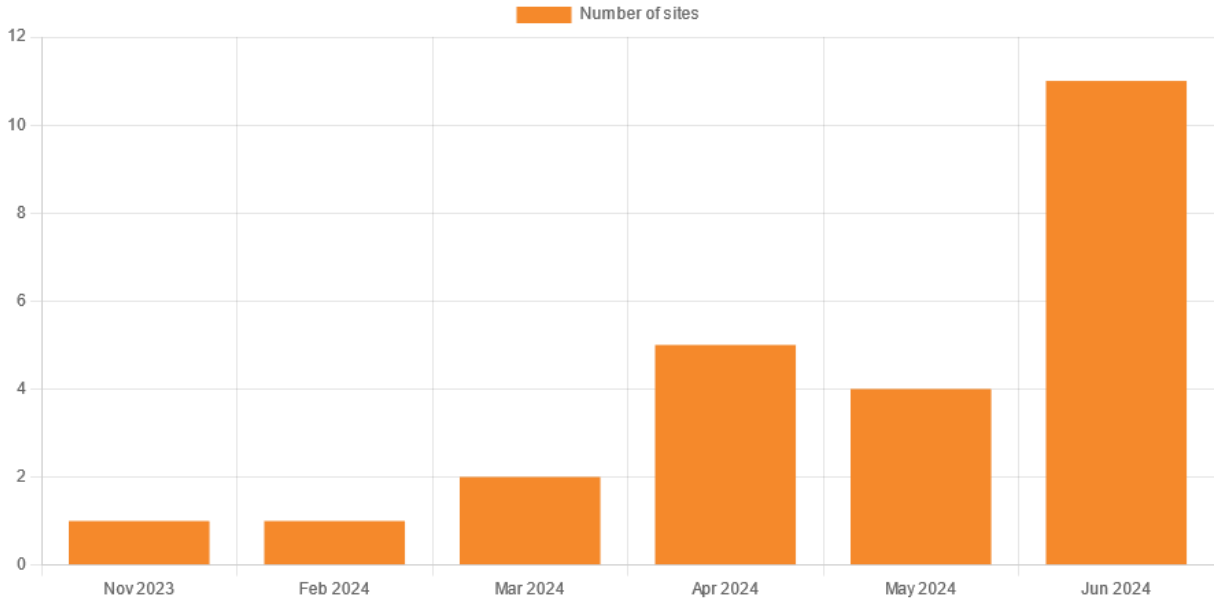

Figure 3-6. Number of sites using the MolSimilarity plugin from its submission on the Moodle platform up to June 2024 (data from Moodle plugin repository).

The proof version of the manuscript presented here has been accepted in Journal of Cheminformatics after reviewing the 21/07/2024.

3.4 ChemEngineering

3.4.1 Introduction

When learning chemistry, students are required to master elements of chemical engineering and are therefore required to produce process flow diagrams (PFD, Figure 3-7). A PFD in chemical engineering represents the equipment used in the chemical industry and their relationship to describe a process (e.g., chemical transformation, distillation, etc). The purpose of the diagram is to provide a comprehensive view of the necessary equipment, and their settings for the process and describe the material flows.

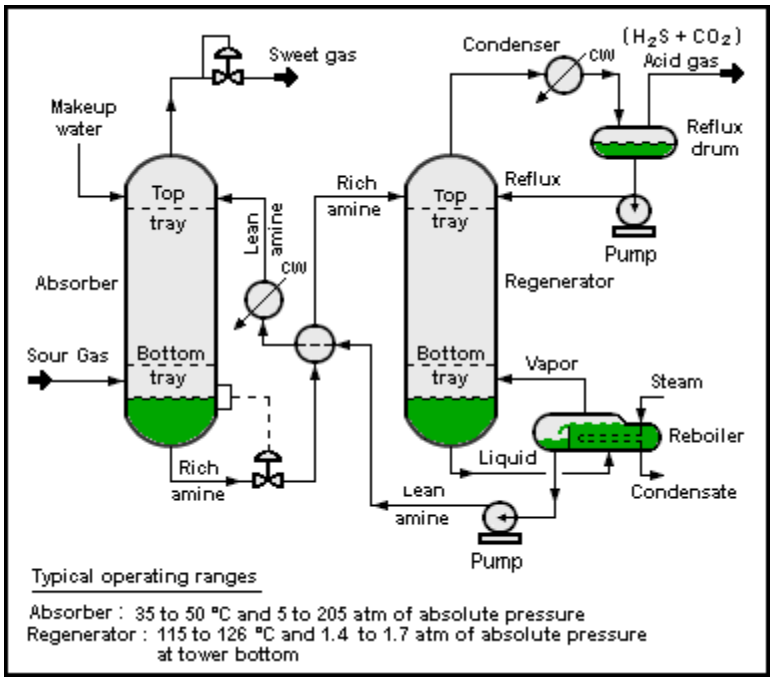

Figure 3-7. An example flow diagram of a typical amine treating process used in industrial plants.

PFD play an important role in chemical engineering, as they visually depict the sequence of operations in a chemical process, illustrating how raw materials are converted into valuable products. These diagrams are essential for designing, optimizing, troubleshooting, and improving processes and existing systems. By learning to create and interpret these diagrams, students will have a clearer view of how chemical processes are structured and managed.

Presently, the tools for drawing PFD are commercial, not freely available, and lack integration with LMS. To address this issue, we are developing a new solution: an

open-source plugin that allows the insertion of process diagrams into educational content, integrated into the Moodle platform.

The lack of accessible and integrated tools for creating process diagrams poses a significant barrier to education in chemical engineering. Most commercially available software packages are expensive and require specialized training, making them impractical for widespread use in educational settings. Furthermore, the absence of these tools in LMSs, such as Moodle, means that teachers and students cannot easily create and share their diagrams within their coursework. By developing an open-source plugin specifically for Moodle, our goal is to democratize access to these essential tools by developing an open-source plugin specifically for Moodle, allowing students and educators to incorporate process diagrams into their online learning environment easily. This integration will facilitate a more interactive and practical learning experience, enabling students to apply theoretical knowledge in a practical context.

The development of a robust and user-friendly plugin called for collaboration between experts with diverse expertise. Chemical engineering experts, Sébastien THOMAS and Célien JACQUEMARD, ensure the tool meets the specific needs to the field, while web developers bring the technical skills required to build a functional and efficient plugin. Distance education specialists, Sophie KENNEL, contributes insights into how the tool can best support online learning environments, while Moodle expert Céline PERVES ensure that the tool answers the technical needs of the platform. This interdisciplinary approach ensures that the plugin is technically sound and educationally effective. Together, we worked on identifying user needs, designing the user interface, and developing key functionalities.

Following this work, we created a white paper outlining the goals, features, and implementation plan. The Ikigai Games for Citizens association (a non-profit organization aimed at developing "serious games," i.e.) implemented the tool based on this document. This partnership with Ikigai Games for Citizens aims to benefit from their expertise in creating engaging educational tools. Our team supervises the work (functionalities, graphical interface) and participates in integrating the editor into Moodle.

This plugin represents the first open-source solution of its kind, developed entirely in TypeScript [117], relying on JavaScript compilation to make it compatible with a wide range of web browsers. With this plugin, chemical engineering instructors

can easily create and integrate process diagrams into their online courses, thus providing an interactive and immersive learning experience. Developing the plugin for JavaScript ensures it can run efficiently on any modern web browser, offering accessibility across different devices and operating systems. This universality is crucial for educational tools, as it ensures that all students, regardless of their device, can use the plugin.

The open-source aspect of this plugin is also a key element of its design. By publishing the source code on GitHub, a collaborative development platform, we encourage the community to adopt it and contribute to its continuous improvement. The management of this work involves two libraries hosted on GitHub, one for the process diagram editor code and the other for the Moodle plugin. Users are encouraged to contribute with suggestions, bug fixes, and additional features, ensuring that the plugin remains up-to-date and meets the evolving needs of chemical engineering instructors and students. The dual-library structure on GitHub separates the core functionalities of the diagram editor from the Moodle-specific integration, making it easier for contributors to focus on specific areas of improvement.

In conclusion, the development of an open-source PFD plugin for Moodle signals the beginning of a significant advancement in chemical engineering education. By providing a free, accessible, and collaborative tool, we aim to enhance the educational experience, promote community engagement, and ensure the tool continuously evolves and improves to meet the diverse needs of its users. This initiative not only democratizes access to essential educational resources but also fosters a spirit of collaboration and innovation within the academic and professional community.

In the following section, we present the white paper. This document outlines the plan and detailed considerations that guided the development of the open-source PFD. For clarity, we provide the translation of the white paper.

3.4.2 White paper

3.4.2.1 Introduction

A process flow diagram in chemical engineering represents the equipment of the chemical industry and their relationships to describe a process (e.g., chemical transformation, distillation, etc.). The diagram aims to provide an overview of the necessary equipment for the process and to describe the material flows.

3.4.2.1.1 Equipment

Equipment plays an essential role in the treatment and transformation of chemical raw materials. They are designed to perform specific operations such as chemical reaction, separation, purification, distillation, condensation, crystallization, filtration, etc.

Equipment is represented by distinct graphic elements. They are depicted simply with black lines without background color (Figure 3-8). Each equipment has entry and/or exit points called **I/O points**.

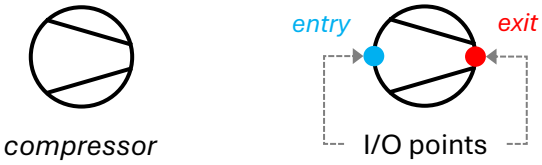

Figure 3-8. Graphic representation of a compressor with an entry point and an exit point.

An I/O point can be either an entry point, an exit point, or both at the same time, called a bidirectional point. The nature of the I/O point is important. It can be either **material** or **information** depending on what is being transported. For a given piece of equipment, groups of points can be defined (Figure 3-9).

In the example of Figure 3-9, 4 groups are defined:

- *Reagent entries* – 3 material entry points
- *Product exit* – 1 material exit point
- *Input/output heat carrier* – 4 bidirectional material points
- *Information outputs* – 2 bidirectional information points

The groups allow representing the logic of the I/O points of equipment. A group can only consist of the same type of I/O points.

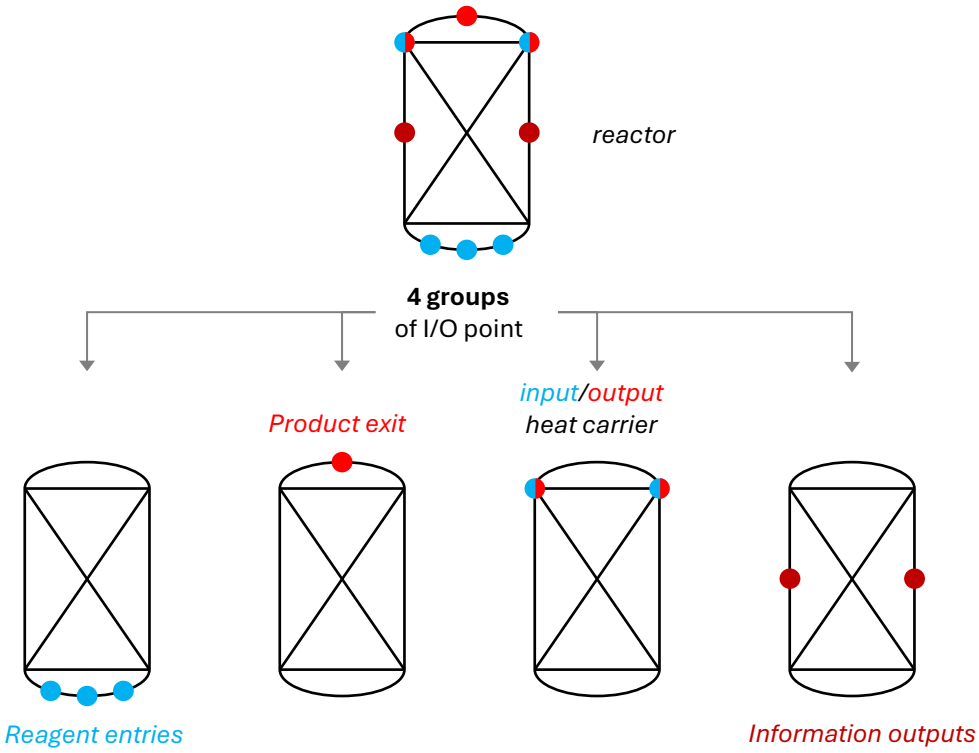

Figure 3-9. A reactor with its I/O points.

When a process flow diagram is completed, two "special" pieces of equipment must be present: the process entry and the process exit. These two pieces symbolize the beginning and the end of a process. A diagram can have multiple entries and multiple exits. The entry has a single exit point, and the exit has a single-entry point (Figure 3-10).

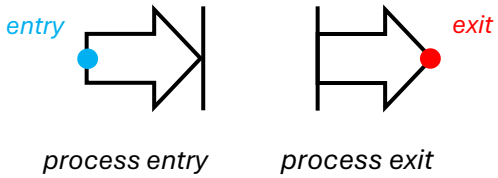

Figure 3-10. Graphic representation of the process entry and exit with their I/O points.

The complete list of equipment is provided in the dedicated section. Details about the equipment are given in this section.

3.4.2.1.2 Connector

Equipment is connected by connectors represented by perpendicular jagged lines with an arrow indicating the flow direction (exit to entry). There are two types of connectors (Figure 3-11):

- **pipe** – used to connect material I/O points.
- **cable** – used to connect information I/O points.

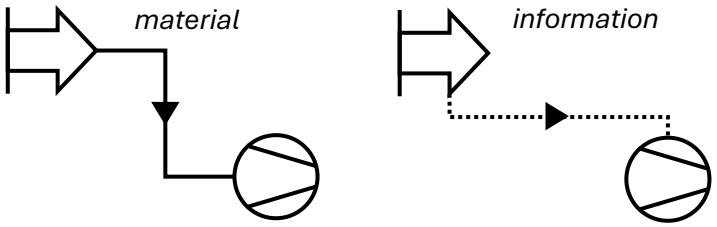

Figure 3-11. *Graphic representation of a pipe (left) and a cable (right) connecting two pieces of equipment.*

The rules for linking equipment together will be explained later.

3.4.2.1.3 Annotation

Equipment and connectors can be annotated to describe components, parameters, etc. An annotation must be attached to equipment or a connector (Figure 3-12). An annotation cannot be attached to multiple equipment/connectors.

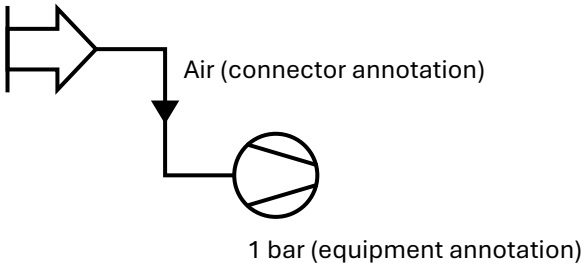

Figure 3-12. *Graphic representation of two annotations, one on an equipment, and one on a connector.*

Annotations can be added in two ways. Either the user creates their annotations, or the user places predefined annotations made by another user. In the latter case, it is called a predefined annotation. These predefined annotations can be attached to a specific piece of equipment or not.

3.4.2.2 User Interface

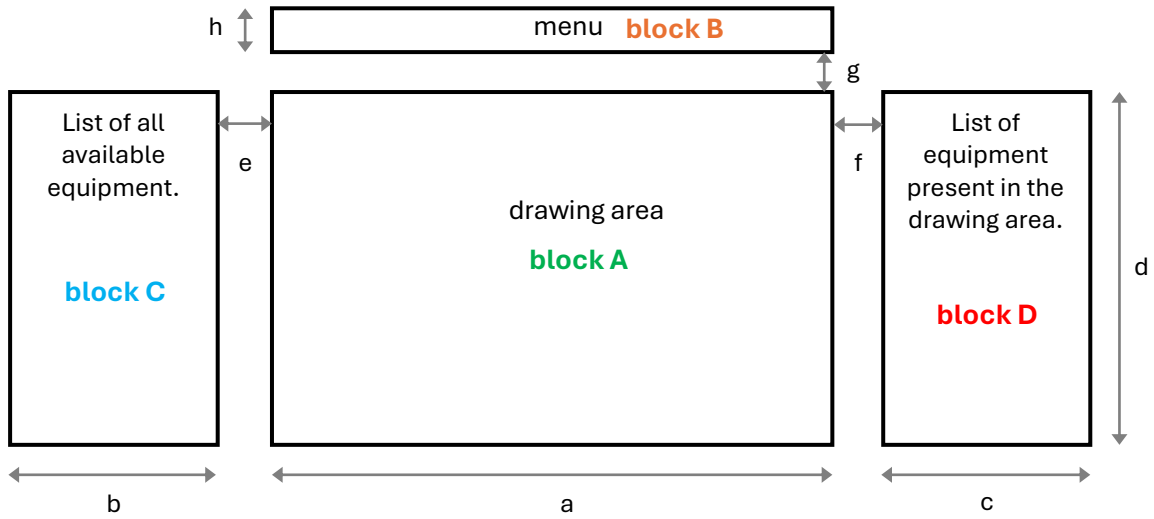

Figure 3-13. Interface diagram.

The user interacts using the keyboard and mouse. The ChemDoodle sketcher (https://web.chemdoodle.com/demos/2d-sketcher) is a good source of inspiration. The goal is to have an intuitive interface working similarly to ChemDoodle. For the chemical engineering part, the ProSimPlus3 software is a good source of inspiration, although the interface is archaic. The interface is divided into 4 blocks (Figure 3-13):

- **block A** : drawing area where the process flow diagram is drawn by the user.
- **block B** : menu for performing classic actions like open, save, or export diagrams, as well as changing the editing mode.
- **block C** : list of equipment that can be added to the drawing area, including predefined annotations.
- **block D** : window displaying information about the diagram (e.g., list of equipment).

The blocks have a well-defined size and spacing (highlighted in the diagram) to fit well on a Moodle page. Attention must be paid to the interface colors for colorblind users (use of red and blue, for example).

Provide three types of interfaces:

Viewer : Only the drawing area, non-modifiable by the end user. Allows the display of existing drawings stored in a database or a text field on the page, for example.

Sketcher : Complete interface described below.

Student : Interface described below but removing some buttons from block B's menu. Remove open, save, export.

3.4.2.2.1 Block A : the drawing area.

This is the main interface of the tool. The user draws the process flow diagram in this block. It is possible to add equipment, remove them, and connect them with pipes or cables.

The drawing area is described by a grid (invisible or visible depending on whether the option is enabled). This means that elements cannot be placed randomly but at specific coordinates. Additionally, the dimensions of the elements must match one or more grid cells. The drawing area is virtually infinite. It is possible to move the view of the drawing area by holding down the middle mouse button. Zooming in/out is possible with the scroll wheel.

Equipment is represented as vector images. Connectors are represented by perpendicular jagged lines with arrows. These connectors should, as much as possible, not cross each other. The lines are black, and the background color is white.

Interaction with the drawing area and the display of graphic elements on the screen depend on the active editing mode. The editing modes are:

- Add (add equipment or predefined annotations)
- Connect (connect equipment)
- Select/Move (select/move one or more elements)
- Delete (delete one or more elements)
- Rotate (rotate one or more elements)

The mouse cursor changes shape depending on the editing mode.

3.4.2.2.1.1 Add Mode

The mode is activated by selecting equipment from **block C**. The mode allows placing equipment or an annotation in the drawing area. Equipment and annotations are placed differently.

3.4.2.2.1.1.1 Adding equipment

Once the equipment is selected from **block C**, the user places the object in the drawing area by left clicking. A representation of the equipment is drawn under the mouse cursor to guide the user during placement. Note: It is not possible to place equipment on another. The color of the equipment to be placed is different from the already present elements (e.g., gray). Once the equipment is placed, the editor remains in Add mode, allowing the same equipment to be added again.

3.4.2.2.1.1.2 Adding a predefined annotation

Once the annotation is selected from **block C**, the user places it on equipment or a connector. A representation of the annotation is drawn under the mouse cursor to guide the user during placement (Figure 3-14).

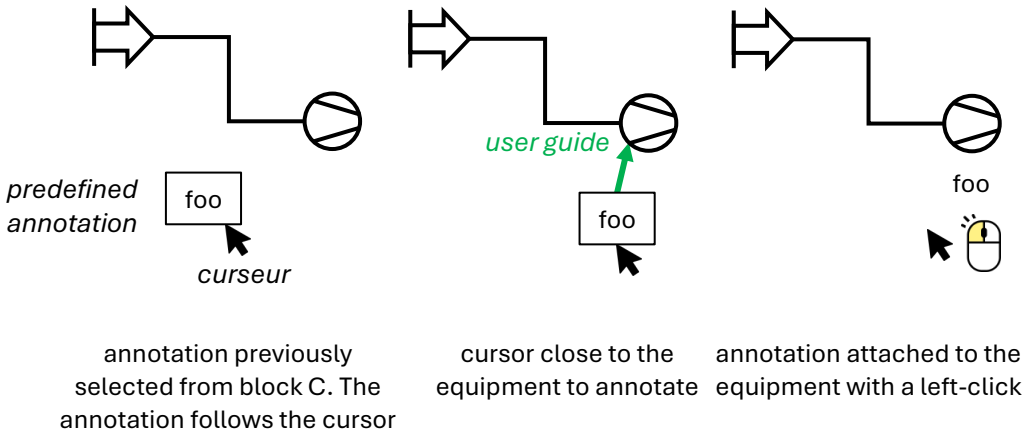

Figure 3-14. Adding a predefined annotation to equipment.

It is not possible to place an annotation freely. The annotation must be linked to equipment or a connector. Additionally, a constraint can be applied to the annotation to be placed. For example, if a predefined annotation can only be attached to specific equipment like a compressor (Figure 3-15).

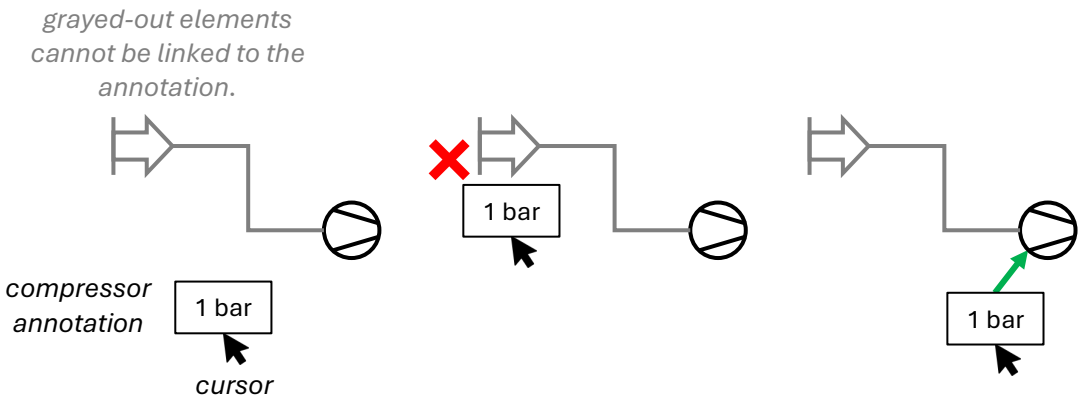

Figure 3-15. Adding a predefined annotation to a compressor. It is not possible to link the annotation to a connector or another type of equipment.

3.4.2.2.1.2 Connect mode

This mode allows connecting two pieces of equipment with a connector. The mode is activated by clicking on the "Pipes" or "Cables" button in **block B**. Once the mode is activated, the I/O points are represented by relatively small solid disks. The color of the points depends on the direction (entry, exit, or bidirectional). If the "Pipe" tool is used, only material I/O points are displayed. The same applies to the "Cables" tool and the information I/O points. Outside of Connect mode, the I/O points are not displayed.

3.4.2.2.1.2.1 Connection rules

Several rules exist to connect equipment. The same rules apply to both pipes and cables:

- Connecting an entry point to an exit point (or vice versa) is possible.
- Only connecting two I/O points of the same nature (material or information) is possible.
- An I/O point can have only one connection at a time.
- A bidirectional point can be connected to an entry point or an exit point.
- Two bidirectional points can be connected.
- It is not possible to connect the same I/O point.
- Connection on the same equipment is prohibited.
- It is not possible to connect two entry points. The same applies to two exit points.
- It is not possible to place a connection with only one I/O point.

3.4.2.2.1.2.2 Operation

When the mouse cursor is sufficiently close to an I/O point, it becomes active for connection. If several I/O points are near the cursor, the closest point is active. When the user left clicks on an active I/O point, it is defined as the 1st connection point. The connection can be canceled by right-clicking. The user can then define a 2nd connection point in the same way as previously described. Once the connection is made, the editor remains in Connect mode to create a new one. Figure 3-16 provides an overview of the expected behavior.

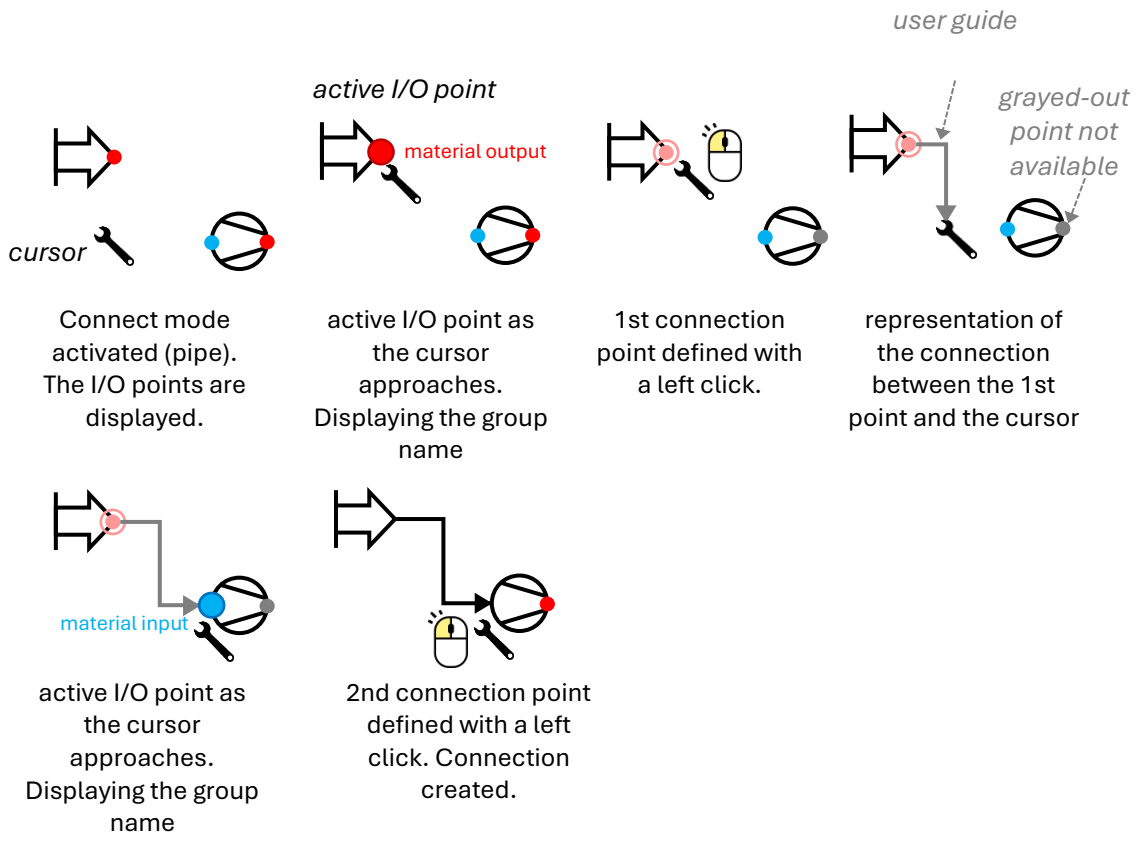

Figure 3-16. Connection between two pieces of equipment.

It is possible to redefine an existing connection. To do this, the user left clicks on one end of a connector. The end can be relocated to another I/O point.

3.4.2.2.1.3 Select/Move Mode

The mode is activated by clicking on the "Select" button in **block B**. It is the default mode when the application starts.

When the cursor passes over an element, it is highlighted (or another graphic element giving user feedback) indicating the active element for selection.

3.4.2.2.1.3.1 Selecting elements

Internally, a selection list contains the different selected elements (called the selection list). All elements can be selected (equipment, connectors, and annotations). The user can select each element individually by left clicking (Figure 3-17).

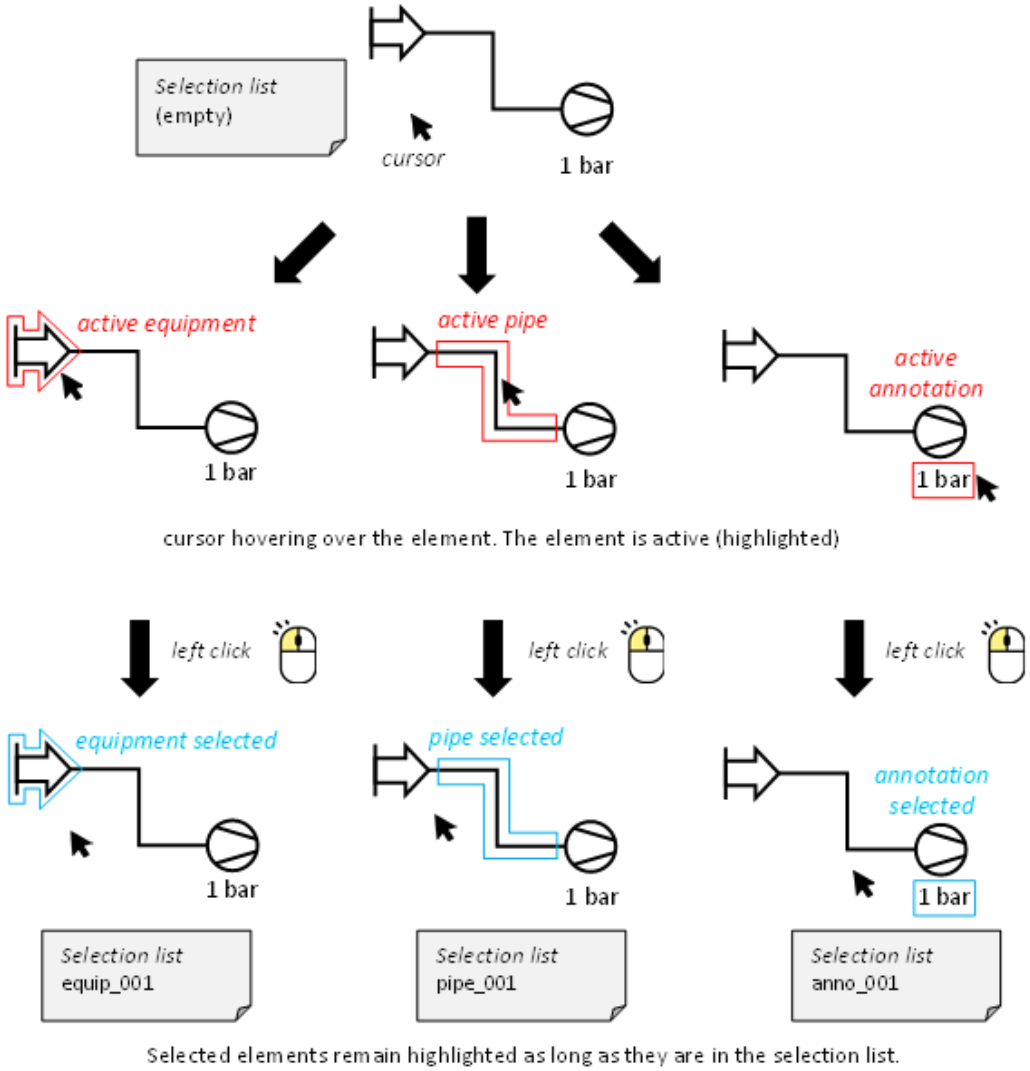

Figure 3-17. Selecting elements.

Dragging from the void defines a selection area (Figure 3-18).

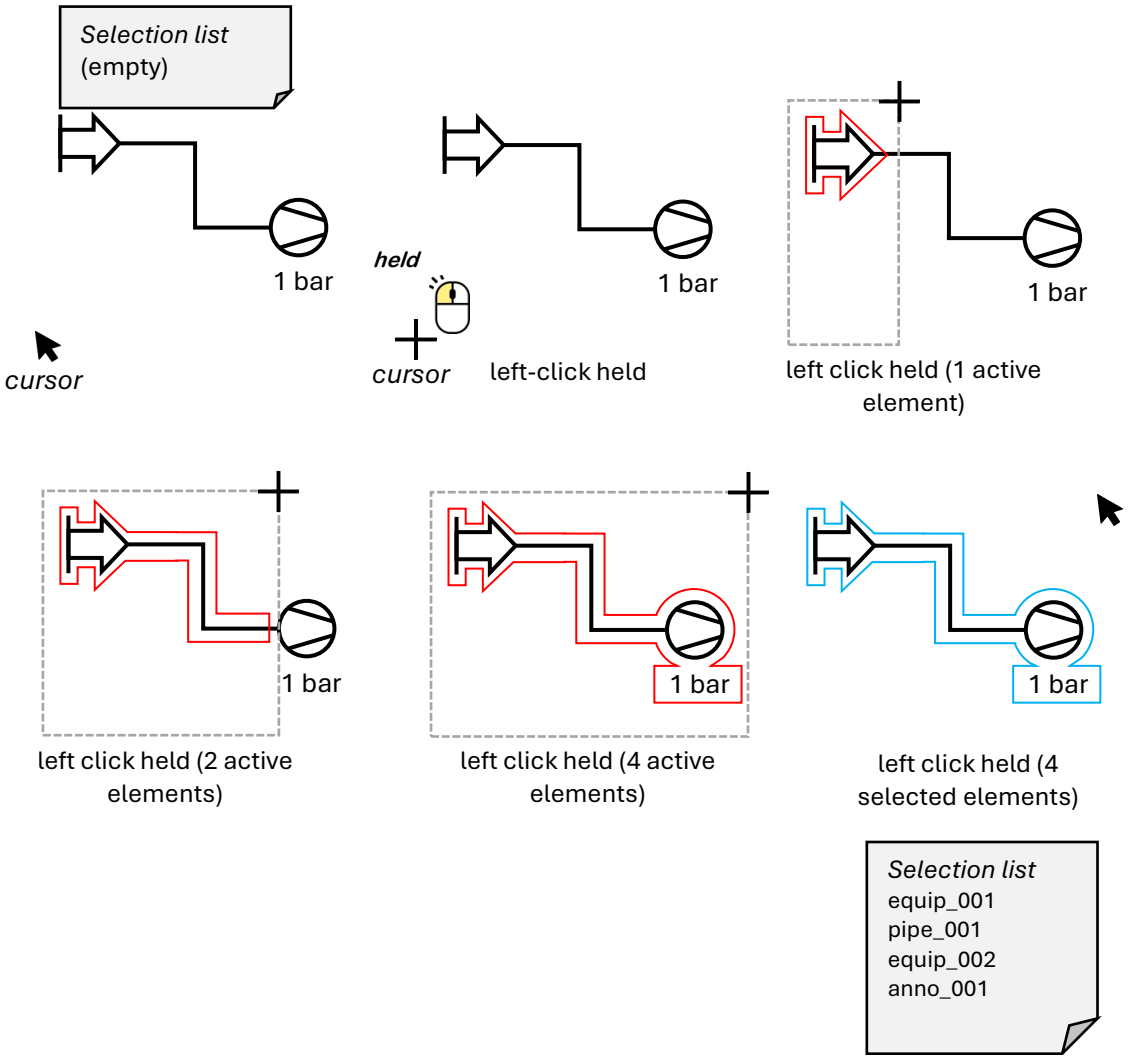

Figure 3-18. Selecting elements using a selection area.

Each left-click empties the selection list. If the user clicks on an unselected element, the list is first emptied, then the element is selected (Figure 3-19). A left click in the void clears the selection.

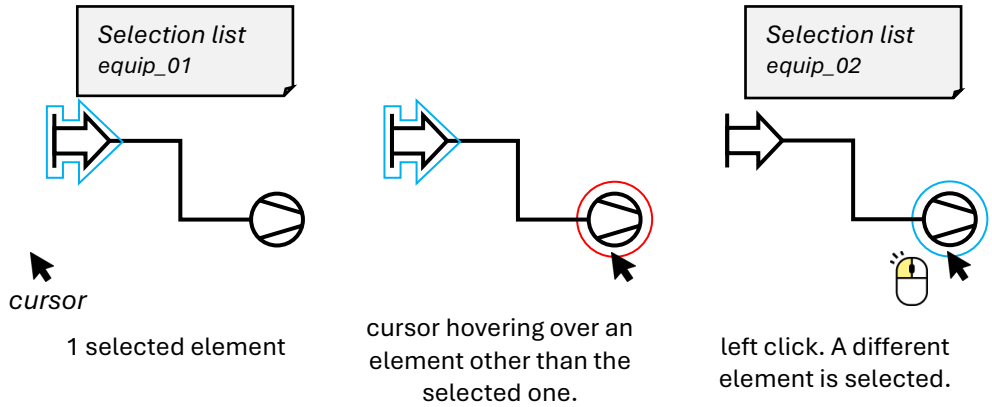

Figure 3-19. Selecting a new element.

The Ctrl key extends the selection (Figure 3-20).

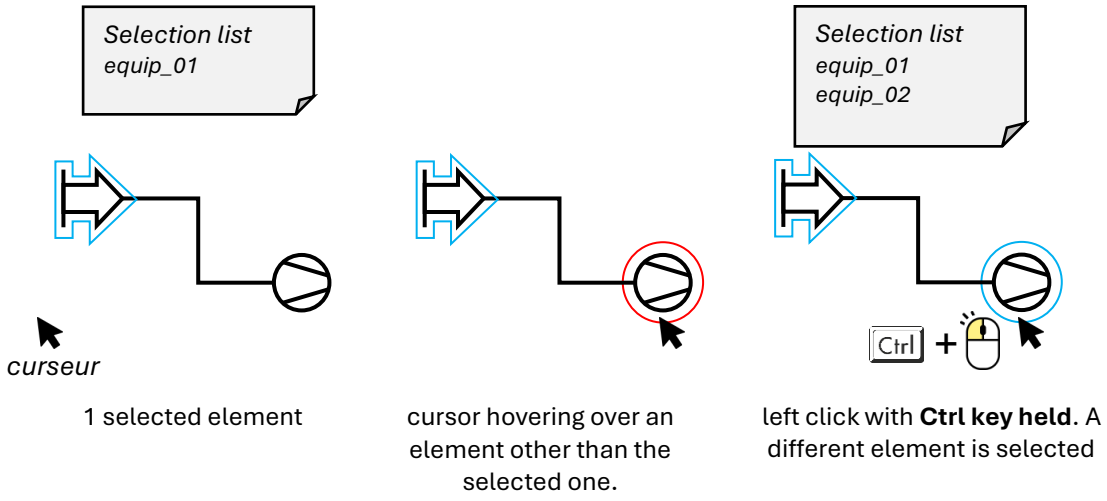

Figure 3-20. Adding an element to the selection list using the Ctrl key.

A left click in the void clears the active selection. It is possible to select one or more pieces of equipment from **block D**.

3.4.2.2.1.3.2 Moving elements

Movable elements include equipment, connectors, and annotations with a particularity for pipes, described later.

The user can move equipment or an annotation by holding down the left-click. As long as the click is held, the user can move the element with the mouse movement. If elements were previously selected, the user can move the selection with a held left-click (Figure 3-21). The Select/Move mode remains active until the user changes the mode.

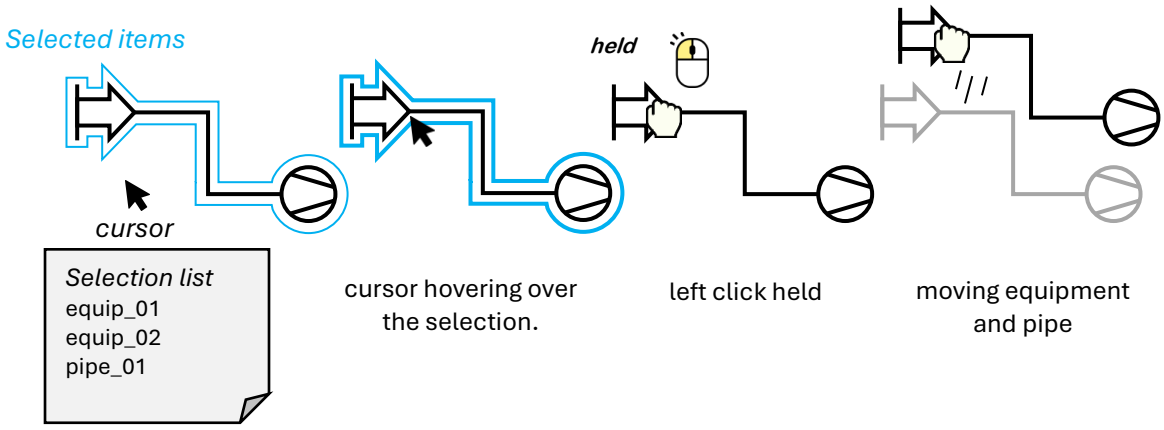

Figure 3-21. Moving a selection of elements.

The reaction of pipes during movement depends on the scenario. The pipe alone cannot be moved properly. Only sections can be moved and only along one axis (Figure 3-22).

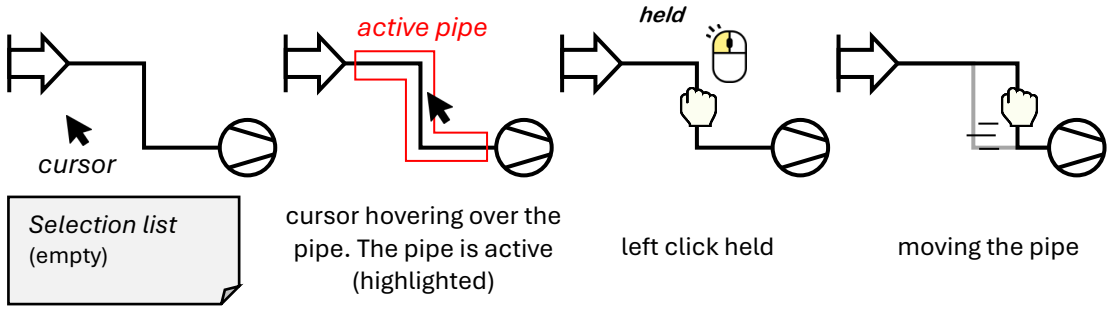

Figure 3-22. Moving a section of a pipe.

When moving equipment, the ends of the connectors must follow the equipment (Figure 3-23).

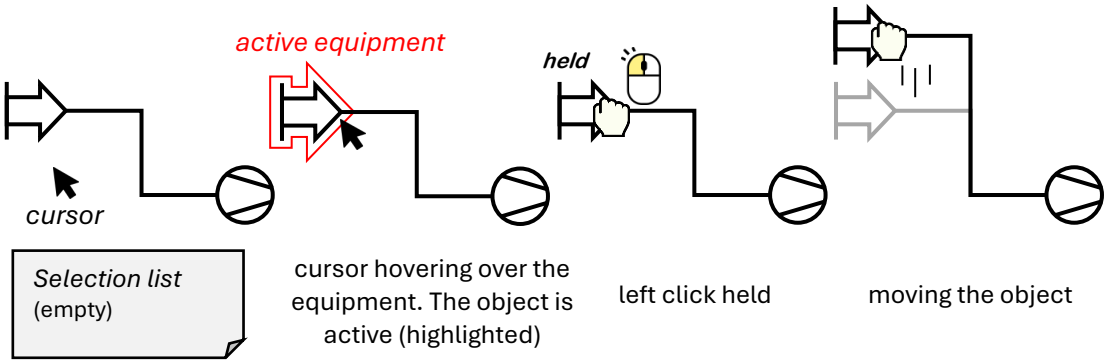

Figure 3-23. Moving equipment with a connector.

3.4.2.2.1.4 Delete mode

The mode is activated by clicking on the "Delete" button in **block B**. The user can delete equipment, a connector, or an annotation by left clicking on the element to delete. By holding down the left-click, elements passing under the mouse cursor are deleted. If equipment with one or more connectors is deleted, the connectors are also deleted (a connector must connect two pieces of equipment).

The Delete mode remains active until the user changes the mode. The Del key deletes the selection in Select/Move mode.

3.4.2.2.1.5 Rotate mode

The mode is activated by clicking on the "Rotate" button in **block B**. The user can rotate equipment or a selection by holding down the left-click.

The way the element or selection is rotated depends on the scenario. Some equipment can be oriented according to the 4 cardinal points (north, south, east, and west - Figure 3-24). On others, horizontal or vertical reflection is applied. Finally, some equipment cannot be rotated, such as symmetrical equipment. Details for each piece of equipment are provided in the Equipment List.

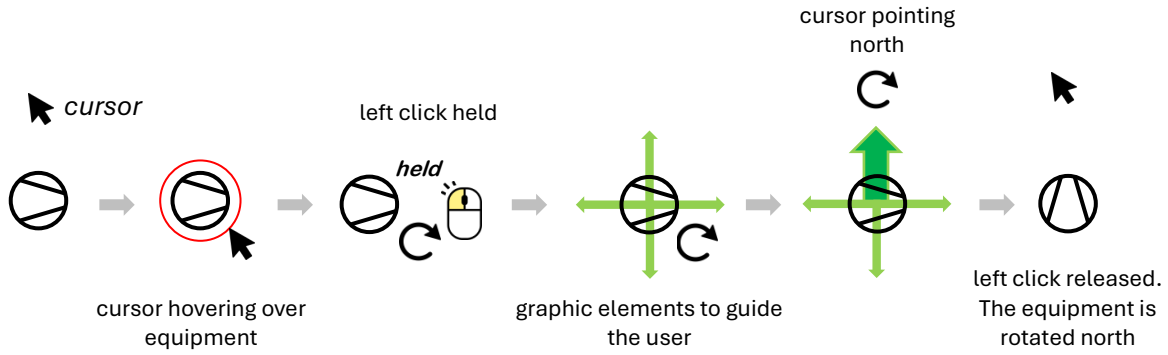

Figure 3-24. Rotating equipment.

If equipment to be rotated is connected to one or more connectors, they must be redrawn (Figure 3-25).

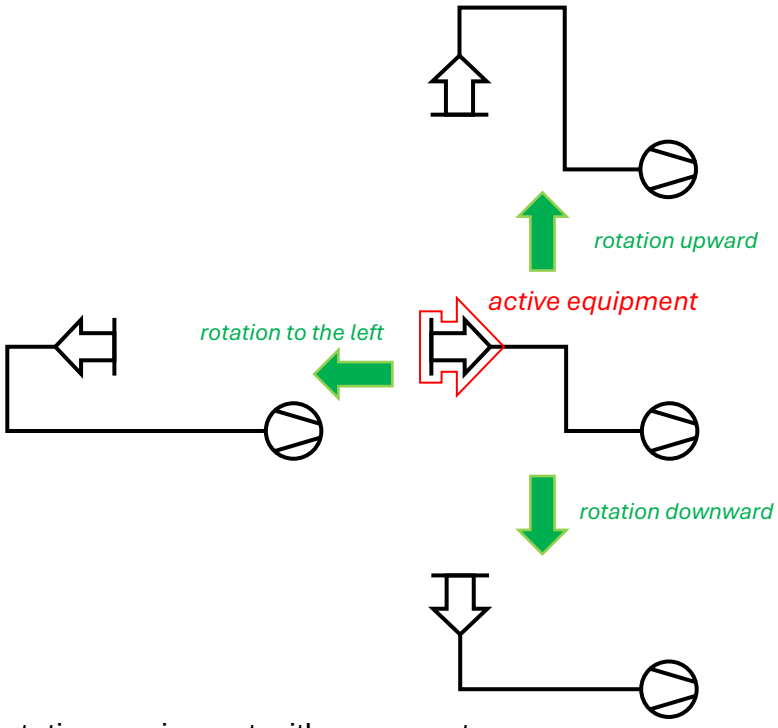

Figure 3-25. Rotating equipment with a connector.

Annotations cannot be rotated. The Rotate mode remains active until the user changes the mode.

3.4.2.2.2 Block B : the menu

Block B contains the application's menu bar. The menu bar is divided into several squares called "Buttons" (Figure 3-26). The buttons are square shaped with a logo inside.

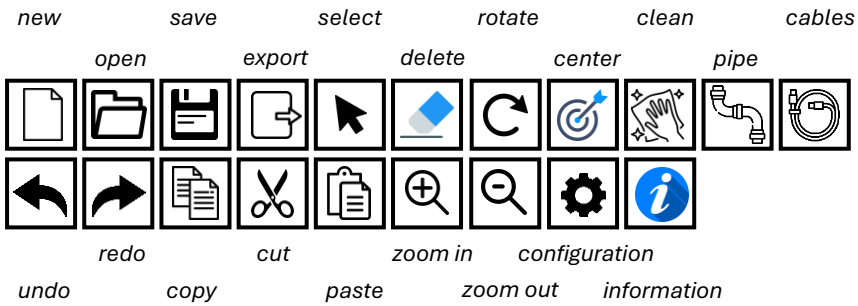

Figure 3-26. Example of a menu bar.

Interaction with the buttons is done with the left-click or a keyboard shortcut. When the mouse cursor hovers over a button, a text box appears to help the user (Figure 3-27).

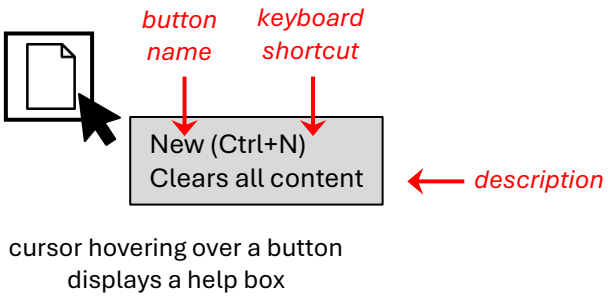

Figure 3-27. Help box of a button after cursor hover.

Below are the descriptions of all the buttons from Figure 3-26.

3.4.2.2.2.1 New

Clears the content of the drawing area in **block A**. The application asks the user to confirm the deletion.

Contextual help text box

- **Name** : New
- **Keyboard shortcut** : Ctrl+N
- **Description** : Clears all content from the drawing area.

3.4.2.2.2.2 Open

Loads a diagram from a file. The file format is to be determined. A file explorer allows the user to choose the file to load. When the user loads a file, the content of the drawing area in **block A** is cleared. The application asks the user if the drawing area's content should be cleared beforehand.

Contextual help text box

- **Name** : Open
- **Keyboard shortcut** : Ctrl+O
- **Description** : Loads a diagram. The current work will be erased.

3.4.2.2.2.3 Save

Saves the current content of the drawing area in **block A** to a file. The file format is to be determined. The file save is done using a file explorer. The file explorer opens each time the button is activated. A two-button solution can be considered: "Save" and "Save As".

Contextual help text box

- **Name** : Save
- **Keyboard shortcut** : Ctrl+S
- **Description** : Saves the diagram.

3.4.2.2.2.4 Export

Generates and saves an image of the diagram. The allowed formats are PNG and SVG (to determine if other formats are to be implemented). The file save is done using a file explorer. The file explorer opens each time the button is activated.

Contextual help text box

- **Name** : Export
- **Keyboard shortcut** : Ctrl+Maj+E
- **Description** : Exports the current drawing in image format.

3.4.2.2.2.5 Select

Activates the Select mode described in the Select/Move Mode section.

Contextual help text box

- **Name** : Select
- **Keyboard shortcut** : None or to be determined.
- **Description** : Selects one or more elements in the drawing area. *Briefly describe the operation.*

3.4.2.2.2.6 Pipes

Connects two pieces of equipment with a pipe (material transport). Details of the operation are provided in the Connect Mode section.

Contextual help text box

- **Name** : Tuyaux
- **Keyboard shortcut** : None or to be determined.
- **Description** : To be determined.

3.4.2.2.2.7 Cables

Connects two pieces of equipment with a cable (information transport). Details of the operation are provided in the Connect Mode section.

Contextual help text box

- **Name** : Câbles
- **Keyboard shortcut** : None or to be determined.
- **Description** : To be determined.

3.4.2.2.2.8 Rotate

Activates the Rotate mode described in the Rotate Mode section.

Contextual help text box

- **Name** : Rotation
- **Keyboard shortcut** : None or to be determined.
- **Description** : To be determined.

3.4.2.2.2.9 Delete

Activates the Delete mode described in the Delete Mode section.

Contextual help text box

- **Name** : Effacer
- **Keyboard shortcut** : None or to be determined. The Del key does not allow switching to Delete mode.
- **Description** : Deletes one or more elements in the drawing area. *Briefly describe the operation.*

3.4.2.2.2.10 Center

The view is centered on the drawing.

Contextual help text box

- **Name** : Center
- **Keyboard shortcut** : None or to be determined.
- **Description** : To be determined.

3.4.2.2.2.11 Clean

Reorganizes the layout of elements in the drawing area optimally. This button is useful if the initial positioning of the elements was not well thought out (e.g., crossed pipes, equipment too close, etc.).

Contextual help text box

- **Name** : Clean
- **Keyboard shortcut** : None or to be determined.
- **Description** : To be determined.

3.4.2.2.2.12 Undo

Cancels the previous action. All actions of the different editing modes (Add, Select, Move, Delete, and Rotate) can be undone.

Contextual help text box

- **Name** : Undo
- **Keyboard shortcut** : Ctrl+Z
- **Description** : To be determined.

3.4.2.2.2.13 Redo

Repeats an action that was previously undone with the Undo button.

Contextual help text box

- **Name** : Redo
- **Keyboard shortcut** : Ctrl+Y
- **Description** : To be determined.

3.4.2.2.2.14 Copy

Copies the selected elements to the clipboard.

Contextual help text box

- **Name** : Copy
- **Keyboard shortcut** : Ctrl+C
- **Description** : To be determined.

3.4.2.2.2.15 Cut

Copies the selected objects to the clipboard and then deletes them from the drawing area.

Contextual help text box

- **Name** : Cut
- **Keyboard shortcut** : Ctrl+X
- **Description** : To be determined.

3.4.2.2.2.16 Paste

Places the clipboard elements in the drawing area. Placement depends on the scenario. If the mouse cursor is in the drawing area (via the Ctrl+V shortcut), the clipboard elements are placed under the cursor. If the user clicks the "Paste" button, the elements are placed next to the existing ones. Each time elements are pasted they are in the selection list.

Contextual help text box

- **Name** : Paste
- **Keyboard shortcut** : Ctrl+V
- **Description** : To be determined.

3.4.2.2.2.17 Zoom in

Increases the size of the elements drawn in the drawing area.

Contextual help text box

- **Name** : Zoom in
- **Keyboard shortcut** : None or to be determined.
- **Description** : To be determined.

3.4.2.2.2.18 Zoom out

Reduces the size of the elements drawn in the drawing area.

Contextual help text box

- **Name** : Zoom out
- **Keyboard shortcut** : None or to be determined.
- **Description** : To be determined.

3.4.2.2.2.19 Settings

Configures the application. *To be detailed.*

Contextual help text box

- **Name** : Settings
- **Keyboard shortcut** : None or to be determined.
- **Description** To be determined.

3.4.2.2.2.20 Information

Provides information and help about the application (on a web page?).

Contextual help text box

- **Name** : Information
- **Keyboard shortcut** : None or to be determined.
- **Description** : To be determined.

3.4.2.2.3 Block C : the equipment list

All equipment that can be placed in the drawing area is accessible from **block C** (Figure 3-28). Equipment is grouped by category (e.g., Supply, Absorbers, Heat Exchanger, Mixers, etc.). Each category is represented by a dropdown menu. A search bar is available to search for equipment by typing its name.

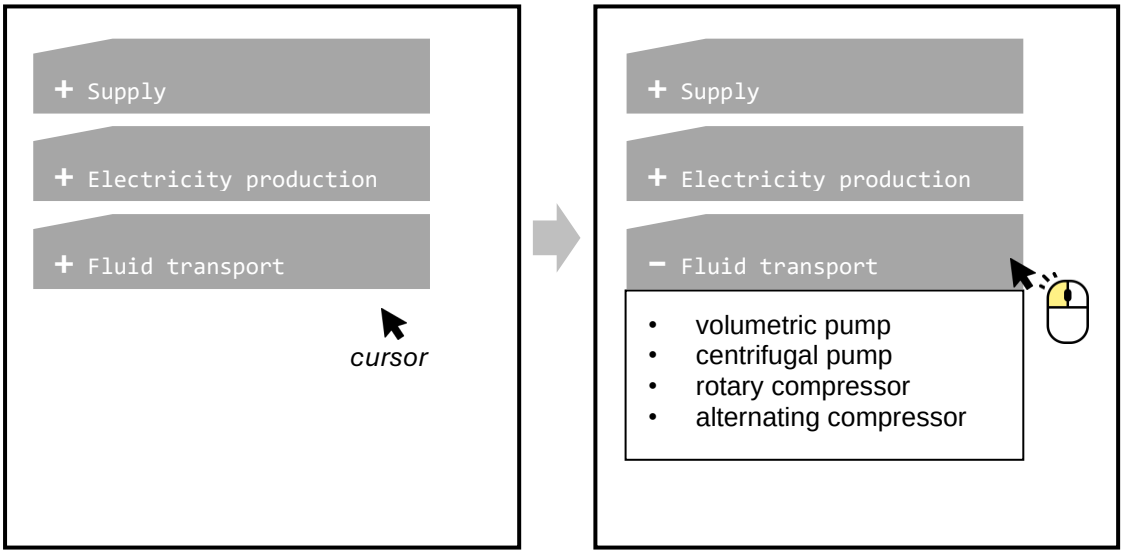

Figure 3-28. Block C menu.

3.4.2.2.4 Bloc D : Information

Block D contains information related to the diagram.

The block contains the list of equipment in the current diagram. It is possible to select equipment with a left click from this block. Each left-click empties the selection list. If the user clicks on an unselected element, the list is first emptied, then the element is selected. It is possible to extend the selection by holding down the Ctrl key. Holding down the Shift key allows selecting multiple consecutive pieces of equipment in the list.

Additional features can be considered in this block (e.g., contextual help).

3.4.2.3 Other features

The application is available in French and English. The application loads the default equipment list in XML, JSON, or another format at startup. It is possible to specify a custom equipment list to the application, allowing redefining equipment or adding predefined annotations, for example.

The application allows exporting in the SDF format by adapting it. Each piece of equipment is represented by an atom, and the pipes and cables are represented by single or double bonds.

Right-click displays a context menu.

Displaying equipment names.

Interface between the application and Moodle.

3.4.2.4 Keyboard shortcuts list

Ctrl+N: clears the drawing area (see New button)

Ctrl+O: opens a file (see Open button)

Ctrl+S: saves the current drawing (see Save button)

Ctrl+Shift+S: (see Save button)

Ctrl+Z: undoes the previous action (see Undo button)

Ctrl+Y: redoes an action undone by Ctrl+Z (see Redo button)

Ctrl+A: selects all elements in the drawing area

Ctrl+C: copies the selection elements to the clipboard (see Copy button)

Ctrl+X: copies the selection elements to the clipboard and then deletes the selection (see Cut button)

Ctrl+V: places the clipboard elements in the drawing area (see Paste button)

Ctrl + left click: extends the selection (see Select/Move Mode section)

Del (Delete): Deletes the selected elements from the drawing area (see Delete button)

3.4.2.5 List of equipment

To be established.

3.4.3 Interface and application

At the time of writing, the process diagram editor is under development and the initiation of testing, based on the provided specifications (Figure 3-29). The current phase is focusing on the project's foundational framework, creating a minimal drawing tool that supports SVGs for equipment, inputs/outputs, and connectors, along with implementing "double data-binding" for synchronizing the "graph-drawing" and "graph-data." The electronic format to store the diagram is also in development and testing for implementation with a soft grading system.

This groundwork is essential for the modular addition of features such as annotations and extra toolbar tools. This phase will extend until mid-August to early September, after which the process diagram editor plugin will continue to an "early alpha" version with weekly iterations. This structured approach ensures a systematic transition from research to development, aiming for a user-centric tool refined through continuous feedback and improvement.

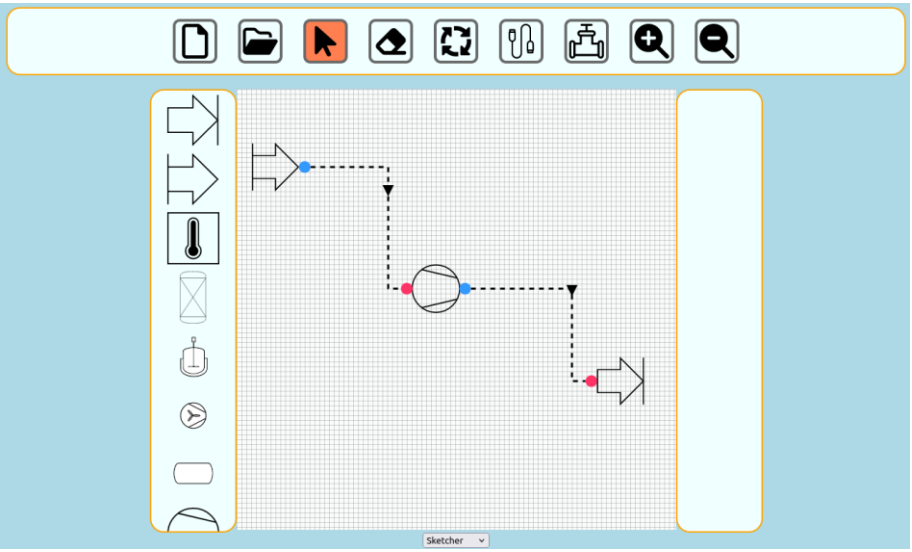

Figure 3-29. First version of the process diagram editor.

3.4.4 Summary

Although delays were encountered during the project launch, the development of the process diagram editor is now progressing well. The project is expected to enter a cycle of testing and further development towards the end of summer 2024. Following this phase, the editor is scheduled for delivery at the beginning of the autumn 2024. Once delivered, its integration into a TinyMCE plugin is anticipated to be fast, with the aim to have it deposited on the Moodle plugin repository and available during the autumn.

4 Question generator pGA-GTM/SVM

4.1 Motivation

The creation of relevant and varied question banks represents a significant task for educators. Traditionally, this task relies on a sequential approach where each question is individually written, even when there are variations on the same theme. Although natural and straightforward, this methodology is labor-intensive and time-consuming, and improving its efficiency is necessary.

Therefore, we propose to develop an innovative tool for drafting questions as concepts and applying them to concrete chemical examples. These examples should be chosen based on the chemical context to ensure relevance and applicability. This approach enables the generation of multiple concrete questions from a single conceptual framework, reducing redundancy and promoting a deeper understanding of the subject matter. Such a tool would allow educators to focus more on the pedagogical aspects of question design rather than the mechanical process of question generation.

An example scenario of the above can be the following. A teacher has a lecture on naming nitrogen containing heterocycles. The concept of the question would be: “Name the following structure”. The actual structure and expected answer should be computed based on a random selection of instances in a database of chemical structures. One of the challenges is how to let the teacher intuitively define a request for this database to generate relevant cases to the topic of his quiz test.

We propose to express this chemical context as chemical space maps, using Generative Topographic Mapping (GTM) technology [12]. This visual and analytical tool allows users to explore and interpret complex chemical spaces in a more intuitive way. Users can then select which area of chemical space to use for creating questions, ensuring they are grounded in real, contextual chemical knowledge.

The aim is to facilitate the process of creating question banks for educators, particularly in cases where it is necessary to generate questions on similar types of molecules or reactions. This tool aims to reduce the workload associated with manually designing each question by enabling the rapid generation of questions containing

molecular structures similar to a given reference structure. By automating the generation of these questions, educators can maintain high standards while significantly reducing the time required for this task.

GTM is a dimensionality reduction technique that allows transitioning from a space of chemical compounds to a 2D latent space while preserving distance relationships in this latent space and the similarity between chemical compounds. This method thus allows mapping the chemical space, transforming complex data into a simpler and more interpretable 2D graphical representation, as detailed above (see 2.2.4).

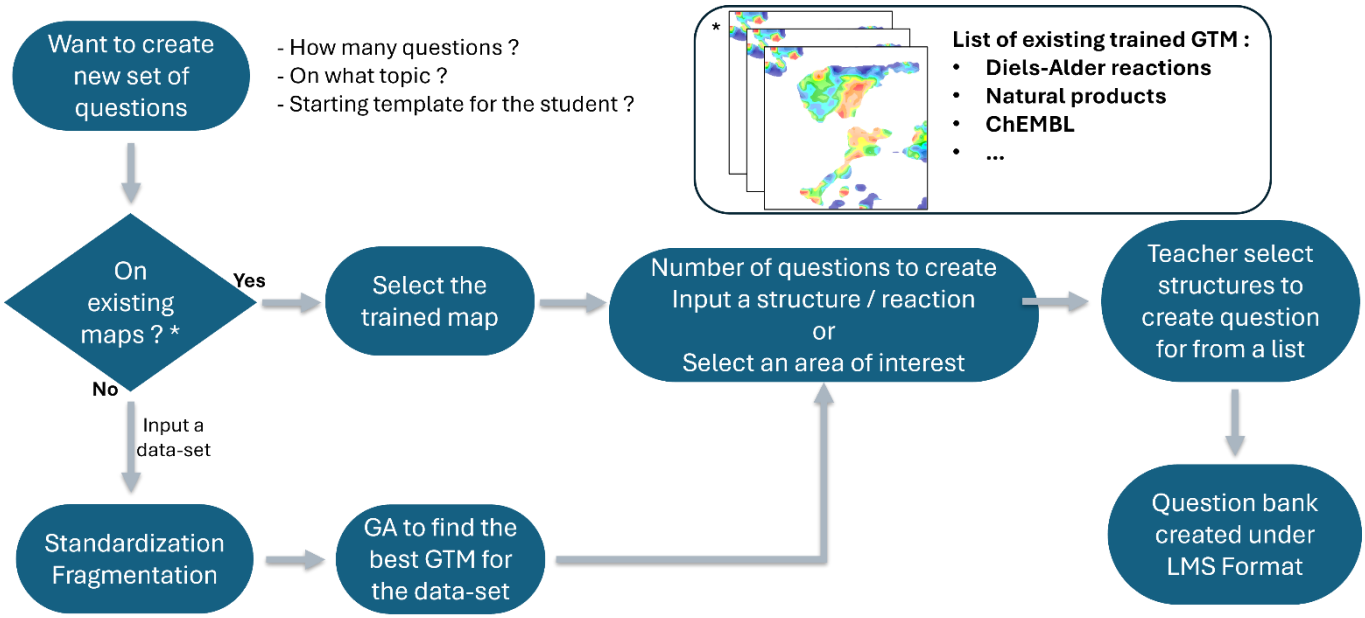

Figure 4-1. Different scenario for automatic question generation based on the concept of GTM. The GTM can be pre-existing, if not it needs to be generated with minimal expertise. The generative capacities of the map are used to produce questions relevant to the needs of the teacher.

In this segment of the project, the objective is to prepare a tool to help educators identify the most relevant areas for generating questions in three different ways (Figure 4-1):

- Using a set of pre-prepared and annotated maps, where educators can easily target areas of the chemical space based on their pedagogical needs and the subjects they wish to address. These maps are an essential resource, providing a structured and systematic way to explore the chemical space.
- Allowing educators to input a chemical structure, projected onto the map and used as a starting point for generating questions on chemical analogs.

- Enabling educators to bring their datasets and use the map of their personalized chemical space. This new map will be available to the educator alongside the default ones for use in the same way as the previous two scenarios. This flexibility allows educators to adjust the tool to their unique requirements, ensuring maximum relevance and effectiveness.

The challenge is that GTM requires selecting values for the method's free parameters, necessitating expertise to generate these personalized maps. We developed a new Genetic Algorithm (GA) [13] to overcome this challenge and to produce high-quality maps without expert intervention, as introduced above (see 2.3). This algorithm iteratively optimizes the meta-parameters, guided by a fitness function that ensures the generated maps accurately represent the chemical space. For this reason, we developed new software tools to be easy to install and use on the most common computing architectures (Linux, Windows, Mac). By design, this tool requires minimal know-how. It is planned to develop ad hoc user interfaces in order to reduce the minimal technical expertise. This new GA has been tested and validated on sets of QSAR test cases where it could be compared to standard implementations. The test cases are developed later.

In practice, the purpose of a map is to propose chemical structures analogous to those located in a region defined by the user. Educators can then feed their question concept with these structures, to make sure that the generated questions are relevant. The map retrieves chemical structures, and the educator's interface generates questions that are stored in question banks to populate a Learning Management System (LMS) such as Moodle. By automating and easing the question generation process, this tool empowers educators to focus on teaching and engaging with students, rather than being bogged down by repetitive tasks.

The scope of the work completed focuses exclusively on the GA component. The technical implementation of the question generator is the responsibility of another project and another team. For this reason, it is out of the scope of this manuscript.

4.2 GA overview

In developing this genetic algorithm, we incrementally added features and made iterative improvements as we progressed. Our initial focus was on the GTM genetic algorithm GA (pGA-GTM). It was necessary to rewrite several units in Pascal Object [106], which facilitated the application of GTM within our specific context and ensured proper functionality and seamless integration. Subsequently, we turned our attention to the implementation of the SVM GA (pGA-SVM), leveraging the capabilities of the libsvm library [118]. The integration of libsvm posed additional challenges, necessitating the creation of an interface in Pascal Object to enable its usage within our software.

As for the libSVM-GA [77] it has been inspired of, meta-parameters of the SVM/GTM and the Descriptor Spaces (DS) are optimized at once. Indeed, as the meta-parameters are often dependent of the DS, they need to be optimized considering the DS.

The development process involved continuous adaptation and refinement, guided by ongoing feedback and emerging requirements. This approach ensured that our GA evolved in response to the project's dynamic needs, resulting in a robust and flexible solution. This iterative methodology facilitated the progressive enhancement of the genetic algorithm while ensuring rigorous testing and optimization, contributing to the overall efficacy and reliability of the final implementation.

In the following sections, I will detail the various architectures we developed and the several genetic operations that were used (4.2.1), and the specific enhancements made during each development version. Then, I will introduce a test case (4.2.2), where we compared the usage of pGA-SVM to libsvm-GA [77] on ChEMBL [119–122] targets under regression and classification tasks for the prediction of activity (pChEMBL, [123]). Then, we will see how the modification of the GA inner parameters affects the results and computing time on a specific test case (4.2.3). Once we selected the best suited parameters, we will compare with the results obtained from 4.2.2. Then, we will see how the GA was used for classification and regression tasks with prediction of antioxidant capacity (4.3). The documentation of the GA is provided in the Appendix 2: Manual of the pGA-GTM/pGA-SVM.

4.2.1 Description of the GA algorithm

A nonparallel generational version of the GA was initially constructed before going to a fully parallelized GA. This step is crucial for several reasons:

- It allows us to rigorously validate the fitness function to ensure it accurately evaluates the solutions generated by the algorithm.
- It enables us to refine the selection procedures, which are vital for maintaining genetic diversity and guiding the algorithm toward optimal solutions.
- It provides a controlled environment to experiment with genetic operations such as crossover and mutation, which are mechanisms for creating new candidate solutions.

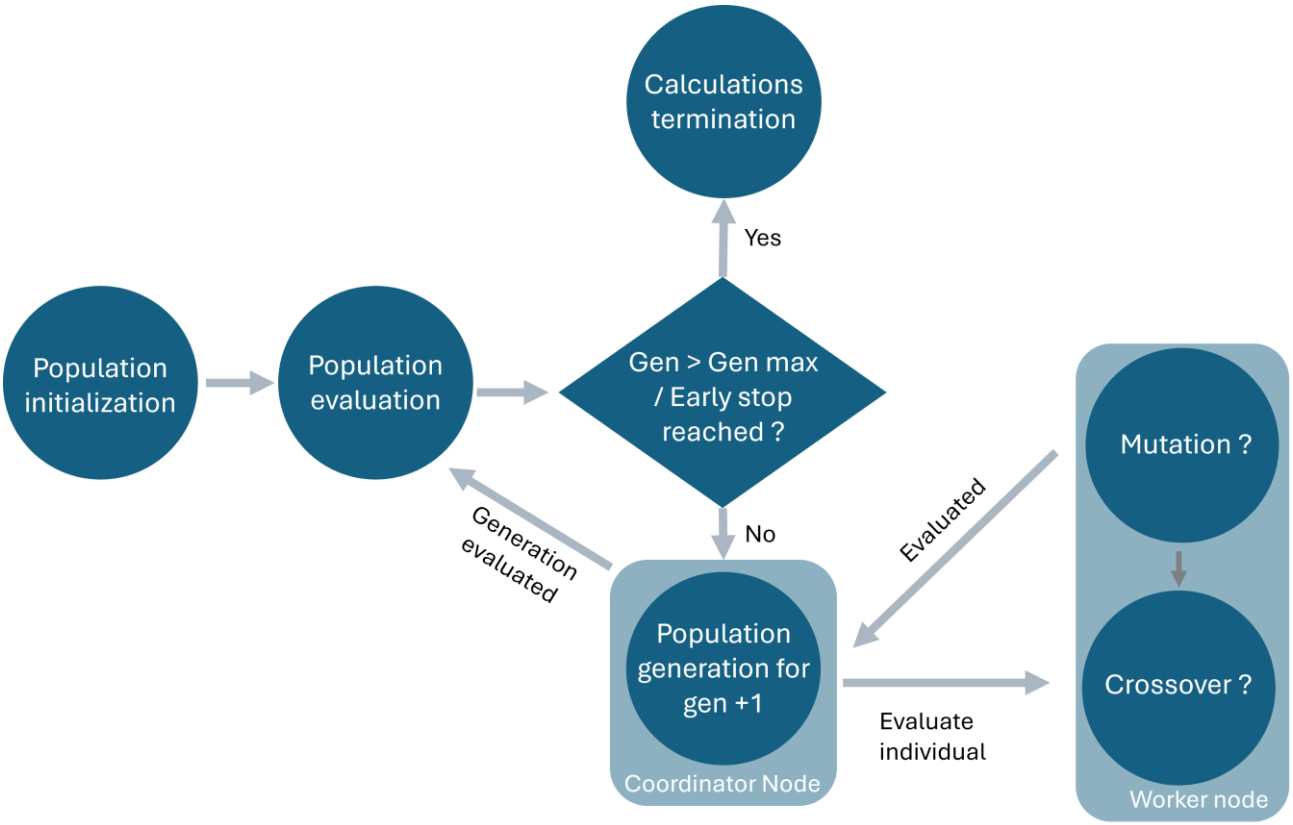

Figure 4-2. Base architecture developed.

By confirming the correctness and efficiency of these core components in a sequential context, we establish a robust framework that ensures a smoother and more effective transition to a parallelized implementation. This method helps minimize potential errors and optimize the algorithm's performance, ultimately leading to a more reliable and efficient system. Concerning the pGA-SVM and pGA-GTM, the only difference resides in the fitness function implementation, the remainder of the architecture being similar (Figure 4-2).

In these GAs, the population is initialized according to the parameters of the search space that the user inputs, and each individual is given a score according to the fitness function calculation. Then, the whole population is evaluated to get the mean and best score of the population. Afterwards, each individual in the population has a probability of undergoing various genetic operations. Once the whole population has been processed, the generation is complete, and the population evaluated again. The process goes on until the stopping criteria are met. As stopping criteria, we use both the maximum number of generation and the number of generations without improvement of the best individual of the population.

First, will first discuss the two fitness functions that are used, as the method used in these functions plays a role in other genetic operations. A fitness function evaluates the suitability or quality of a solution in optimization algorithms, guiding the search for optimal solutions.

Both fitness functions aim at creating models that can predict new instances rather than just fitting the training data. We use a M-times repeated N-folds cross-validation (XV) with reshuffling of training set instances [124, 125]. In the pGA-SVM test case (4.2.2), we used 12 times 3-fold XV procedure. A reference implementation by Horvath et al. [77] is configured the same way for benchmarking. However, by default, we propose to use the values of M=3 and N=3, which is less biased toward the composition of the dataset but estimating performances with a larger variance. Of course, these details of the validation procedure are left to the preferences of the end users and other choices can be met in the literature [125, 126], depending on the context.

The fitness function used in this optimization considers the average XV performance. However, the “best” model observed at a given moment of the GA is only better for a given repartition of the data between the training and test sets of the cross-validation. Therefore, models that perform less well may be equivalent or even better than the top-performing ones. To identify models that are statistically indistinguishable from the “best” model, a threshold is defined using the repeatability of the performances across a moderate repetition of the cross-validation procedure.

For both pGA-SVM and pGA-GTM, we reduced as much as possible the number of disk accesses, relying on function calls instead of system calls, in order to speed up the calculations.

For SVM or SVR, using libsvm software [118], the goal of the pGA-SVM is to find the chromosome leading to the best classification or regression model. The chromosome encodes for the DS to use, the value of the cost parameter, the kernel to use, the gamma, coeff0 values, and the epsilon value for the regression. When the chromosome needs to be evaluated, the value of the gene needs to be “translated” to libsvm parameters.

As the DS chosen may modify highly the vector representation of each compound of the dataset, we needed to address the variability induced to the gamma parameter. The gamma is a kernel parameter scaling as the standard deviation of the instances in the data space. Therefore, to be integrated in a GA, it needs to be standardized. As in libsvm-GA [77], a preliminary "gamma factor" is encoded in the gene. To translate it to libsvm parameter, it is divided by the average Euclidean distance or dot product values over training set pairs, depending on the kernel used, allowing for a reasonable range of gamma. These metrics are computed for each DS at the initialization step of the GA. The epsilon parameter for regression tasks is in the same units as the target property to model. For this reason, the actual epsilon value is calculated by multiplying the epsilon parameter by the standard deviation of the training property values. The cost parameter is encoded using a log-scale conversion and the actual cost value is obtained by taking the exponential of the cost parameter from the chromosome.

The objective of this GA is to simultaneously optimize the meta-parameters of SVM/SVR problems, alongside determining the DS, with SVM parameters being contingent upon the chosen DS. If any model fails to fit during training (R2 XV_M < minperfstop), it is immediately discarded (the solution gets a FS of 0.00), to speed the computation process, by discarding “bad” models. By default, minperfstop is equal to 0.

For that reason, the FS is calculated as follows:

- For each M N-fold XV: shuffle the training set and N-fold XV (default: M=3, N=3).
- Compute R^2/BA for this XV step.
- At the end of the M N-fold XV: Calculate the mean $\langle BA \rangle$ or $\langle R^2 \rangle$ and standard deviation σ of the M N-fold XV: $\sigma(BA)$ or $\sigma(R^2)$.
- The FS is defined as $\langle R^2 \rangle - 2\sigma$ for the SVR and $\langle BA \rangle - 2\sigma$ for the SVC.

For GTM, using the in house GTM implementation in Pascal [127], the goal is to find the chromosome leading to the manifold best representing the dataset. The chromosome encodes for the DS to use, the number of RBFs, the width of the RBFs, and the regularization parameter. As for the pGA-SVM, the genes need to be translated to GTM parameters. The number of RBFs is taken from equation (2-9) where nb_{mol} is the number of molecules in the frameset, and $gene_{RBF}$ is the value of the gene encoding for the number of RBFs.

$$nRBF = Min(floor\left(\frac{nb_{mol}}{2}\right), gene_{RBF} \times 10) \quad (4\text{-}1)$$

The regularization parameter is equal to 10 to the power of the gene value, and the width of the RBFs is taken directly from the value of the gene. The number of nodes is taken as 25 times the number of RBFs. Unlike in the previous implementation of the in-house GA, we compute the PCA that are used to initialize the manifold once at the initialization step of the GA for a given DS, and not for each FS calculation, saving computation time.

The difference with the pGA-SVM lies in the fact that the pGA-GTM is a multi-objective optimization. Indeed, to give a FS to each manifold, they are evaluated on each property given by the user (on each of the landscape created).

The value of the gene encoding for the DS is used as frameset for unsupervised training of the manifold. To create a landscape, the user has the option to provide a scoring set of annotated compounds, projected on the manifold and used for the calculation of the FS. Otherwise, the compounds used for manifold creation will be projected on it for landscape creation.

For that reason, the FS is calculated as follows:

- The manifold is initialized and trained using the frameset.
- Frameset or scoring set are projected on the trained manifold.
- For each property given by the user: M x N-fold XV (by default: M=3, N=3)
- Compute R^2/BA for this XV step.
- At the end of the M N-fold XV: Calculate the mean $\langle BA \rangle$ or $\langle R^2 \rangle$ and standard deviation σ of the M $\sigma(BA)$ or $\sigma(R^2)$.
- For each P property compute a score $S = \langle R^2 \rangle - 0{,}5\sigma$ for the regression and $S = \langle BA \rangle - 0{,}5\sigma$ for the classification.
- The FS is defined as $\langle S \rangle$.

The FS is defined as a strict average of the modeling tasks for simplicity. However, it is also possible to optimize the Pareto front.

We will now discuss the genes encoding, the selection mechanism, and explore the mechanisms of mutation and crossover, the key genetic operations that introduce diversity and create new, potentially superior solutions. To discuss these processes, we will follow the workflow of the GA. By detailing each of these elements, we aim to provide a thorough understanding of the algorithm's operation.

To encode for each chromosome, we used real value numbers. The values chromosomes can take (ie. *search space*) are taken from a user input when launching the GA. At population initialization step, the chromosomes are initialized randomly, and evaluated.

Selection is an important step of the GA, as it determines which chromosomes will participate in the crossover and mutation steps. In this work, we chose to use a mix of two selection techniques. For each generation, we loop in the population, and for each individual C_i, they have a probability to crossover (Figure 4-3).

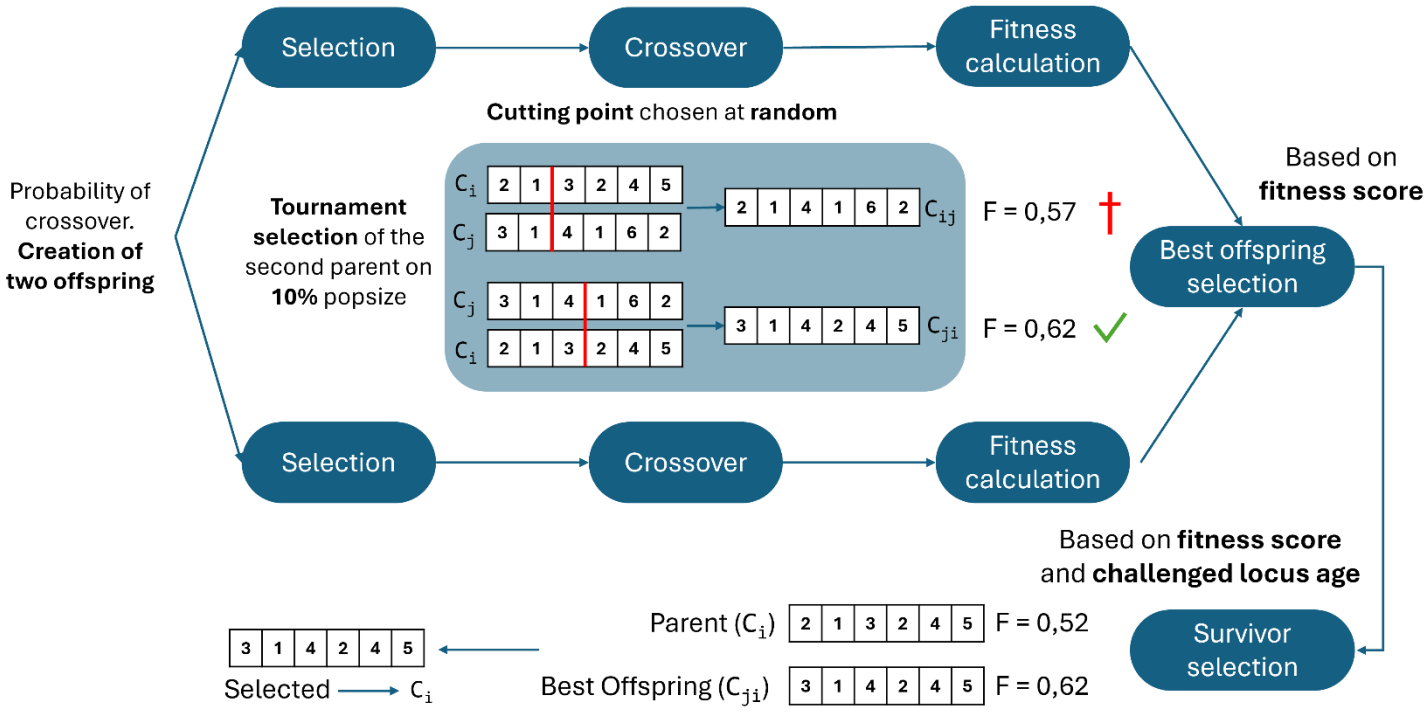

Figure 4-3. Summary of the crossover operation. Two offsprings are generated, and the best one compared to the chromosome C_i. The winner of the survivor selection is sent back to the population.

If they crossover, we choose one other individual C_j by tournament selection [128]. This pair (C_i,C_j) is selected for the crossover operation, and the fittest of their offsprings (C_{ij} | C_{ji}) will be kept for the following step. We chose to use a simple single

point crossover as it is the simplest approach and has shown good results in the literature [129, 130]. If the newly chromosome if not different from its 'parents', we allow for 10 tries to try and create a new different chromosome. The difference is computed by Hamming distances. For SVC and SVR, the linear (2-9) and RBF (2-10) kernels do not utilize the $coeff0$ parameter. Additionally, the linear kernel does not use the γ (gamma) parameter. When comparing two chromosomes that use the same kernel type, we don't increase their distance if they fall into one of these scenarios. After 10 tries, if the created chromosome is not different, we create a new random one. Once a chromosome is created, it is evaluated using the fitness function.

The fittest of the offsprings is then compared to C_i through a *survivor selection* procedure, where we account not only for the fitness of an individual (4-2), but also for its age and more specifically the difference between its age and the age of C_i (4-3).

$$\mathrm{C}_i = \mathrm{C}_{ix}, \qquad if\ \mathrm{C}_{i\ fitness} < \mathrm{C}_{ix\ fitness} \tag{4-2}$$

$$\mathrm{C}_i = \mathrm{C}_{ix}, \qquad if\ \mathrm{rand} < \mathrm{age}_{fact} \times \mathrm{fit}_{fact} \tag{4-3}$$

Where age_{fact} (4-4) and fit_{fact} (4-6) take in account the difference of age between C_i and C_{ix}.(where C_{ix} is the fittest of the offsprings (C_{ij} | C_{ji})).

$$\mathrm{age}_{fact} = \frac{\mathrm{C}_{ix\ age} - \mathrm{C}_{i\ age}}{\mathrm{max}_{age}} \tag{4-4}$$

Where $\mathrm{C}_{i\ age}$ and $\mathrm{C}_{ix\ age}$ are the generation at which the chromosomes have been created, and max age is computed from formula (4-5).

$$max_{age} = \max(10, \mathrm{round}(\max(100, max_{gen} \times 0.2) \times 0.1)) \tag{4-5}$$

Where max_{gen} is the maximum number of generations allowed by the user.

$$fit_{fact} = \frac{\mathrm{best}_{fitness} - \mathrm{C}_{i\ fitness}}{\mathrm{C}_{i\ fitness} - \mathrm{C}_{ix\ fitness} + \mathrm{Signif}_{fitnessDiff}} \tag{4-6}$$

Where $\mathrm{Signif}_{fitnessDiff}$ is a fitness function-specific value. In this work, we used a value of 0,5 and $\mathrm{best}_{fitness}$ is the fitness of the best individual in the population.

We believe that by implementing this survivor selection procedure, the trade-off between exploration and exploitation is well addressed. Indeed, formula (4-4) accounts for the amount a time a solution has stayed in the pool of solutions, and formula (4-6) allows a non-optimal solution to replace C_i. Indeed, as $C_{i\ fitness}$ approaches best_{fit}, the numerator decreases, and the denominator increases as $C_{i\ fitness}$ moves away from $C_{ix\ fitness}$., allowing new chromosomes that are close in the fitness score space to replace existing solutions.

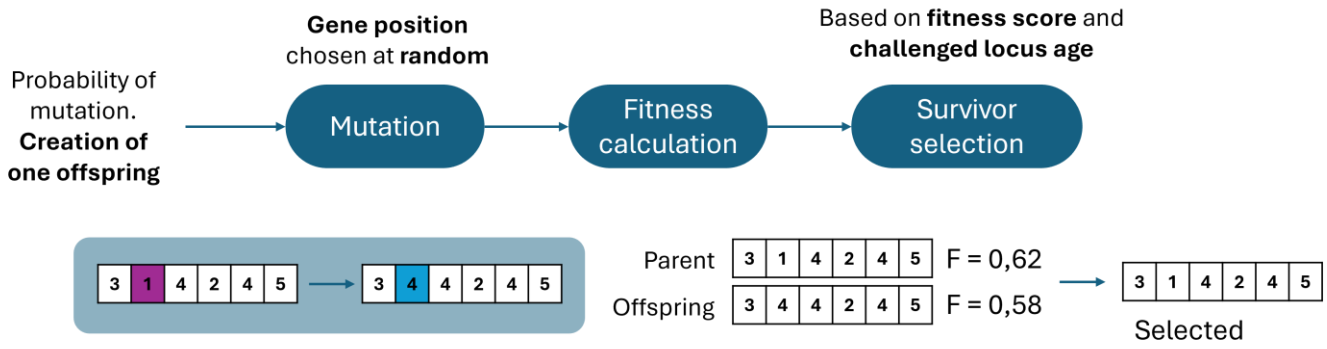

Figure 4-4. Summary of the mutation operation. Top: the gene is modified by a mutation and survives according to its fitness. Bottom: illustration of the different steps of the process.

Then, the given individual (C_i) has a probability to be selected for the mutation operation (Figure 4-4). Concerning the mutation, we chose to use a single gene mutation, chosen at random, which value is modified randomly. The offsprings is then compared to C_i through a *survivor selection* procedure, as for the crossover operation.

Once these steps are finished, the resulting chromosome is reintroduced into the population, and the process repeated for each chromosome of the population.

A loop threw each member of the population is called a generation. At the end of a generation, a backup of the population may be created under two formats. One where the command line is given for the user to recreate the model using the information of the chromosome, and another one which is machine readable, and used to restart the GA from the given pre-computed generation.

Once the stopping criteria has been reached, a BestPop folder is created. It contains n folders, one for each of the n first chromosomes of the last generation and

their corresponding models and statistics are computed. Each of these sub-folders contain models, statistics and predicted properties of the training or test set if provided.

More information about the pGA-SVM and pGA-GTM usage and outputs can be found in the Appendix 2: Manual of the pGA-GTM/pGA-SVM. For SVM, we used an adaptation of the work of Hsu et al. [131] where we first compute the XV scores according to the fitness function, before training the model on all of the training set. Then, the model is used on a test set if provided, or on the training set in the other case.

A summary of the whole procedure for the nonparallel GA may be found in the form of pseudo-code in Figure 4-5.

```
Initialize Population

while termination criteria not met do
    for each individual C_i of the population
        if crossover
          Select individual C_j by tournament selection
          Crossover individuals C_i and C_j, creating individuals C_ij and C_ji
          Select fittest offspring
          Survivor selection between C_i and fittest offspring, winner replace C_i
        if mutation
          Mutation point selected at random
          Gene is randomized, creating individual C_j
          Survivor selection between C_i and C_j, winner replace C_i
    end
    BackupPopIfNeeded
end
CreateModelsComputeStat
```

Figure 4-5. Pseudo code of the nonparallel GA.

Now that we've covered the essential genetic processes and selection procedures, we'll explore how we transitioned from our initial sequential implementation to several parallel implementations.

We first developed an initial version of a generational parallel genetic algorithm (Parallel GA v1, Figure 4-6).

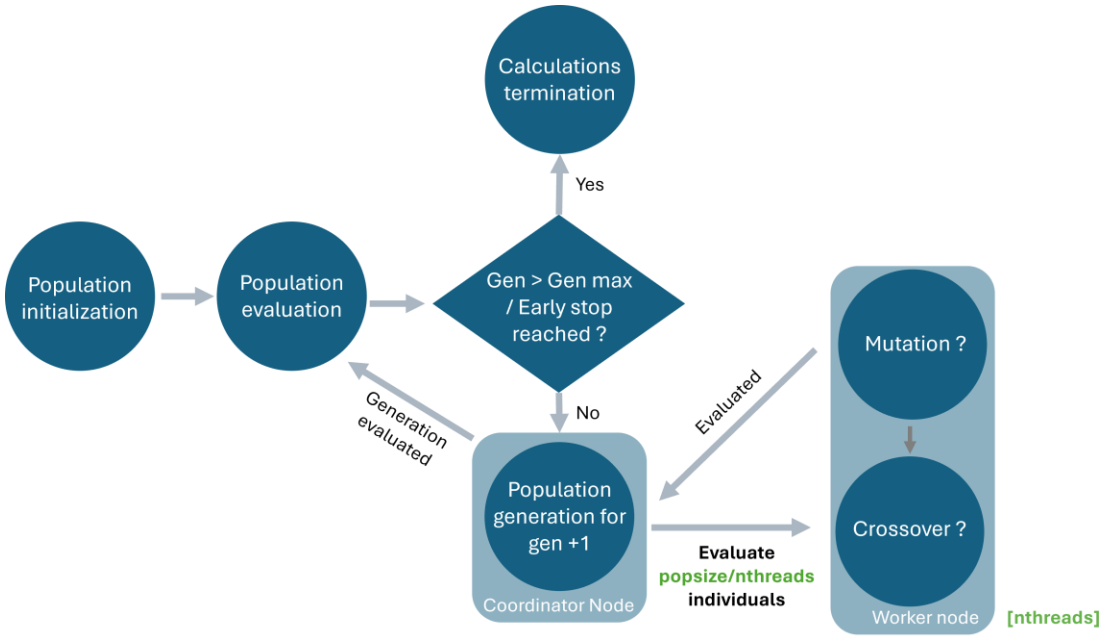

Figure 4-6. Architecture of the Parallel GA v1. The coordinating process fires worker processes, that operate on subset of gene population and compute the fitness of each gene. The coordinator node recovers the subsets and prepare for a next generation or terminates the process.

In this setup, each available thread was tasked with evaluating a subset of $populationSize/NbThreads$ chromosomes. However, this approach led to inefficiencies due to all individuals being dispatched to worker nodes simultaneously. This caused significant waiting times because some threads finished their computations earlier than others. As a result, the GA had to wait for certain threads to finish before proceeding to the next generation. A summary of the whole procedure for the Parallel GA v1 may be found in the form of pseudo-code in Figure 4-7.

```
LaunchThread {
for each individual Ci of populationSize/NbThreads
  if crossover
    Select individual Cj by tournament selection
    Crossover individuals Ci and Cj, creating individuals Cij and Cji
    Select fittest offspring
    Survivor selection between Ci and fittest offspring, winner replace Ci
  if mutation
    Mutation point selected at random
    Gene is randomized, creating individual Cj
    Survivor selection between Ci and Cj, winner replace Ci
}
Initialize Population

while termination criteria not met do
    for i=0 to NbThreads -1 do
      LaunchThread(populationSize/NbThreads)
    end
    BackupPopIfNeeded
end
CreateModelsComputeStat
```

Figure 4-7. Pseudo code of the parallel GA v1.

To answer this issue, we implemented a generational GA where we identified which chromosomes necessitated genetic operations (Parallel GA v2, Figure 4-8).

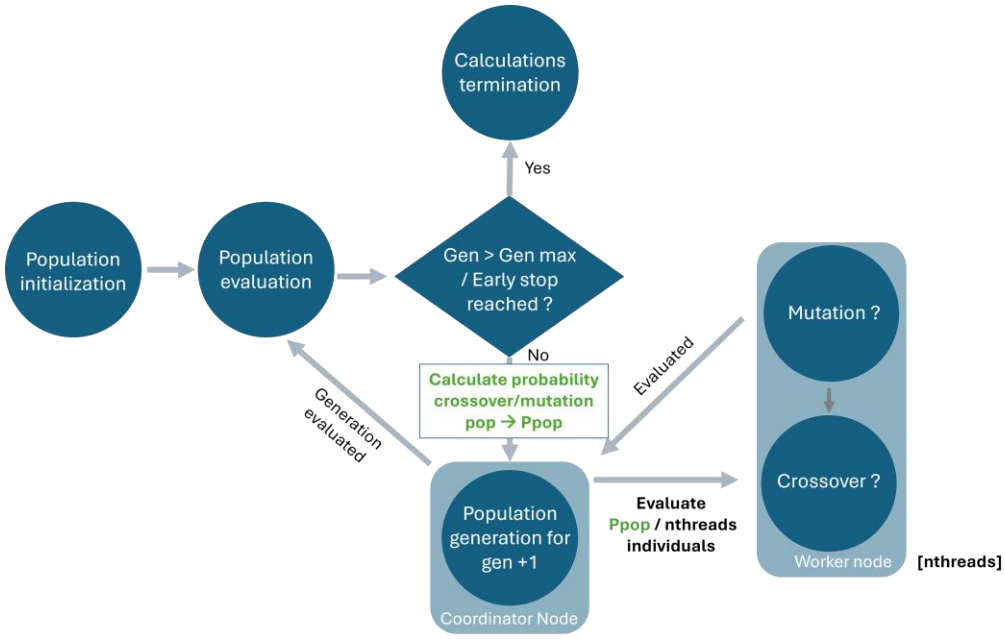

Figure 4-8. Architecture of the Parallel GA v2. The coordinator process distributes to its workers only those chromosomes that will be evolved by genetic operators.

Only these identified chromosomes were distributed across the threads for evaluation. This method aimed to ensure that threads were actively engaged in processing relevant computations. By focusing thread activities on chromosomes requiring genetic operations, we aimed to use more efficiently the workers and mitigate the delays caused by varying completion times among threads. A summary of the whole procedure for the Parallel GA v2 may be found in the form of pseudo-code in Figure 4-9.

```
LaunchThread {
for each individual C_i of Ppop/NbThreads
  if crossover
    Select individual C_j by tournament selection
    Crossover individuals C_i and C_j, creating individuals C_ij and C_ji
    Select fittest offspring
    Survivor selection between C_i and fittest offspring, winner replace C_i
  if mutation
    Mutation point selected at random
    Gene is randomized, creating individual C_j
    Survivor selection between C_i and C_j, winner replace C_i
}

Initialize Population

while termination criteria not met do
    Compute probability crossover/mutation: Ppop
    for i=0 to NbThreads -1 do
      LaunchThread(Ppop/NbThreads)
    end
    BackupPopIfNeeded
end
CreateModelsComputeStat
```

Figure 4-9. Pseudo code of the Parallel GA v2.

To accelerate the process even further, we developed a steady-state GA. Unlike the generational GA, the steady-state GA operates in a continuous fashion (Parallel GA v3, Figure 4-10).

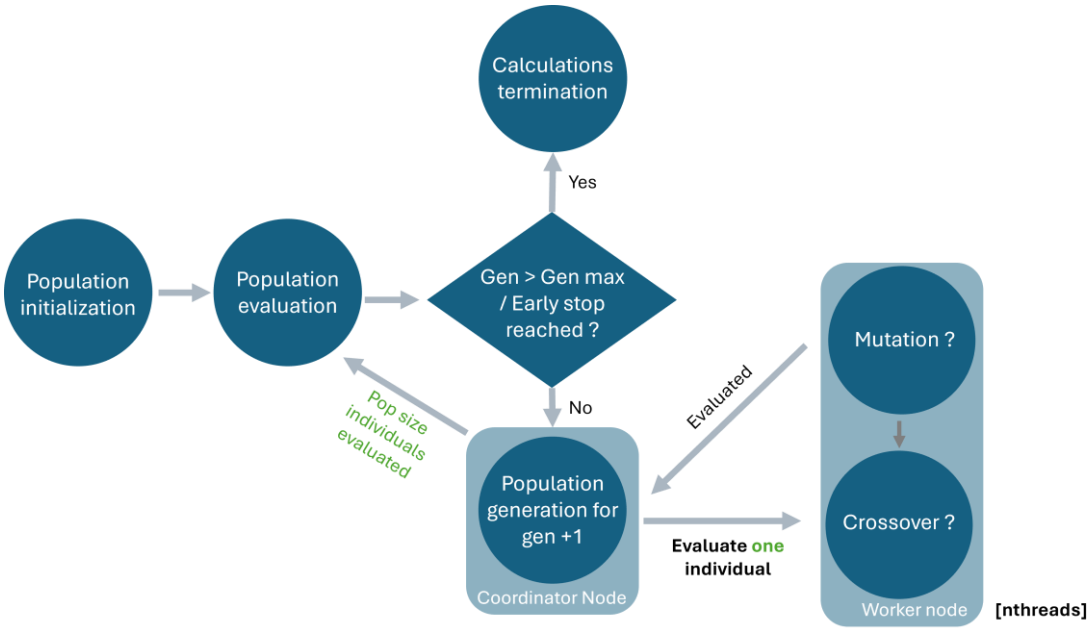

Figure 4-10. Architecture of the Parallel GA v3. As soon as a worker finished its job and returned its fitness scores, the coordinator fires a new worker job with one of the chromosomes it identified in the meantime.

It handles solutions iteratively, generating and replacing them one by one. Once populationSize chromosomes are evaluated, we define it as a pseudo generation. This scheme loses the notion of well-defined generations, but maximizes the occupancy of the workers. A summary of the whole procedure for the Parallel GA v3 may be found in the form of pseudo-code in Figure 4-11.

```
LaunchThread {
if crossover
    Select individual Cj by tournament selection
    Crossover individuals Ci and Cj, creating individuals Cij and Cji
    Select fittest offspring
    Survivor selection between Ci and fittest offspring, winner replace Ci
if mutation
    Mutation point selected at random
    Gene is randomized, creating individual Cj
    Survivor selection between Ci and Cj, winner replace Ci
Inc(Updated)
}

Initialize Population

while termination criteria not met do
for each individual Ci of the population
    if Ci not in calculation
      FindAvailableThread
      LaunchThread
    end
    if Updated >= populationSize
       Inc(gen)
       BackupPopIfNeeded
    end
end
CreateModelsComputeStat
```

Figure 4-11. Pseudo code of the Parallel GA v3.

Additionally, we developed a version of the Parallel GA V3 where duplicate chromosomes are removed before being added to the population and replaced by random offspring (Parallel GA v3 NoDuplicate). This enhancement ensures a more diverse gene pool, preventing the algorithm from stagnating and improving its ability to explore the solution space effectively. By eliminating duplicates, we aim to enhance the overall performance of the GA, leading to better optimization results and faster convergence. A summary of the whole procedure for the Parallel GA v3 NoDuplicate may be found in the form of pseudo-code in Figure 4-12.

```
LaunchThread {
if crossover
    Select individual C_j by tournament selection
    Crossover individuals C_i and C_j, creating individuals C_ij and C_ji
    Select fittest offspring
    CheckDuplicate
    Survivor selection between C_i and fittest offspring, winner replace C_i
if mutation
    Mutation point selected at random
    Gene is randomized, creating individual C_j
    CheckDuplicate
    Survivor selection between C_i and C_j, winner replace C_i
Inc(Updated)
}

Initialize Population

while termination criteria not met do
for each individual C_i of the population
    if C_i not in calculation
      FindAvailableThread
      LaunchThread
    end
    if Updated >= populationSize
       Inc(gen)
       BackupPopIfNeeded
    end
end
CreateModelsComputeStat
```

Figure 4-12. Pseudo code of the Parallel GA v3 NoDuplicate.

Finally, we adapted the procedure introduced by Srinivas et al. [87], where they worked on adaptive probabilities of crossover and mutation (Parallel GA v3 Adaptive). In this paper, they introduce an approach where the probabilities of crossover and mutation are not predefined, but function of the FS of the individual. High-fitness solutions are assigned low probabilities to improve exploitation, while low-fitness solutions are assigned high probabilities to improve exploration, the aim is to prevent the GA from getting stuck in local optima. In this implementation, similar to Parallel GA v3 NoDuplicate, we ensure that a chromosome is not already present in the population before adding it. A summary of the whole procedure for the Parallel GA v3 Adaptive may be found in the form of pseudo-code in Figure 4-13.

```
LaunchThread {
if AdaptiveCrossover
    Select individual Cj by tournament selection
    Crossover individuals Ci and Cj, creating individuals Cij and Cji
    Select fittest offspring
    CheckDuplicate
    Survivor selection between Ci and fittest offspring, winner replace Ci
if AdaptiveMutation
    Mutation point selected at random
    Gene is randomized, creating individual Cj
    CheckDuplicate
    Survivor selection between Ci and Cj, winner replace Ci
Inc(Updated)
}

Initialize Population

while termination criteria not met do
for each individual Ci of the population
    if Ci not in calculation
      FindAvailableThread
      LaunchThread
    end
    if Updated >= populationSize
       Inc(gen)
       BackupPopIfNeeded
    end
end
CreateModelsComputeStat
```

Figure 4-13. Pseudo code of the Parallel GA v3 Adaptive.

To summarize, we have covered several aspects: we discussed the fitness functions employed in both pGA-GTM and pGA-SVM, explained how we selectively distributed chromosomes for crossover and mutation operations to minimize waiting times, and detailed the specific genetic operations used for these processes. Furthermore, we provided insights into the architectural differences between the few generational GA and the steady-state GA, emphasizing the latter's continuous integration of new solutions to achieve faster population convergence. We will now see of the three versions of the parallel GA v3 were applied to a test case and compared to the state of the art.

4.2.2 Comparison with libsvm-GA

We compared the 3 versions of the pGA-SVM parallel v3 with libsvm-GA [77] on 20 ChEMBL [119–122] targets under regression and classification tasks for the prediction of activity (pChEMBL, [123]). The aim is to validate this new GA engine. ChEMBL is an open database that provides curated bioactivity data. The latest version of ChEMBL (ChEMBL 34, [132]) contains over 2.4 million compounds, 20 million activities, and more than 15,000 targets. pChEMBL allows for a comparison of comparable measures of half-maximal responses (molar IC50, XC50, EC50, AC50, Ki, Kd, Potency, ED50), on a negative logarithmic scale.

For this comparison, we use a workflow for automatic generation of QSAR models for biological targets from the ChEMBL database to which contributed Erik Yeghyan, a M2 student in 2024. A reference implementation, libsvm-GA was used in allowing for direct comparison to the pGA-SVM new implementation. Unlike the method employing libsvm-GA, which generated 703 regression and 745 classification models, we chose to evaluate only the top 10 unambiguous models in each category (models scoring too high in Yeghyan's work were flagged as ambiguous, as they need to be inspected). This selection was based on their fitness score, ensuring that we concentrated on the most effective models. All the models were built on the same computer using (for the parallelized version) the same number of threads. We used the standard parameters for the libsvm-GA (around 3000 chromosomes evaluated). For the 3 versions of the pGA-SVM parallel v3, we used a population of 150 and a stopping criteria of 150 generations without improvement.

The results of the comparison can be found in Table 4-1, and a graphic view of the results of each architecture may be found on Figure 4-14.

Target	Libsvm-GA	Parallel GA v3	parallel GA v3 Adaptive	parallel GA v3 NoDuplicate
CHEMBL1250348_REGstr	0,811	0,817	0,822	0,82
CHEMBL1801_REGstr	0,822	0,828	0,828	0,84
CHEMBL1804_REGstr	0,828	0,844	0,858	0,855
CHEMBL1827_CLS	0,82	0,825	0,829	0,826
CHEMBL1871_CLS	0,8	0,806	0,819	0,807
CHEMBL1889_CLS	0,809	0,894	0,896	0,894
CHEMBL1908_CLS	0,899	0,947	0,947	0,946
CHEMBL202_CLS	0,831	0,894	0,912	0,912
CHEMBL2035_CLS	0,812	0,873	0,878	0,88
CHEMBL216_CLS	0,819	0,826	0,834	0,826
CHEMBL2459_CLS	0,801	0,985	0,985	0,986
CHEMBL2581_REGstr	0,844	0,863	0,868	0,868
CHEMBL2695_REGstr	0,83	0,83	0,832	0,834
CHEMBL275_REGstr	0,831	0,834	0,838	0,838
CHEMBL3572_CLS	0,804	0,857	0,865	0,86
CHEMBL3798_REGstr	0,812	0,811	0,819	0,817
CHEMBL4508_REGstr	0,859	0,859	0,868	0,869
CHEMBL4860_REGstr	0,858	0,86	0,865	0,862
CHEMBL5658_REGstr	0,833	0,836	0,839	0,835
CHEMBL5921_CLS	0,829	0,877	0,88	0,878

Table 4-1. Results of the comparison between libsvm-GA and the 3 kinds of parallel GAv3. The v3 refers to the version presented in Figure 4-10.

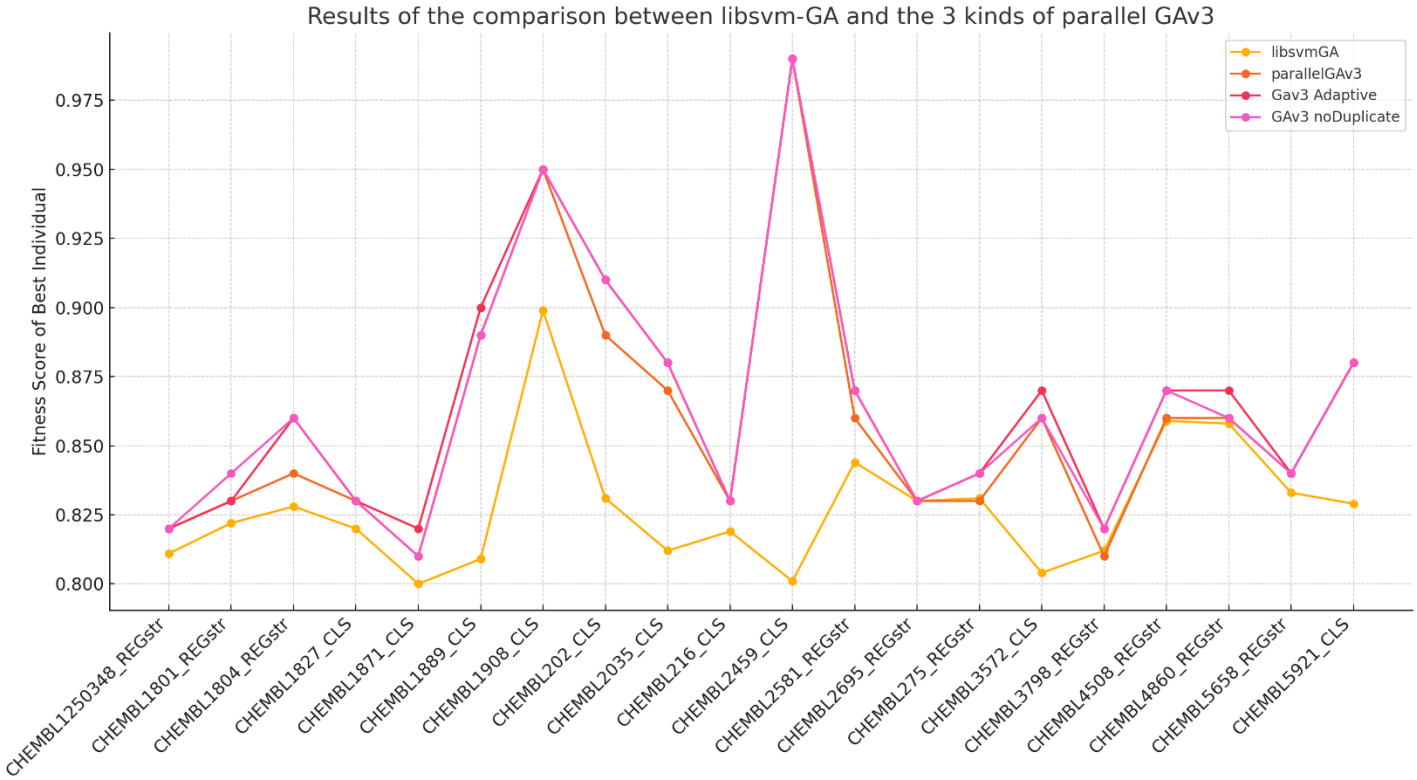

Figure 4-14. Graphical view of the comparison between the GA implementations – data from Table 4-1.

We compared the new GA engines to the reference implementation (libsvm-GA) using a student paired test (paired t-test), a statistical method used to compare the means of two related groups to determine if there is a significant difference between them [133]. For these tests, we checked for the null hypothesis (no difference between the two tested methods) with 19 degrees of freedom. The results of the 3 paired t-test may be found in Table 4-2.

Comparison	t-value	p-value
libsvm-GA/parallelGAv3	3,0635	0.0032
libsvm-GA/parallelGAv3 Adaptive	3,7045	0.00075
libsvm-GA/parallelGAv3 NoDuplicate	3,5112	0.00115

Table 4-2. Results of the paired t-test between libsvm-GA (reference implementation) and the 3 versions of parallelGA v3.

From these t-values, with 19 degrees of freedom, we can say that for both parallel GAv3, parallel GAv3 Adaptive and parallelGAv3 NoDuplicate, the null hypothesis is rejected at the standard 1% risk level. The new GAs engines performances are improved compared to the reference implementation.

These results can be explained as we leave more time for the pGA-SVM to explore the solution space. Indeed, for the 10 classification tasks, libsvm-GA evaluates an average of 2142 chromosomes, while pGA-SVM evaluates an average of 15291 chromosomes. We are able to do so while keeping the computation time lower, thanks to the usage of compiled code and the integration of libsvm code in Pascal object, allowing for more exploration of the solution space while not improving the time spent by the user looking for better solutions. Indeed, we can compute an average of 9.9 chromosomes/s for a SVC, while libsvm-GA can compute an average of 1.6 chromosomes/s. For the SVR, we compute an average of 1.25 chromosomes/s while libsvm-GA can compute an average of 0.8. These are indicative values on a single multi-core CPU architecture – the new engines do not cover distributed calculations on a cluster.

To summarize, we have statistically demonstrated using the paired t-test that the three implementations of the pGA-SVM parallel v3 are more effective than libsvm-GA on this problem under this configuration, indeed we cannot say for sure that it will be more effective in every situation, according to the *no free lunch theorem* [134]. Furthermore, we can observe qualitatively that both the Adaptive and NoDuplicate versions of the parallelGAv3 are better than the parallelGAv3.

4.2.3 Analysis on the modification of GA inner parameters

We conducted a detailed analysis of the modification of the GA inner parameters. It involves systematically tuning each parameter one at a time while keeping the other parameters set to their default values. By default, the stopping criteria is 50, the population size is 100, the crossover rate is 0,3 and the mutation rate is 0,1. By isolating the effects of individual parameters in this manner, we aim to understand their specific contributions to the overall performance. This approach allows us to identify the impact of each parameter on the performance and computation time usage of the algorithm. We repeated the procedure for the different type of GA implemented.

For this study, we varied the crossover rate, the mutation rate, the population size, and the stopping criteria. The exploration of these parameters has been conducted on the target CHEMBL1908 for a classification task, as the parallel GAv3 implementations performed well compared to libsvm-GA for this task and because the data set is reasonably sized. Increasing the crossover and mutation rate will increase the number of chromosomes to evaluate, hence increasing calculation time.

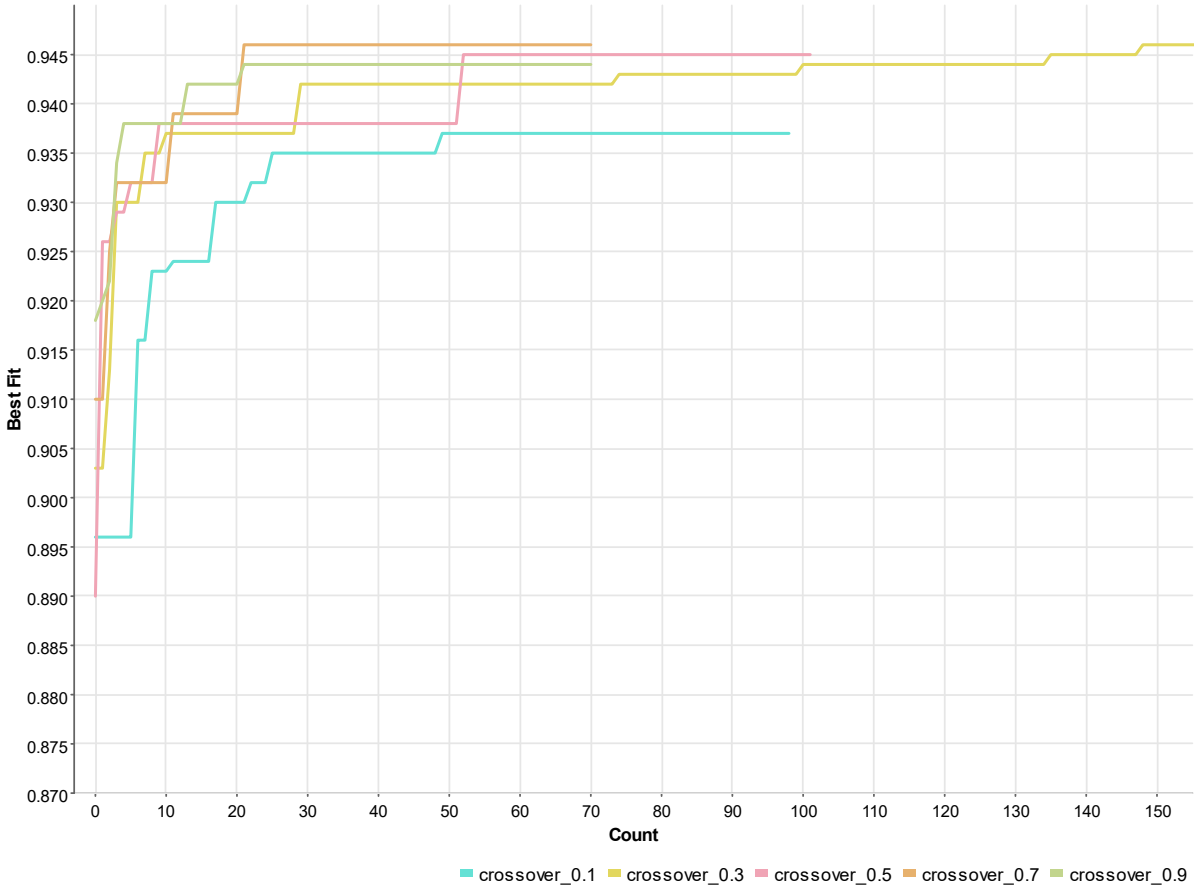

Figure 4-15. Fitness score of the best individual in function of the generation and the crossover rate.

In Figure 4-15, we observe the evolution of the fitness score of the best chromosome in the population across generations for the five tested crossover rates. To facilitate selecting the best parameters and considering that a higher crossover rate yields to increased computation time, in Table 4-3 are indicated the crossover rate and corresponding computation time, fitness score of the last best chromosome.

Crossover rate	Fitness Score	Computation time (minute)
x0.1	0.937	2
x0.3	0.946	6
x0.5	0.945	5
x0.7	0.946	5
x0.9	0.944	6

Table 4-3. Results of the study on the modification of the crossover rate. Fitness score of the best chromosome of the last generation and computation time in minutes.

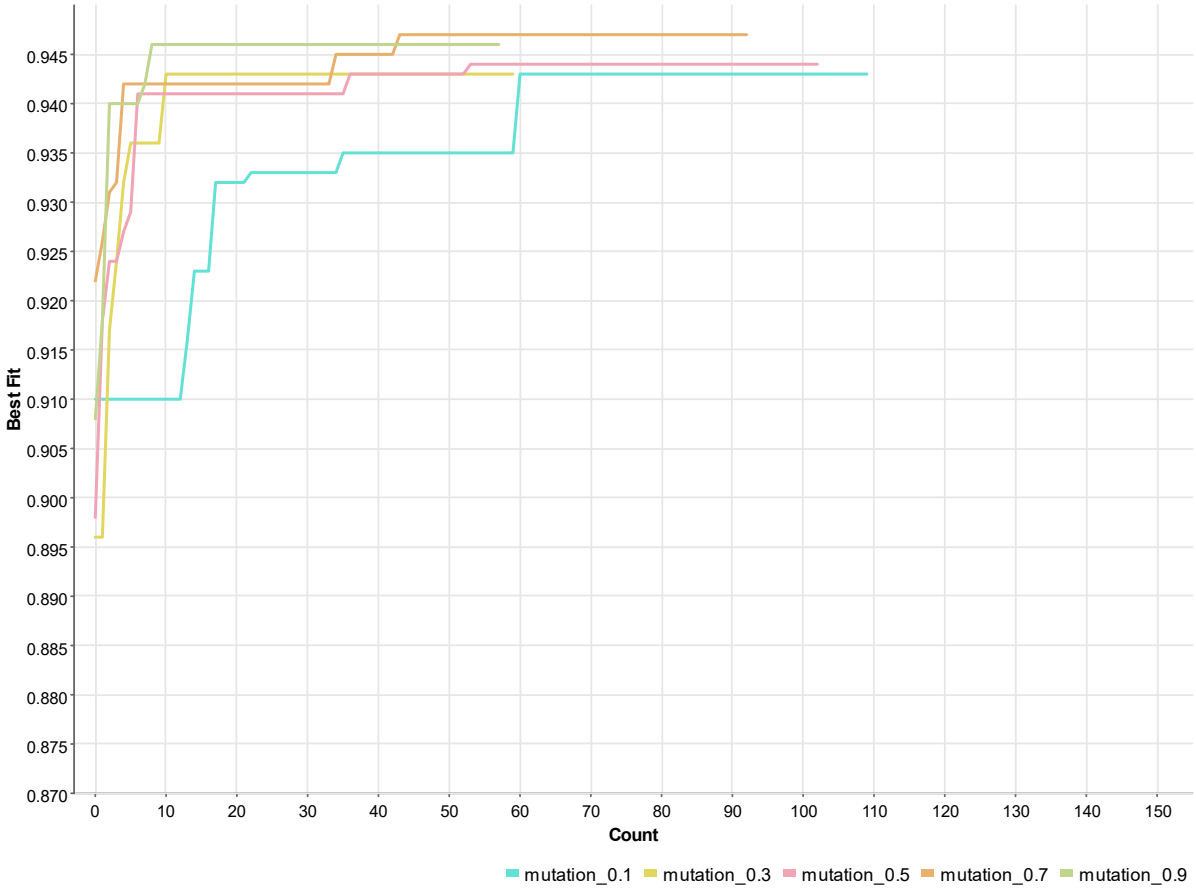

Figure 4-16. Fitness score of the best individual in function of the generation and the mutation rate.

In Figure 4-16, we observe the evolution of the fitness score of the best chromosome in the population across generations for the five tested mutation rates. Considering that a higher crossover rate yields to increased computation time, in Table 4-4 are indicated the mutation rate, and corresponding computation time and fitness score of the last best chromosome.

Mutation rate	Fitness Score	Computation time (minute)
m0.1	0.943	2
m0.3	0.943	2
m0.5	0.944	6
m0.7	0.947	6
m0.9	0.946	6

Table 4-4. Results of the study on the modification of the mutation rate. Fitness score of the best chromosome of the last generation and computation time in minutes.

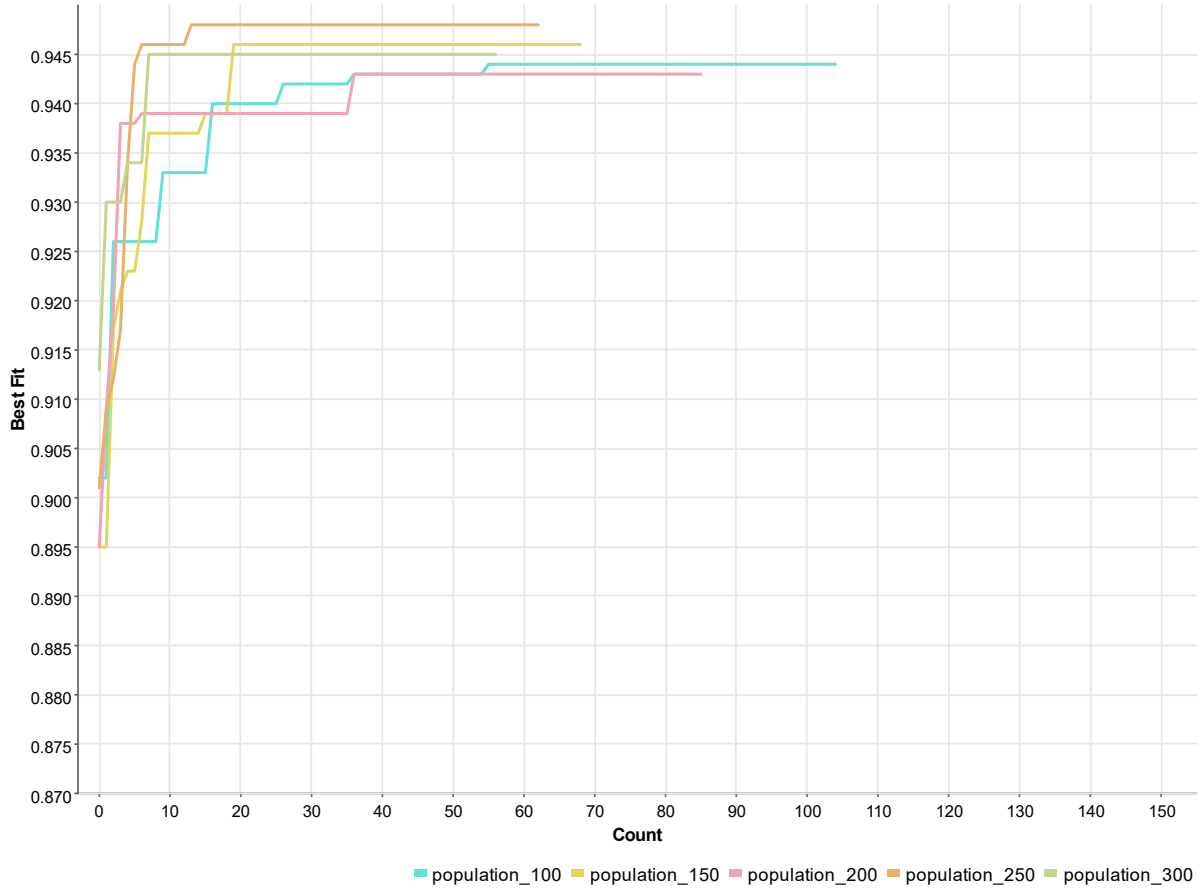

Figure 4-17. Fitness score of the best individual in function of the generation and the population size.

In Figure 4-17, we observe the evolution of the fitness score of the best chromosome in the population across generations for the five tested population sizes. In Table 4-5 are indicated the population size, corresponding computation time and fitness score of the last best chromosome.

Population size	Fitness Score	Computation time (minute)
p100	0.944	4
p150	0.946	4
p200	0.943	8
p250	0.948	7
p300	0.945	6

Table 4-5. Results of the study on the modification of the population size. Fitness score of the best chromosome of the last generation and computation time in minutes.

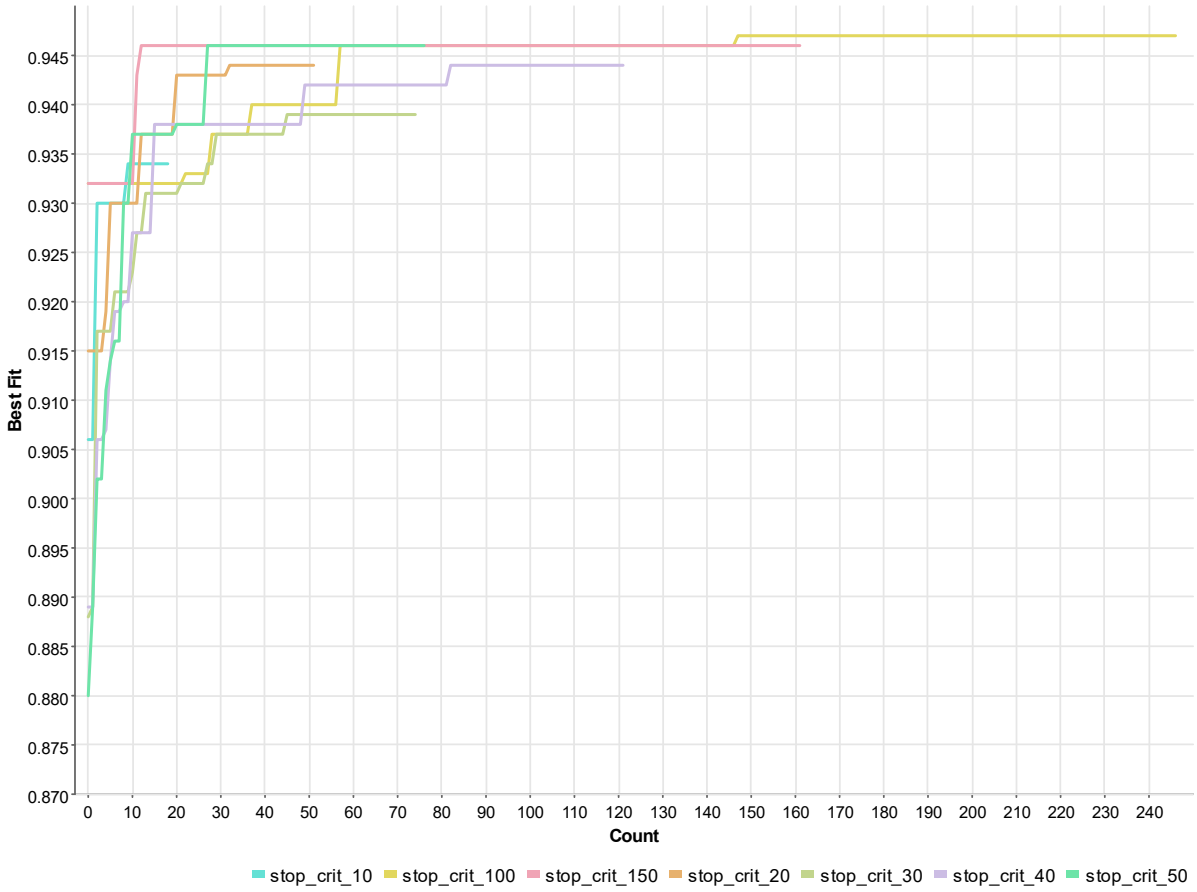

Figure 4-18. Fitness score of the best individual in function of the generation and the stopping criteria.

In Figure 4-18, we observe the evolution of the fitness score of the best chromosome in the population across generations for the seven tested stopping criteria. In Table 4-6 are indicated the stopping criteria value, corresponding computation time and fitness score of the last best chromosome.

Stopping criteria	Fitness Score	Computation time (minute)
Stop10	0.934	4
Stop20	0.944	4
Stop30	0.939	8
Stop40	0.944	7
Stop50	0.946	6
Stop100	0.947	9
Stop150	0.946	6

Table 4-6 Results of the study on the modification of the stopping criteria value. Fitness score of the best chromosome of the last generation and computation time in minutes.

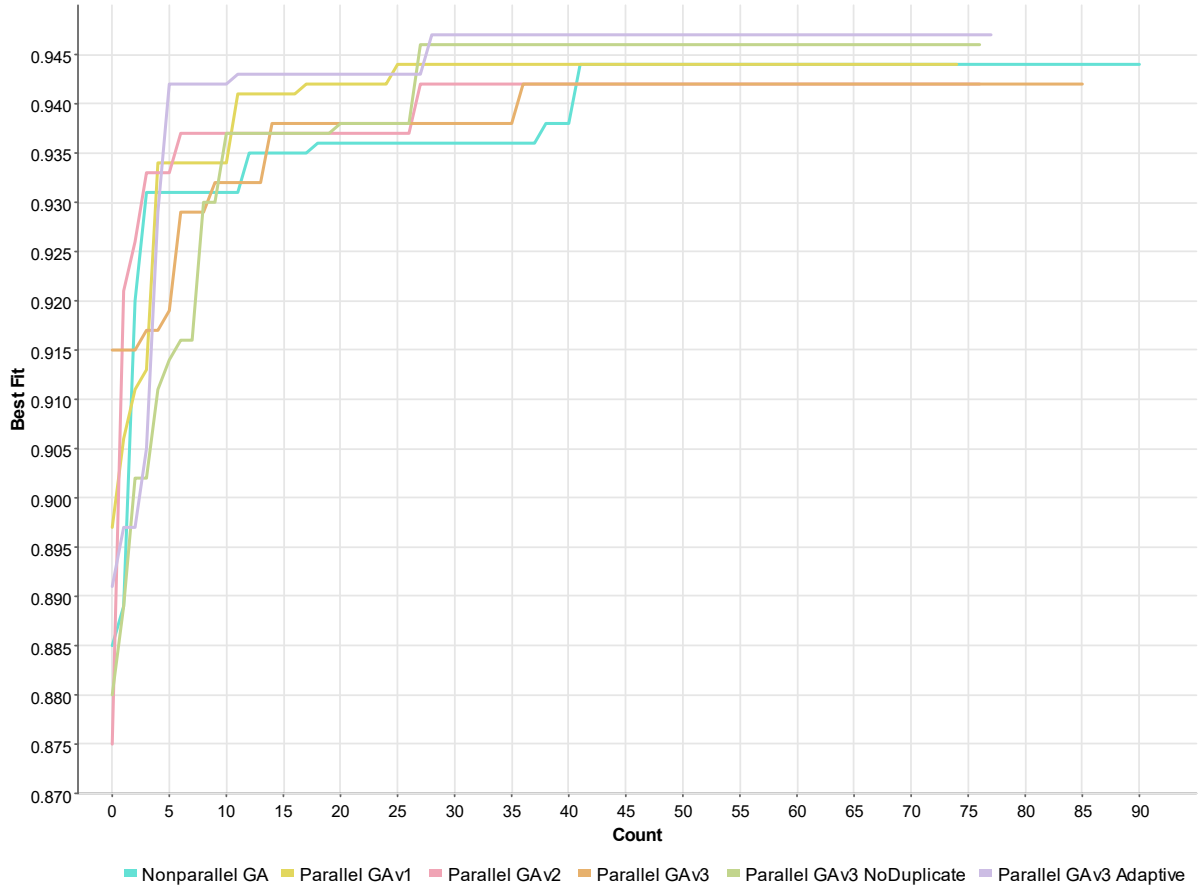

Figure 4-19. Fitness score of the best individual in function of the generation and different type of GA implemented.

In Figure 4-19, we observe the evolution of the fitness score of the best chromosome in the population across generations for the different type of GA implemented. In Table 4-7 are indicated the implementation of GA, corresponding computation time and fitness score of the last best chromosome.

Stopping criteria	Fitness Score	Computation time (minute)
SimpleGA	0.944	87
parallelGAv1	0.944	16
parallelGAv2	0.942	17
parallelGAv3	0.942	7
parallelGAv3Adaptive	0.947	8
parallelGAv3NoDuplicate	0.946	3

Table 4-7. Results of the study on the modification of the type of GA. Fitness score of the best chromosome of the last generation and computation time in minutes.

To summarize, we explored the influence of GA inner parameters on the fitness score and the computational time, on the CHEMBL1908 task. While it is impossible to generalize, we provide two sets of parameters, one that seems to lead to good models in shorter time and one for best model generation, meaning more exploration of the solution space:

Good models in shorter time:

- `parallelGAv3 NoDuplicate, p150, x0.5, m0.1, stop20`

Best model generation:

- `parallelGAv3 Adaptive, p250, stop150`
- `parallelGAv3 NoDuplicate, p250, x0.3, m0.7, stop150`

4.3 Benchmarking of the Genetic algorithm

The pGA-SVM and pGA-GTM tools were also tested on a few other test cases, on the work of Dr. Mikhail Volkov. Thanks to this, we had three new environments to benchmark our tools and make sure of their proper functioning. Therefore, we present three QSAR modeling tasks: QSAR modeling of the antioxidant activity, QSAR modeling of antibacterial activity against Staphylococcus aureus and against Staphylococcus Epidermidis.

In these three cases, similar methods were used and compared to the pGA-SVM. For this reason and to avoid repeating ourselves, we will present them in this introduction.

To describe the data sets, two kinds of descriptors were used:

- ISIDA fragment descriptors were computed for 100 descriptor spaces.
- Morgan fingerprints [135] of radius 4 were computed using RDKit [136] implementation

Several machine learning methods were used for this comparison:

- The Random Forest (RF) models used in this study were implemented using the scikit-learn library [137] in Python, with the default parameters (100 decision trees).
- The scikit-learn implementation of the SVM using default parameters (RBF kernel, scaled gamma, coeff0 of 0.0, C of 1, epsilon of 0.1).
- pGA-SVM parallelGA v3 NoDuplicate was employed with a population size of 200 and a stopping criteria of 50.

The detail of these QSAR modeling task is presented below.

4.3.1 QSAR modeling of the antioxidant activity

Antioxidants (AOX) are chemicals used due to their ability to counteract the effect of reactive oxygen species (ROS) [138]. In biological systems, ROS formation leads to oxidative stress, causing damage to biomolecules, and contributing to diseases and premature aging [139, 140]. AOX compounds counteract these effects by neutralizing free radicals and inhibiting oxidation processes.

The Trolox Equivalent Antioxidant Capacity (TEAC) assay [141] is an accepted method to measure the antioxidant potency. It involves measuring the decrease in the concentration of the free radical monocation 2,2′-azinobis-(3-ethylbenzothiazoline-6-sulfonic acid) by colorimetry in presence of a test compound, normalizing with the activity of Trolox determined under the same conditions (TEAC value is defined as the concentration of standard Trolox with the same antioxidant capacity as a 1mM concentration of the antioxidant investigated sample).

QSAR modeling was performed with a dataset of heterogeneous phenolic compounds from Idrovo-Encalada et al. (IE) [142] extended by including entries from the AODB database [143].

This merged dataset was randomly split between train, test and external validation sets (Table 4-8).

Total	Training set	Test set	External validation set
533	350	92	91

Table 4-8. Distribution of the data between the training, test and external validation sets.

Random forest models were built for each ISIDA DS, and the models best performing in 5-fold cross-validation were selected. An ensemble model was constructed.

Results in Table 4-9 indicates that the pGA-SVM performs comparably to the other tested methods (metrics on external validation set).

Method used	R^2	RMSE
Random forest **ISIDA descriptors**	0.619	0.365
Random forest (ensemble modeling of 94 models) **ISIDA descriptors**	0.685	0.333
pGA-SVM **ISIDA descriptors**	0.659	0.346
pGA-SVM (ensemble of 3 models) **ISIDA descriptors**	0.669	0.341

Table 4-9. Results of the benchmark study on the modeling of the antioxidant activity.

pGA-SVM was used to create models for predicting AOX activity. The results were comparable to those obtained from other tested methods, confirming the effectiveness of pGA-SVM.

4.3.2 QSAR modeling of antibacterial activity against Staphylococcus aureus and Staphylococcus Epidermidis

Antimicrobial resistance (AMR) is a major global public health concern [144]. This study focused on the QSAR modeling of the antibacterial activity against two strains of Staphylococcus: aureus (S. aureus) ATCC 6538, and Epidermidis (S. epidermidis) ATCC 14990.

The Minimal Inhibitory Concentration (MIC) [145] is a measure indicating the lowest concentration of an antimicrobial agent that inhibits *in vitro* growth of the bacterium. In this study, activity is expressed as log(MIC), with MIC in nM units.

Random forest models were built using Morgan fingerprints. Scikit-learn implementation of the SVM was used to train SVM model using Morgan fingerprints.

QSAR modeling of S. aureus was performed from a curated dataset of 1628 compounds, retrieved from ChEMBL database, after selection of relevant assays, activity curation, standardization and duplicates removal. This dataset was clustered by Murcko scaffold [146] (using RDKit implementation), and test set constructed by randomly selecting approximately 20% of each scaffold clusters, the remaining being used for the training set (Table 4-10).

Total	Training set	Test set
1618	1264	354

Table 4-10. Distribution of the data between the training and test sets for the modeling of S. aureus.

Results in Table 4-11 indicates that the pGA-SVM performs comparably to the other tested methods. Statistics on the test set.

Method used	R^2	RMSE
Random forest **Morgan fingerprints (radius 4)**	0.749	1.588
SVM **Morgan fingerprints (radius 4)**	0.631	1.927
pGA-SVM **ISIDA descriptors**	0.757	1.507

Table 4-11. Results of the benchmark study on the modeling of MIC against S. aureus ATCC 6538.

QSAR modeling of S. epidermidis was performed from a curated dataset of 189 compounds, retrieved from ChEMBL database, after selection of relevant assays, activity curation, standardization and duplicates removal. This dataset was clustered by Murcko framework, and test set constructed by randomly selecting approximately 20% of each scaffold clusters, the remaining being used for the training set (Table 4-12).

Total	Training set	Test set
189	145	44

Table 4-12. Distribution of the data between the training and test sets for the modeling of S. epidermidis.

Results in Table 4-13 indicates that the pGA-SVM performs comparably to the other tested methods. Statistics observed on the test set.

Method used	R^2	RMSE
Random forest **Morgan fingerprints (radius 4)**	0.636	1.721
SVM **Morgan fingerprints (radius 4)**	0.680	1.613
pGA-SVM **ISIDA descriptors**	0.870	1.199

Table 4-13. Results of the benchmark study on the modeling of MIC against S. epidermidis ATCC 14990.

pGA-SVM was used to create models for predicting antibacterial activity against two Staphylococcus. The results were comparable or even better to those obtained from other tested methods, confirming the effectiveness of pGA-SVM.

5 Conclusions and perspectives

In this thesis work, we have developed and published several innovative open-source tools and plugins aimed at enhancing educational and research experiences within the field of chemistry and cheminformatics. The primary focus has been to create tools that facilitate the insertion and grading of chemical drawings and reactions in Moodle, a widely used learning management system.

With the ChemMoodle project, we presented three new Moodle plugins built to ease the learning and teaching of chemistry within Moodle:

The MolStructure plugin allows users to insert chemical structures and reactions into any text area. This plugin supports 2D and 3D representations of molecules and the display of spectral information such as mass, NMR, and spectroscopy. It has been updated to a TinyMCE plugin to follow the modifications of Moodle software development policy. Since its release in June 2022, the MolStructure Atto plugin has been downloaded 394 times and is actively used on 79 sites, with its adoption steadily increasing each month. The MolStructure TinyMCE plugin, released in March 2024, has also shown promising uptake, with 88 downloads and usage on 29 sites.

The MolSimilarity plugin is a question-type plugin that allows teachers to ask chemistry-related questions. The questions are auto corrected by a smooth grading algorithm that has been created during this thesis. The correction algorithm accounts for the complexity introduced by questions about chemical structures.

The ReacSimilarity plugin is a question-type plugin that allows teachers to ask chemical reaction questions. The workflow was modified following the work of MolSimilarity using CGR to work with chemical reactions.

These two plugins employ a soft grading approach, implemented through a REST API server specifically developed for this purpose. ISIDA software was also extended by developing new units that facilitate the construction of CGRs.

The plugins developed for ChemMoodle are integrated into a software maintenance program supported by the University of Strasbourg and the Laboratoire de Chémoinformatique (UMR 7140), ensuring ongoing updates and sustainability.

The development of a new process flow diagram open-source sketcher, “ChemEngineering” has begun. It is a significant step forward in chemical engineering

as it is the first open-source process flow diagram sketcher. It will be included in a TinyMCE plugin, following the MolStructure TinyMCE implementation. By offering a free and accessible tool, we aim to enhance the educational experience. By releasing the project on an open platform, we hope to engage and frame a community by providing a clear contribution policy. To ensure the security of the source code in this open environment, we will establish a code review process and implement rigorous unit testing procedures.

Moreover, we laid the basis for the development of an innovative tool for drafting chemical questions based on conceptual frameworks, allowing educators to generate multiple concrete questions from a single concept. This approach aims at reducing redundancy for the teacher, enabling educators to focus on pedagogical aspects rather than the mechanics of question generation. We focused on the genetic algorithm part of this development, by developing a version for SVM and one for GTM. We detailed the development of various architectures and genetic operations, comparing the performance of pGA-SVM to libsvm-GA on ChEMBL targets for regression and classification tasks. Our findings indicated that certain implementations of the parallel GA, particularly the Adaptive and NoDuplicate versions, performed better than others. We explored the influence of GA inner parameters on fitness scores and computational time, providing parameter sets for different optimization goals. Then, we applied the developed GAs to three modeling tasks using: the modeling of the antioxidant activity and the modeling of antibacterial activity against staphylococcus aureus and staphylococcus epidermidis. It showed good results, comparable with other machine learning methods such as random forest.

This work could be extended by adding new tools to the ChemMoodle project, and by continuing the development of the other projects. The ChemMoodle project could be enhanced with new question-type plugins, where the grading procedures could consider the teacher’s defined sub-graphs of chemical structures.

Furthermore, the ChemEngineering sketcher could be included in a question-type plugin, which will require further developments on the server side, to account for this new kind of graphs to grade. Then, the technical part of the question generator will need to be implemented, to include the pGA-GTM in it.

The ChemMoodle project also needs to be developed in other ways. First, one objective will be to convince the academies, at the French national level, to experiment with it in the context of secondary education. We believe that the availability of Moodle

and the self-assessment possibilities of ChemMoodle can contribute to chemical education early in students' academic journey. For instance, Slovenia has started a process to include ChemMoodle in the tool list for secondary education.

Second, a pedagogical analysis of the ChemMoodle tools is needed, meaning that the impact of these tools must be quantified and proven. A pedagogical research program is required for this reason. For instance, we can target specific lectures that could use activities based on ChemMoodle and compare the results of two cohorts of students: one group using the tool and the other not using it. In the Faculty of Chemistry of the University of Strasbourg, we could target the lab sessions, where students are invited to answer questions regarding the practical lectures before starting the experiments. The questions are formulated today without the help of ChemMoodle. The "Synthèse de Connaissances" consists of a long and difficult exam session in semester 2 and semester 3 of the BSc of Chemistry, that challenges all the knowledge acquired by the students during the year. The preparation for these exams currently does not use ChemMoodle. We can offer to translate previous exams into ChemMoodle format, for student training purposes. Of course, these perspectives will need the help of pedagogy engineers and researchers, and we could think of a collaboration with the Institut de Développement et d'Innovation Pédagogique (IDIP) of the University of Strasbourg.

The pGA-GTM project, as discussed in this thesis, also has promising applications beyond the ChemMoodle project. It offers users an open-source project that implements portable genetic algorithm code. While more computationally intensive projects will still require distribution on computer grids, pGA-GTM aims to solve other problems in a more cost-effective manner. The project could use two key improvements. First, multiple ways for the user to bind custom scoring functions: adding source code in Free Pascal, calling shared libraries or making system calls to compiled applications. This would require designing a specialized API and creating the corresponding technical documentation. Second, it could feature a graphical user interface to offer a richer user experience and be extended into a web application. Additionally, pGA-GTM could be delivered as an API compatible with popular scripting environments like R, KNIME, and Python, and as a set of web resources, to facilitate its integration into web-distributed data processing pipelines.

In conclusion, this thesis has laid the groundwork for several impactful tools that enhance chemical education and cheminformatic research.

6 List of abbreviations

AAM Atom to Atom Mapping
ADMET absorption, distribution, metabolism, excretion, and toxicity
BA Balanced Accuracy
CGR Condensed Graph of Reaction
CHFP Chemical Hashed Fingerprints
CS Chemical Space
DS Descriptor Space
EA Evolutionary Algorithms
FN False Negative
FP False Positive
FS Fitness Score
GA Genetic Algorithm
GTM Generative Topographic Mapping
ISIDA In SIlico design and Data Analysis
LLh LogLikelihood
LMS Learning Management System
PCA Principal Component Analysis
PFD Process Flow Diagrams
QSAR Quantitative Structure-Activity Relationship
QSPR Quantitative Structure-Property Relationship
R Coefficient of correlation
R^2 Coefficient of determination
RBFs Radial Basis Functions
RDF Reaction Data File
RMSD Root Mean Square Deviation
RMSE Root Mean Square Error
ROS Reactive Oxygen Species
SMF Substructure Molecular Fragments
SMILES Simplified Molecular Input Line Entry Specification
SOCOT Structure-Based Organic Chemistry Online Tutorials
SOM Self-Organizing Maps
SSR Sum of Squared Residuals
SST Total Sum of Squares
SVM Support Vector Machines
SVR Support Vector Regression
TD Travaux Dirigés
TEAC Trolox Equivalent Antioxidant Capacity
TinyMCE Tiny Moxiecode Content Editor
TN True Negative
TNR True Negative Rate
TP Travaux Pratiques
TP True Positive
TPR True Positive Rate
t-SNE t-Distributed Stochastic Neighbor Embedding
WYSIWYG What You See Is What You Get
XV cross-validation
YUI Yahoo ! User Interface

7 References

1. Baturay MH (2015) An Overview of the World of MOOCs. Procedia - Social and Behavioral Sciences 174:427–433. https://doi.org/10.1016/j.sbspro.2015.01.685

2. Moodle home page. https://moodle.org. Accessed 9 Dec 2021

3. ChemAxon. Marvin JS. https://chemaxon.com/marvin#MarvinJS. Accessed 4 May 2024

4. ChemAxon main page. https://chemaxon.com/. Accessed 4 May 2024

5. Vyas VS, Reid SA (2023) What Moves the Needle on DFW Rates and Student Success in General Chemistry? A Quarter-Century Perspective. J Chem Educ 100:1547–1556. https://doi.org/10.1021/acs.jchemed.2c01121

6. Burger MC (2015) ChemDoodle Web Components: HTML5 toolkit for chemical graphics, interfaces, and informatics. J Cheminform 7:35. https://doi.org/10.1186/s13321-015-0085-3

7. Ruggiu F, Marcou G, Varnek A, Horvath D (2010) ISIDA Property-Labelled Fragment Descriptors. Mol Inf 29:855–868. https://doi.org/10.1002/minf.201000099

8. Fujita S (1986) Description of organic reactions based on imaginary transition structures. 1. Introduction of new concepts. J Chem Inf Comput Sci 26:205–212. https://doi.org/10.1021/ci00052a009

9. Fujita S (1987) Description of organic reactions based on imaginary transition structures. 6. Classification and enumeration of two-string reactions with one common node. J Chem Inf Comput Sci 27:99–104. https://doi.org/10.1021/ci00055a002

10. Varnek A, Fourches D, Hoonakker F, Solov'ev VP (2005) Substructural fragments: an universal language to encode reactions, molecular and supramolecular structures. J Comput Aided Mol Des 19:693–703. https://doi.org/10.1007/s10822-005-9008-0

11. Hoonakker F, Lachiche N, Varnek A, Wagner A (2011) A representation to apply usual data mining techniques to chemical reactions -- Illustration on the rate constant of S_N2 reactions in water. Int J Artif Intell Tools 20:253–270. https://doi.org/10.1142/S0218213011000140

12. Bishop CM, Svensén M, Williams CKI (1998) GTM: The Generative Topographic Mapping. Neural Computation 10:215–234. https://doi.org/10.1162/089976698300017953

13. De Jong KA (1975) Analysis of the behavior of a class of genetic adaptive systems. University of Michigan

14. Eichler JF, Peeples J (2013) Online Homework Put to the Test: A Report on the Impact of Two Online Learning Systems on Student Performance in General Chemistry. J Chem Educ 90:1137–1143. https://doi.org/10.1021/ed3006264

15. Freasier B, Collins G, Newitt P (2003) A Web-Based Interactive Homework Quiz and Tutorial Package To Motivate Undergraduate Chemistry Students and Improve Learning. J Chem Educ 80:1344. https://doi.org/10.1021/ed080p1344

16. Richards-Babb M, Jackson JK (2011) Gendered responses to online homework use in general chemistry. Chem Educ Res Pract 12:409–419. https://doi.org/10.1039/C0RP90014A

17. Faulconer EK, Griffith JC, Wood BL, et al (2018) A comparison of online and traditional chemistry lecture and lab. Chem Educ Res Pract 19:392–397. https://doi.org/10.1039/C7RP00173H

18. Conole G, Warburton B (2005) A review of computer-assisted assessment. Research in Learning Technology 13:. https://doi.org/10.3402/rlt.v13i1.10970

19. Al-Arimi AMA-K (2014) Distance Learning. Procedia - Social and Behavioral Sciences 152:82–88. https://doi.org/10.1016/j.sbspro.2014.09.159

20. Mohd Hamid SN, Lee TT, Taha H, et al (2021) E-content module for Chemistry Massive Open Online Course (MOOC): Development and students' perceptions. J Technol Sci Educ 11:67. https://doi.org/10.3926/jotse.1074

21. O'Malley PJ, Agger JR, Anderson MW (2015) Teaching a Chemistry MOOC with a Virtual Laboratory: Lessons Learned from an Introductory Physical Chemistry Course. J Chem Educ 92:1661–1666. https://doi.org/10.1021/acs.jchemed.5b00118

22. Nennig HT, Idárraga KL, Salzer LD, et al (2020) Comparison of student attitudes and performance in an online and a face-to-face inorganic chemistry course. Chem Educ Res Pract 21:168–177. https://doi.org/10.1039/C9RP00112C

23. Piketty T (2017) Budget 2018: la jeunesse sacrifiée. Le Monde.fr

24. Piketty T, Chancel L La jeunesse sacrifiée. https://lucaschancel.com/etudiants/. Accessed 4 May 2024

25. Dietrich N, Kentheswaran K, Ahmadi A, et al (2020) Attempts, Successes, and Failures of Distance Learning in the Time of COVID-19. J Chem Educ 97:2448–2457. https://doi.org/10.1021/acs.jchemed.0c00717

26. O'Sullivan TP, Hargaden GC (2014) Using Structure-Based Organic Chemistry Online Tutorials with Automated Correction for Student Practice and Review. J Chem Educ 91:1851–1854. https://doi.org/10.1021/ed500140n

27. Weininger D (1988) SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci 28:31–36. https://doi.org/10.1021/ci00057a005

28. Otálvaro F (2022) Merging Drawing-Based Questions with Automatic Assessment in Organic Chemistry Using Smartphones. J Chem Educ 99:3044–3048. https://doi.org/10.1021/acs.jchemed.2c00278

29. Bienfait B, Ertl P (2013) JSME: a free molecule editor in JavaScript. J Cheminform 5:24. https://doi.org/10.1186/1758-2946-5-24

30. ChemDraw JS Sample Page. https://chemdrawdirect.perkinelmer.cloud/js/sample/index.html. Accessed 25 Apr 2023

31. Heller SR, McNaught A, Pletnev I, et al (2015) InChI, the IUPAC International Chemical Identifier. J Cheminform 7:23. https://doi.org/10.1186/s13321-015-0068-4

32. Socrative home page. https://www.socrative.com/. Accessed 25 Apr 2023

33. Flynn AB, Caron J, Laroche J, et al (2014) Nomenclature101.com: A Free, Student-Driven Organic Chemistry Nomenclature Learning Tool. J Chem Educ 91:1855–1859. https://doi.org/10.1021/ed500353a

34. LeBlond C, Bucholtz E, Muzyka J (2019) OpenOChem: An LMS Agnostic Chemistry Quizzing Platform. In: DivCHED CCCE: Committee on Computers in Chemical Education. http://confchem.ccce.divched.org/2019CCCENLP3. Accessed 13 Dec 2021

35. Todeschini R, Consonni V (2009) Molecular Descriptors for Chemoinformatics, 1st ed. Wiley

36. Wiener H (1947) Structural Determination of Paraffin Boiling Points. J Am Chem Soc 69:17–20. https://doi.org/10.1021/ja01193a005

37. Gutman I, Trinajstić N (1972) Graph theory and molecular orbitals. Total φ-electron energy of alternant hydrocarbons. Chemical Physics Letters 17:535–538. https://doi.org/10.1016/0009-2614(72)85099-1

38. Randic M (1975) Characterization of molecular branching. J Am Chem Soc 97:6609–6615. https://doi.org/10.1021/ja00856a001

39. Daylight Theory: Fingerprints. https://www.daylight.com/dayhtml/doc/theory/theory.finger.html. Accessed 8 Jul 2024

40. Durant JL, Leland BA, Henry DR, Nourse JG (2002) Reoptimization of MDL Keys for Use in Drug Discovery. J Chem Inf Comput Sci 42:1273–1280. https://doi.org/10.1021/ci010132r

41. Solov’ev VP, Varnek A, Wipff G (2000) Modeling of Ion Complexation and Extraction Using Substructural Molecular Fragments. J Chem Inf Comput Sci 40:847–858. https://doi.org/10.1021/ci9901340

42. Bonachéra F, Parent B, Barbosa F, et al (2006) Fuzzy Tricentric Pharmacophore Fingerprints. 1. Topological Fuzzy Pharmacophore Triplets and Adapted Molecular Similarity Scoring Schemes. J Chem Inf Model 46:2457–2477. https://doi.org/10.1021/ci6002416

43. Varnek A, Fourches D, Horvath D, et al (2008) ISIDA - Platform for Virtual Screening Based on Fragment and Pharmacophoric Descriptors. CAD 4:191–198. https://doi.org/10.2174/157340908785747465

44. Oprea TI, Gottfries J (2001) Chemography: The Art of Navigating in Chemical Space. J Comb Chem 3:157–166. https://doi.org/10.1021/cc0000388

45. Papadatos G, Cooper AWJ, Kadirkamanathan V, et al (2009) Analysis of Neighborhood Behavior in Lead Optimization and Array Design. J Chem Inf Model 49:195–208. https://doi.org/10.1021/ci800302g

46. Bajorath J (2001) Selected Concepts and Investigations in Compound Classification, Molecular Descriptor Analysis, and Virtual Screening. J Chem Inf Comput Sci 41:233–245. https://doi.org/10.1021/ci0001482

47. Cherkasov A, Muratov EN, Fourches D, et al (2014) QSAR Modeling: Where Have You Been? Where Are You Going To? J Med Chem

48. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297. https://doi.org/10.1007/BF00994018

49. Breiman L (2001) Random Forests. Machine Learning 45:5–32. https://doi.org/10.1023/A:1010933404324

50. Cover T, Hart P (1967) Nearest neighbor pattern classification. IEEE Trans Inform Theory 13:21–27. https://doi.org/10.1109/TIT.1967.1053964

51. Tropsha A, Isayev O, Varnek A, et al (2024) Integrating QSAR modelling and deep learning in drug discovery: the emergence of deep QSAR. Nature Reviews Drug Discovery 23:141–155

52. Rish I (2001) An empirical study of the naive Bayes classifier. In: IJCAI 2001 workshop on empirical methods in artificial intelligence. Citeseer, pp 41–46

53. Patterson DE, Cramer RD, Ferguson AM, et al (1996) Neighborhood Behavior: A Useful Concept for Validation of “Molecular Diversity” Descriptors. J Med Chem 39:3049–3059. https://doi.org/10.1021/jm960290n

54. Hotelling H (1933) Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology 24:417–441. https://doi.org/10.1037/h0071325

55. Kohonen T (2001) Self-Organizing Maps. Springer Berlin Heidelberg, Berlin, Heidelberg

56. Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. Journal of machine learning research 9:

57. Tropsha A (2010) Best Practices for QSAR Model Development, Validation, and Exploitation. Molecular Informatics 29:476–488. https://doi.org/10.1002/minf.201000061

58. Bishop CM, Svensén M, Williams CKI (1998) Developments of the generative topographic mapping. Neurocomputing 21:203–224. https://doi.org/10.1016/S0925-2312(98)00043-5

59. Gaspar HA, Baskin II, Marcou G, et al (2015) Chemical Data Visualization and Analysis with Incremental Generative Topographic Mapping: Big Data Challenge. J Chem Inf Model 55:84–94. https://doi.org/10.1021/ci500575y

60. Lin A, Baskin II, Marcou G, et al (2020) Parallel Generative Topographic Mapping: An Efficient Approach for Big Data Handling. Molecular Informatics 39:2000009. https://doi.org/10.1002/minf.202000009

61. Sidorov P, Gaspar H, Marcou G, et al (2015) Mappability of drug-like space: towards a polypharmacologically competent map of drug-relevant compounds. J Comput Aided Mol Des 29:1087–1108. https://doi.org/10.1007/s10822-015-9882-z

62. Klimenko K, Marcou G, Horvath D, Varnek A (2016) Chemical Space Mapping and Structure–Activity Analysis of the ChEMBL Antiviral Compound Set. J Chem Inf Model 56:1438–1454. https://doi.org/10.1021/acs.jcim.6b00192

63. Kayastha S, Horvath D, Gilberg E, et al (2017) Privileged Structural Motif Detection and Analysis Using Generative Topographic Maps. J Chem Inf Model 57:1218–1232. https://doi.org/10.1021/acs.jcim.7b00128

64. Tino P, Nabney I (2002) Hierarchical GTM: constructing localized nonlinear projection manifolds in a principled way. IEEE Trans Pattern Anal Machine Intell 24:639–656. https://doi.org/10.1109/34.1000238

65. Lin A, Beck B, Horvath D, et al (2020) Diversifying chemical libraries with generative topographic mapping. J Comput Aided Mol Des 34:805–815. https://doi.org/10.1007/s10822-019-00215-x

66. Pikalyova R, Zabolotna Y, Volochnyuk DM, et al (2022) Exploration of the Chemical Space of DNA-encoded Libraries. Molecular Informatics 41:2100289. https://doi.org/10.1002/minf.202100289

67. Pikalyova R, Zabolotna Y, Horvath D, et al (2023) Chemical Library Space: Definition and DNA-Encoded Library Comparison Study Case. J Chem Inf Model 63:4042–4055. https://doi.org/10.1021/acs.jcim.3c00520

68. Pikalyova R, Zabolotna Y, Horvath D, et al (2023) Meta-GTM: Visualization and Analysis of the Chemical Library Space. J Chem Inf Model 63:5571–5582. https://doi.org/10.1021/acs.jcim.3c00719

69. Vapnik VN (1982) Estimation of dependences based on empirical data. Springer-Verlag, New York

70. Drucker H, Burges CJC, Kaufman L, et al (1996) Support Vector Regression Machines. In: Advances in Neural Information Processing Systems 9, NIPS, Denver, CO, USA, December 2-5, 1996. pp 155–161

71. Schölkopf B, Smola AJ (2002) Learning with kernels: support vector machines, regularization, optimization, and beyond, Reprint. MIT Press, Cambridge, Mass.

72. Katoch S, Chauhan SS, Kumar V (2021) A review on genetic algorithm: past, present, and future. Multimed Tools Appl 80:8091–8126. https://doi.org/10.1007/s11042-020-10139-6

73. Sahingoz OK (2014) Generation of Bezier Curve-Based Flyable Trajectories for Multi-UAV Systems with Parallel Genetic Algorithm. J Intell Robot Syst 74:499–511. https://doi.org/10.1007/s10846-013-9968-6

74. Huang Y, Gao Y, Gan Y, Ye M (2021) A new financial data forecasting model using genetic algorithm and long short-term memory network. Neurocomputing 425:207–218. https://doi.org/10.1016/j.neucom.2020.04.086

75. Deng X, Li M, Deng S, Wang L (2022) Hybrid gene selection approach using XGBoost and multi-objective genetic algorithm for cancer classification. Med Biol Eng Comput 60:663–681. https://doi.org/10.1007/s11517-021-02476-x

76. Ghaheri A, Shoar S, Naderan M, Hoseini SS (2015) The Applications of Genetic Algorithms in Medicine. Oman Med J 30:406–416. https://doi.org/10.5001/omj.2015.82

77. Horvath D, Brown J, Marcou G, Varnek A (2014) An Evolutionary Optimizer of libsvm Models. Challenges 5:450–472. https://doi.org/10.3390/challe5020450

78. Safe M, Carballido J, Ponzoni I, Brignole N (2004) On Stopping Criteria for Genetic Algorithms. In: Bazzan ALC, Labidi S (eds) Advances in Artificial Intelligence – SBIA 2004. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 405–413

79. Aytug H, Koehler GJ (2000) New stopping criterion for genetic algorithms. European Journal of Operational Research 126:662–674. https://doi.org/10.1016/S0377-2217(99)00321-5

80. Whitley D (2001) An overview of evolutionary algorithms: practical issues and common pitfalls. Information and Software Technology 43:817–831. https://doi.org/10.1016/S0950-5849(01)00188-4

81. Harada T, Alba E (2021) Parallel Genetic Algorithms: A Useful Survey. ACM Comput Surv 53:1–39. https://doi.org/10.1145/3400031

82. Pandey HM, Chaudhary A, Mehrotra D (2014) A comparative review of approaches to prevent premature convergence in GA. Applied Soft Computing 24:1047–1077. https://doi.org/10.1016/j.asoc.2014.08.025

83. Friedrich T, Oliveto PS, Sudholt D, Witt C (2009) Analysis of Diversity-Preserving Mechanisms for Global Exploration. Evolutionary Computation 17:455–476. https://doi.org/10.1162/evco.2009.17.4.17401

84. Hussain A, Muhammad YS (2020) Trade-off between exploration and exploitation with genetic algorithm using a novel selection operator. Complex Intell Syst 6:1–14. https://doi.org/10.1007/s40747-019-0102-7

85. Pham DT, Castellani M (2010) Adaptive Selection Routine for Evolutionary Algorithms. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 224:623–633. https://doi.org/10.1243/09596518JSCE942

86. Goldberg DE, Deb K (1991) A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. In: Foundations of Genetic Algorithms. Elsevier, pp 69–93

87. Srinivas M, Patnaik LM (1994) Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans Syst, Man, Cybern 24:656–667. https://doi.org/10.1109/21.286385

88. Hassanat A, Almohammadi K, Alkafaween E, et al (2019) Choosing Mutation and Crossover Ratios for Genetic Algorithms—A Review with a New Dynamic Approach. Information 10:390. https://doi.org/10.3390/info10120390

89. Spiegel JO, Durrant JD (2020) AutoGrow4: an open-source genetic algorithm for de novo drug design and lead optimization. J Cheminform 12:25. https://doi.org/10.1186/s13321-020-00429-4

90. Bai Q, Tan S, Xu T, et al (2021) MolAICal: a soft tool for 3D drug design of protein targets by artificial intelligence and classical algorithm. Briefings in Bioinformatics 22:bbaa161. https://doi.org/10.1093/bib/bbaa161

91. Nigam A, Pollice R, Aspuru-Guzik A (2022) Parallel tempered genetic algorithm guided by deep neural networks for inverse molecular design. Digital Discovery 1:390–404. https://doi.org/10.1039/D2DD00003B

92. Boone K, Wisdom C, Camarda K, et al (2021) Combining genetic algorithm with machine learning strategies for designing potent antimicrobial peptides. BMC Bioinformatics 22:239. https://doi.org/10.1186/s12859-021-04156-x

93. Lunghini F, Marcou G, Azam P, et al (2020) Consensus QSAR models estimating acute toxicity to aquatic organisms from different trophic levels: algae, *Daphnia* and fish. SAR and QSAR in Environmental Research 31:655–675. https://doi.org/10.1080/1062936X.2020.1797872

94. Nekoei M, Mohammadhosseini M, Pourbasheer E (2015) QSAR study of VEGFR-2 inhibitors by using genetic algorithm-multiple linear regressions (GA-

MLR) and genetic algorithm-support vector machine (GA-SVM): a comparative approach. Med Chem Res 24:3037–3046. https://doi.org/10.1007/s00044-015-1354-4

95. Morris GM, Huey R, Lindstrom W, et al (2009) AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. J Comput Chem 30:2785–2791. https://doi.org/10.1002/jcc.21256

96. Jones G, Willett P, Glen RC, et al (1997) Development and validation of a genetic algorithm for flexible docking 1 1Edited by F. E. Cohen. Journal of Molecular Biology 267:727–748. https://doi.org/10.1006/jmbi.1996.0897

97. Verdonk ML, Cole JC, Hartshorn MJ, et al (2003) Improved protein–ligand docking using GOLD. Proteins 52:609–623. https://doi.org/10.1002/prot.10465

98. Jones G (2010) GAPE: An Improved Genetic Algorithm for Pharmacophore Elucidation. J Chem Inf Model 50:2001–2018. https://doi.org/10.1021/ci100194k

99. Richmond NJ, Abrams CA, Wolohan PRN, et al (2006) GALAHAD: 1. Pharmacophore identification by hypermolecular alignment of ligands in 3D. J Comput Aided Mol Des 20:567–587. https://doi.org/10.1007/s10822-006-9082-y

100. (2023) Moodle plugins directory: Chemical substances (Atto). https://moodle.org/plugins/atto_molstructure. Accessed 7 Feb 2024

101. Lecomte J Important Announcement Regarding YUI. In: Yahoo Engineering. https://yahooeng.tumblr.com/post/96098168666/important-announcement-regarding-yui. Accessed 4 May 2024

102. React. https://react.dev/. Accessed 8 Jul 2024

103. Angular. https://angular.dev/. Accessed 8 Jul 2024

104. (2024) Moodle plugins directory: Chemical substance (TinyMCE). https://moodle.org/plugins/tiny_molstructure. Accessed 4 May 2024

105. Richards-Babb M, Curtis R, Georgieva Z, Penn JH (2015) Student Perceptions of Online Homework Use for Formative Assessment of Learning in Organic Chemistry. J Chem Educ 92:1813–1819. https://doi.org/10.1021/acs.jchemed.5b00294

106. Alekseev E, Chesnokova O, Kucher T (2010) Free Pascal and Lazarus - A textbook on programming. ALT Linux library, Moskow

107. Hoonakker F, Lachiche N, Varnek A, Wagner A Condensed Graph of Reaction: Considering a Chemical Reaction As one Single Pseudo Molecule

108. Gimadiev T, Nugmanov R, Batyrshin D, et al (2021) Combined Graph/Relational Database Management System for Calculated Chemical Reaction Pathway Data. J Chem Inf Model 61:554–559. https://doi.org/10.1021/acs.jcim.0c01280

109. Bort W, Baskin II, Gimadiev T, et al (2021) Discovery of novel chemical reactions by deep generative recurrent neural network. Sci Rep 11:3178. https://doi.org/10.1038/s41598-021-81889-y

110. Lin AI, Madzhidov TI, Klimchuk O, et al (2016) Automatized Assessment of Protective Group Reactivity: A Step Toward Big Reaction Data Analysis. J Chem Inf Model 56:2140–2148. https://doi.org/10.1021/acs.jcim.6b00319

111. Gimadiev TR, Madzhidov TI, Nugmanov RI, et al (2018) Assessment of tautomer distribution using the condensed reaction graph approach. J Comput Aided Mol Des 32:401–414. https://doi.org/10.1007/s10822-018-0101-6

112. Madzhidov TI, Gimadiev TR, Malakhova DA, et al (2017) Structure–reactivity relationship in Diels–Alder reactions obtained using the condensed reaction graph approach. J Struct Chem 58:650–656. https://doi.org/10.1134/S0022476617040023

113. Madzhidov TI, Polishchuk PG, Nugmanov RI, et al (2014) Structure-reactivity relationships in terms of the condensed graphs of reactions. Russ J Org Chem 50:459–463. https://doi.org/10.1134/S1070428014040010

114. Afonina VA, Mazitov DA, Nurmukhametova A, et al (2021) Prediction of Optimal Conditions of Hydrogenation Reaction Using the Likelihood Ranking Approach. IJMS 23:248. https://doi.org/10.3390/ijms23010248

115. Glavatskikh M, Madzhidov T, Horvath D, et al (2019) Predictive Models for Kinetic Parameters of Cycloaddition Reactions. Mol Inf 38:1800077. https://doi.org/10.1002/minf.201800077

116. Horvath D, Marcou G, Varnek A, et al (2016) Prediction of Activity Cliffs Using Condensed Graphs of Reaction Representations, Descriptor Recombination, Support Vector Machine Classification, and Support Vector Regression. J Chem Inf Model 56:1631–1640. https://doi.org/10.1021/acs.jcim.6b00359

117. JavaScript With Syntax For Types. https://www.typescriptlang.org/. Accessed 8 Jul 2024

118. Chang C-C, Lin C-J (2011) LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol 2:1–27. https://doi.org/10.1145/1961189.1961199

119. Davies M, Nowotka M, Papadatos G, et al (2015) ChEMBL web services: streamlining access to drug discovery data and utilities. Nucleic Acids Res 43:W612–W620. https://doi.org/10.1093/nar/gkv352

120. Gaulton A, Bellis LJ, Bento AP, et al (2012) ChEMBL: a large-scale bioactivity database for drug discovery. Nucleic Acids Research 40:D1100–D1107. https://doi.org/10.1093/nar/gkr777

121. Mendez D, Gaulton A, Bento AP, et al (2019) ChEMBL: towards direct deposition of bioassay data. Nucleic Acids Research 47:D930–D940. https://doi.org/10.1093/nar/gky1075

122. Zdrazil B, Felix E, Hunter F, et al (2024) The ChEMBL Database in 2023: a drug discovery platform spanning multiple bioactivity data types and time periods. Nucleic Acids Research 52:D1180–D1192. https://doi.org/10.1093/nar/gkad1004

123. Bento AP, Gaulton A, Hersey A, et al (2014) The ChEMBL bioactivity database: an update. Nucl Acids Res 42:D1083–D1090. https://doi.org/10.1093/nar/gkt1031

124. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Ijcai. Montreal, Canada, pp 1137–1145

125. Rodriguez JD, Perez A, Lozano JA (2010) Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. IEEE Trans Pattern Anal Mach Intell 32:569–575. https://doi.org/10.1109/TPAMI.2009.187

126. Kim J-H (2009) Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. Computational Statistics & Data Analysis 53:3735–3745. https://doi.org/10.1016/j.csda.2009.04.009

127. Kireeva N, Baskin II, Gaspar HA, et al (2012) Generative Topographic Mapping (GTM): Universal Tool for Data Visualization, Structure-Activity Modeling and Dataset Comparison. Mol Inf 31:301–312. https://doi.org/10.1002/minf.201100163

128. Jebari K, Madiafi M (2013) Selection methods for genetic algorithms. International Journal of Emerging Sciences 3:333–344

129. Hasançebi O, Erbatur F (2000) Evaluation of crossover techniques in genetic algorithm based optimum structural design. Computers & Structures 78:435–448. https://doi.org/10.1016/S0045-7949(00)00089-4

130. Soon GK, Guan TT, On CK, et al (2013) A comparison on the performance of crossover techniques in video game. In: 2013 IEEE International Conference on Control System, Computing and Engineering. IEEE, Penang, Malaysia, pp 493–498

131. Hsu C-W, Chang C-C, Lin C-J (2003) A practical guide to support vector classification. Taipei, Taiwan

132. Adasme Mora MF, Arcila Toro R, Blackshaw J, et al (2011) CHEMBL database release 34

133. Student (1908) The Probable Error of a Mean. Biometrika 6:1. https://doi.org/10.2307/2331554

134. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Computat 1:67–82. https://doi.org/10.1109/4235.585893

135. Rogers D, Hahn M (2010) Extended-Connectivity Fingerprints. J Chem Inf Model 50:742–754. https://doi.org/10.1021/ci100050t

136. Landrum G (2013) RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling. Greg Landrum 8:5281

137. Pedregosa F, Varoquaux G, Gramfort A, et al (2011) Scikit-learn: Machine learning in Python. the Journal of machine Learning research 12:2825–2830

138. Bayr H (2005) Reactive oxygen species: Critical Care Medicine 33:S498–S501. https://doi.org/10.1097/01.CCM.0000186787.64500.12

139. Bergamini C, Gambetti S, Dondi A, Cervellati C (2004) Oxygen, Reactive Oxygen Species and Tissue Damage. CPD 10:1611–1626. https://doi.org/10.2174/1381612043384664

140. Liou G-Y, Storz P (2010) Reactive oxygen species in cancer. Free Radical Research 44:479–496. https://doi.org/10.3109/10715761003667554

141. Miller NJ, Rice-Evans C, Davies MJ, et al (1993) A Novel Method for Measuring Antioxidant Capacity and its Application to Monitoring the Antioxidant Status in Premature Neonates. Clinical Science 84:407–412. https://doi.org/10.1042/cs0840407

142. Idrovo-Encalada AM, Rojas AM, Fissore EN, et al (2023) Chemoinformatic modelling of the antioxidant activity of phenolic compounds. J Sci Food Agric 103:4867–4875. https://doi.org/10.1002/jsfa.12561

143. Deng W, Chen Y, Sun X, Wang L (2023) AODB: A comprehensive database for antioxidants including small molecules, peptides and proteins. Food Chemistry 418:135992. https://doi.org/10.1016/j.foodchem.2023.135992

144. Howard SJ, Catchpole M, Watson J, Davies SC (2013) Antibiotic resistance: global response needed. The Lancet Infectious Diseases 13:1001–1003. https://doi.org/10.1016/S1473-3099(13)70195-6

145. Magréault S, Jauréguy F, Carbonnelle E, Zahar J-R (2022) When and How to Use MIC in Clinical Practice? Antibiotics 11:1748. https://doi.org/10.3390/antibiotics11121748

146. Bemis GW, Murcko MA (1996) The Properties of Known Drugs. 1. Molecular Frameworks. J Med Chem 39:2887–2893. https://doi.org/10.1021/jm9602928

8 Appendix

8.1 Appendix 1: Manual of the question type plugins

Manual of the Question type plugins of the ChemMoodle project: MolSimilarity, ReacSimilarity

Here we present a documentation for both "Molsimilarity" and "Reacsimilarity", two question type plugins created for the Moodle platform. MolSimilarity introduces a graded scoring system to evaluate students' responses when they draw chemical structures automatically. This method assigns a score to the response based on the graph similarity between the student's answer and the correct solution. Reacsimilarity, incorporates the concept of CGR (Condensed Graph of Reaction). When a response to a question involves a chemical transformation, it is represented by a pseudo-molecular graph (the CGR) where modified bonds and atoms are labeled. This representation allows for a graded scoring system similar to the MolSimilarity plugin.

For the MolSimilarity plugin, the student is asked to draw a chemical structure. With ReacSimilarity, the student is asked to draw a reaction equation with the corresponding atom-to-atom mapping (AAM). It consists in assigning to each reactant's atom a unique number and the same unique number to the corresponding atom in the products. . For a given reaction, alternative correct AAM are possible.

Installation of the plugins

To install these two plugins, we recommend following Moodle documentation available on https://moodle.org/. Additional specific documentation for the installation of the grading servers is available on the GitHub of the two question type plugins.

Creation of a question

To create a question in a Moodle quiz, select the type of question needed to be added. In our case, we will select MolSimilarity or ReacSimilarity (Figure 1).

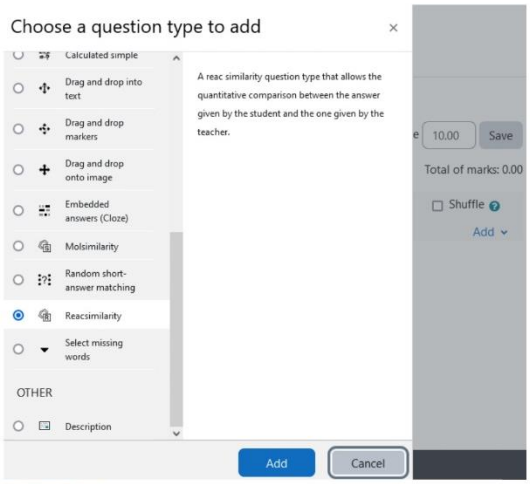

Figure 1. Selection of the question type.

For both plugins, the teacher may draw 1 to N structures or reaction equations that are considered correct. It allows to accommodate for chemical ambiguities (mesomeres, tautomers, etc.). A teacher can incorporate a starting template, representing a part of the molecular structure, or reaction equation. For the ReacSimilarity plugin, the starting template may include some already mapped atoms.

We use the similarity between the molecules to obtain a mark, and the mark is therefore proportional to the similarity between the structure of the students and that of the teachers. If several alternative answers are possible, a maximal similarity score is taken.

The process of drawing the structures is similar to using other chemical drawing tools. In the ReacSimilarity plugin, to add the AAM, the user should selects the “Reaction mapping” arrow from the arrow toolset, click on a reactant atom, and drag to the corresponding product atom (Figure 2).

Figure 2. Atom-to-atom mapping arrow position on the toolset.

To insert an answer, the teacher needs to draw a structure or reaction equation in the corresponding sketcher (“Correct answers”), and click on the “Insert given structure as answer/update the answer with the structure” button, as shown in the Figure 3.

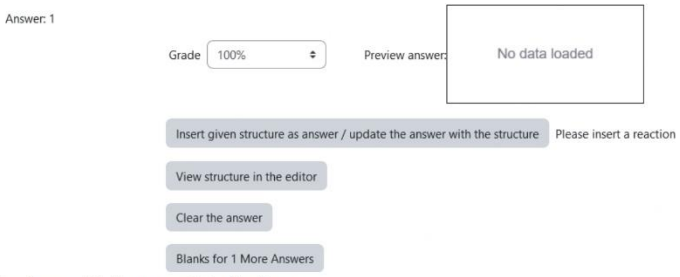

Figure 3. Answer interface, with the associated buttons.

Once a structure or reaction has been added to an answer, the preview is updated, and the “Please insert a reaction”/”Please insert a molecule” text disappears (Figure 4).

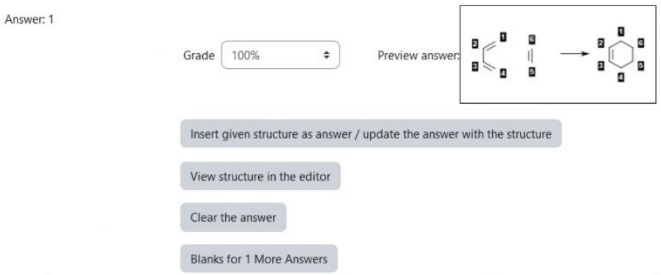

Figure 4. Reaction equation added to the answer 1. The preview of the answer is updated to show it to the teacher.

The “View structure in the editor” button allows to insert the structure or reaction equation associated with the answer in the “Correct answers” sketcher, to modify it. Once modified, the teacher will need to click on the “Insert given structure as answer/update the answer with the structure” button one more time.

To insert a new structure or reaction equation to be considered as correct by the correction algorithm, the teacher needs to click on the “Blanks for 1 More Answers” button. It will create a new answer interface, where the teacher will be able to insert a new structure or reaction equation. To do so, they will draw it in the “Correct answer” sketcher, and click on the “Insert given structure as answer/update the answer with the structure” button, in the corresponding answer interface.

To remove an answer, the teacher needs to set the grade to “Clear the answer”, and click on the “Clear the answer” button. Then, for the action to be taken into account, they will need to Click on the “save changes and continue editing” or “save changes” button.

These two question types have specific options:

- Should a starting structure given to the student ?
- Should the stereochemistry be considered in the correction procedure ?
- Should an alpha parameter and a threshold be used for the computation of the final grade ?

If a starting structure must be given to the student, the teacher needs to draw it in the corresponding sketcher (“You can insert a scaffold for the student.”), and the grading will assess the similarity between the template and the teacher’s answers that yielded the maximal similarity (g_{tpl}), even if a student gives no answer at all. In order to take the template into account, a final score is modified according to the formula bellow, where g_{sim} is a similarity score related to the student’s answer.

$$g_{rest} = \begin{cases} \dfrac{(g_{sim} - g_{tpl})}{(1 - g_{tpl})}, & if\ \text{“reaction template”} \\ g_{sim}, & if\ no\ \text{“reaction template”} \end{cases}$$

If the stereochemistry is considered, the grade will be computed according to the following formula:

$$g_{rest} = \begin{cases} \dfrac{\#Correct\ Stereo\ Center}{\#Total\ Stereo\ Center}, & if\ similarity\ score\ = 1 \\ 0, & if\ similarity\ score\ \neq 1 \end{cases}$$

Once the grade is computed, both the alpha parameter and the threshold t will be used to calculate the final grade according to the following formula:

$$g = \begin{cases} (g_{rest})^{\alpha}, & if\ (g_{rest})^{\alpha}\ \geq t \\ 0, & otherwise \end{cases}$$

By default, t is equal to 0 and has a range going from 0 to 1, and α is equal to 1 and has a range going from 0.1 to 10. These parameters allow to modulate the softness of the grading: for small values of α, more errors will be tolerated while large values will deteriorate the grade for any deviation from the expected answer.

8.2 Appendix 2: Manual of the pGA-GTM/pGA-SVM

Manual of the pGA-GTM/pGA-SVM

Authors : L. Plyer, G. Marcou, F. Bonachera, A. Varnek

Table des matières

1. Introduction

The pGA-GTM and pGA-SVM tools are distributed inside the same software, an executable, easy to use command line Genetic Algorithm (GA) [1, 2]. GA is a type of meta-heuristic within the broader category of evolutionary algorithms (EA), drawing inspiration from natural selection processes.

It can be used in two applications:

- For Generative Topographic Mapping (GTM) [3] using the GTM implementation in pascal [4] where the goal is to find the chromosome leading to the manifold best representing the data-set.
- For Support Vector Machine (SVM) [5] or Support Vector Regression (SVR) [6], using libsvm software [7], where the goal is to find the chromosome leading to the best classification or regression model.

As for the libSVM-GA [8] it has been inspired of, meta-parameters of the SVM/GTM and the Descriptor Spaces (DS) are optimized at once. Indeed, as the meta-parameters are often dependent of the DS, they need to be optimized considering the DS.

In a GA, every candidate is defined by a chromosome, which is a vector of N dimensions, N being the number of parameters requiring optimization. The algorithm randomly generates a set of starting chromosomes, evaluated using the fitness function, which calculates the fitness score (FS) of a given chromosome.
During the execution of the GA, chromosomes will generate “children” using crossovers and mutations.
The termination criteria for the GA include either reaching the maximum number of attempts (can be modified in the parameters) or the absence of FS improvement of the best individual over the last N generations (N can be modified in the parameters).

Furthermore, for the pGA-GTM, we compute the PCA (Principal Component Analysis) that are used to initialize the manifold once at the initialization step of the GA for a given DS, and not for each FS calculation, saving computation time.

For both pGA-SVM and pGA-GTM, we reduced as much as possible the number of disk accesses, relying on function calls instead of system calls, in order to speed up the calculations.

Several architectures of GA are available for the user, including single thread generational GA, non-redundant, adaptive [9], and multi-threaded application of a steady state GA.

2. Installation and usage

a. Installation and setup

To use this software, both the GAproject executable and the libsvm shared objects files need to be in the same repository. Make sure that the executable has the right to be executed and to write/read files (you can use chmod +777 for that).

b. User guide

When launching a GA, in both cases (pGA-SVM and pGA-GTM), you will need to have in the same repository the following files:

- GAproject
- libsvm.so.2
- rng file (GTM.rng or SVMClass.rng/SVMReg.rng)
- descriptor files for the descriptors in the rng file. For the GTM, the descriptor file needs to be in svm format, and for the SVM, it needs to be under the psvm format (first column is the property to be modelled).

During the execution of both GA, multiples files and folders will be written:

- 'alreadySeen.hist', with the chromosomes that have already been seen and their associated FS value.
- 'locusseentwice.hist', a list of the chromosomes that have been seen more than one time by the GA, for analysis.
- By default, 'outfile.hist', a file containing one line for each generation, with the best individual, its FS, the elapsed time since the beginning of the calculation (in CPU time), and the mean FS of the population.
- 'outfile.pop', created when the GA reaches the stopping criteria. Contains the line of command to use with the libsvm tool to reproduce the result of the given chromosome.
- 'outfile_m', created when the GA reaches the stopping criteria. Machine readable chromosome and FS value, to restart the GA with the given pre-computed generation.
- 'BackupGen' folder, containing the population at a given generation (the frequency can be modified in the options), both for external usage, and to start the GA with a pre-computed generation, under two formats:
    - Gen-x: Command line to use with libsvm/GTMapTool to reproduce these results.
    - Gen-x_m: Machine readable, chromosome and FS value, to restart the GA with the given pre-computed generation.
- 'BestPop' folder, containing the best individuals of the last generation, and their according model (see **Output of the GA.**).

- For the pGA-SVM, 'prop.stddev', precalculated standard deviation of the property/class modeled.

- For the pGA-SVM, ‘descrset.EDprops’ for each of the *descrset*, containing means, over training set instance pairs, of Euclidean distance and dot product values.
- For the pGA-GTM, ‘PCAcalculation’ folder, containing the precalculated PCA for each descriptor set.

c. Command line options.

Here are the command line options. Some are used only with the pGA-SVM and some only with the pGA-GTM.
First the common option will be given, then the specific one for each GA.

Shared Options:

***-h, --help*:** Displays help information, detailing the usage and options available within the tool.

-s, --strategy: Selects the GA strategy to be used. Options include:

pGA-GTM	*pGA-SVM*
1: parallelGAv3	4: parallelGAv3 SVM
2: parallelGAv3Adaptive	5: parallelGAv3Adaptive SVM
3: parallelGAv3NoDuplicate	6: parallelGAv3NoDuplicate SVM

***-f, --rng*:** Specifies the GA problem space file. Please don’t modify the order of the columns of provided .rng files.

***-p, --popsize*:** Sets the size of the population. The default is 100.

***-g, --gen*:** Sets the maximum number of generations. The default is 3000.

***-o, --output*:** Specifies the base name for output files.

***-c, --count*:** The maximum number of generations without progress before stopping the GA. The default is 50.

***-u,--procs*:** Determines the number of threads for parallelized GA. The default uses all available threads up to a maximum of popsize.

***-e, --seedFile*:** Specifies an input seed file to start the population from, typically from the BackupGenfolder, under the Gen-x_m format.

***-b, --backupInt*:** Sets the generation interval at which backups are created. The default is 5.

***-x, --crossrate*:** The crossover rate for the GA, applicable except for adaptive strategies. The default is 0.3.

***-a, --murate*:** The mutation rate for the GA, applicable except for adaptive strategies. The default is 0.1.

***-r, --redunlim*:** Sets the chromosome redundancy limit, a value between 0and1. The default is 0.05.

***-n, --numbermdl*:** Sets the number of models to be calculated from the last generation. The default value is 10.

***--MXV*:** Sets the number of repetitions of XV to be performed in the fitness function. The default value is 3.

***--NXV*:** Sets the number of folds for the XV to be performed in the fitness function. The default value is 3.

pGA-SVM specific options:

-m, --mode: Sets the mode for SVM, either CLASSIFICATION or REGRESSION. The default is CLASSIFICATION.

-t, --test: Designates if a test file should be used. It will be used to do an external test on the ***--numbermdl*** individuals of the last generation of the GA. Careful, if the option is set, you need to have a test set in each of the given descriptors set.

--minperfstop: min performance under which a model fails to fit during training. The default value is 0.0.

pGA-GTM specific options:

-l, --classprop: Classification properties to be used for the CV procedure, as a comma separated list.

-k, --regprop: Regression properties to be used for the CV procedure, as a comma separated list.

-q, --scoringset: Should an 'external' set used for the training of the class/reg GTM. Careful, if the option is set, you need to have a test property set in each of the given descriptors set.

d. File format.

LibSVM format

The libSVM format is used for sparse datasets. Each line is an instance, and the first column is a property value or class that shall be numerical. Then, the line is composed of space separated doublet. Each doublet is composed of an integer value and a double value separated by a column.

```
<p> <int>:<double> <int>:<double> … <int>:<double>
```

Anatomy of a SVM line.

The integer is an identifier of an attribute, and the double is the value of the attribute. An example of attribute is a molecular descriptor.

```
? 1:1 4:3 80:3 83:1 1476:0
? 1:4 2:2 5:4 56:4 90:2 92:4 94:2 231:4 1476:0
? 1:2 2:1 89:1 90:1 126:1 132:2 1476:0
? 1:4 4:2 56:4 58:4 69:2 71:4 73:2 301:4 1476:0
? 1:1 4:2 80:1 89:1 406:2 408:1 1476:0
```

Example of an SVM file containing 5 molecules.

For the pGA-SVM, a *psvm* file is needed as input, meaning that the first column must always contain the property or class of the molecule. Using classification, please use -1 and 1 as classes identifiers. For n > 2 multi-classifications, please use 0 to n as classes identifiers.

For the pGA-GTM, a svm file is needed as input and the software supports the character "?" for missing values. It also supports the properties to be strings without blanks.

Property file format

For the pGA-GTM, to color the map in classification or regression tasks, a property file with the assigned classes or property values is needed. Such file should be as a TXT file, where the first line is a free comment. Beginning from the second line all classes or property values are signed for each molecule from the training set one per each line. In classification tasks it is mandatory to use only integers as a class name. Moreover, no empty lines should be added between the lines or in the end of the file.

```
*****MY COMMENT – CLASSIFICATION/REGRESSION************
-1.0
-1.0
-1.0
-0.79
-0.73
```

Example of a .prp file containing the properties of 5 molecules.

e. File nomenclature.

Adhering to a specific file naming convention is imperative for effective functionality of the GA. A noteworthy difference is the file extension of the descriptor files, as for the libSVM GA, the descriptor sets are in *psvm* format.

pGA-SVM

For each of the descriptor set present in the *.rng* file, a file called '*descrset.psvm'* should be present in the directory where the GA is being executed.
If the ***-t, --test*** is set to true, a file called '*test_descrset.psvm'* should be present for each of the descriptor set present in the *.rng* file.

pGA-GTM

For each of the descriptor set present in the *.rng* file, a file called '*descrset.svm'* should be present in the directory where the GA is being executed.
If no scoring set is used, for each of the properties given as input with the options ***-l, --classprop*** or ***-k, --regprop***, a file called '*property.prp*' must be present.

If the ***-q, --scoringset*** is set to true, a file called '*scoringset_descrset.svm'* should be present for each of the descriptor set present in the *.rng* file. For each of the properties given as input with the options ***-l, --classprop*** or ***-k, --regprop***, a file called '*scoringset_property.prp*' must be present.

f. Output of the GA.

Both for the pGA-GTM and the pGA-SVM, once the GA reaches the stopping criteria, a folder 'BestPop' will be created. It contains n folders, one for each of the n first chromosomes, as defined by the option **-n, --numbermdl**, under the following format: 'Chromo-n'. Each of these sub-folders contain models, statistics and predicted properties of the training or test set if provided.

pGA-SVM

For the pGA-SVM, the GA creates the following files when it reaches the stopping criteria :

- *'stats'*, contains statistics about the model. It contains both statistics on XV (conducted under the same parameters as for the fitness function calculation) and statistics on train set (or test set if the option is used) prediction. The file follows the given format:

```
train:meanQ2xv =       0.828
train:maxQ2xv =        0.858
train:minQ2xv =        0.795
train:stddevQ2xv=      0.015
              .
              .
              .
train:R2xv_BCFlog =    0.849
train:RMSExv_BCFlog =  0.852
```

Example of statistics computed for a given chromosome, for a regression task with the statistics of a given property.

- *'descrset_model'*, the model created training on all the training set, given the meta-parameters of the chromosome.
- *'descrset_pred'* by default / *'descrset_testPred'* if the option '***-t true***' is used. This file contains the values of the predicted training or test set properties by applying the created model.

pGA-GTM

For the pGA-GTM, the GA creates the following files when it reaches the stopping criteria :

- *'stats'*, contains statistics about the model. It contains both statistics on XV (conducted under the same parameters as for the fitness function calculation) and statistics on training (or test set if the option is used) prediction. On XV, it contains statistics for all the properties, classification and regression sets as a mean, and separately. The file follows the given format:

```
train:meanBAxv =       0.828
train:maxBAxv =        0.858
train:minBAxv =        0.795
train:stddevBAxv=      0.015

              .
              .
              .
train:BA =             0.849
train:Accu =           0.852
train:Sensi =          0.825
train:Speci =          0.873
```

Example of statistics computed for a given chromosome, for a classification task.

- *'descrset.xml'*, manifold of the map trained given the meta-parameters of the chromosome.
- *'descrsetR.svm'*, responsibilities of the compounds of the training set.
- *'scoringset_descrsetR.svm'* if the option ***–scoringset*** is used.
- *'descrset+property+_reg.xml', 'descrset+property+_cls.xml',* regression and classification landscapes. One file for each property.
- *'Dens.mat', 'LS.mat', 'StdDevLs.mat'* files, one for each property.

3. Fitness functions

Both fitness functions aim at creating models that can predict new instances rather than just fitting the training data. We use a M-times repeated N-folds cross-validation (XV) with reshuffling of training set instances [10, 11]. By default, we propose to use the values of M=3 and N=3, which is less biased toward the composition of the dataset but estimating performances with a larger variance. Of course, these details of the validation procedure are left to the preferences of the end users and other choices can be met in the literature [11, 12], depending on the context.
The fitness function used in this optimization considers the average XV performance. However, the "best" model observed at a given moment of the GA is only better for a given repartition of the data between the training and test sets of the cross-validation. Therefore, models that perform less well may be equivalent or even better than the top-performing ones. To identify models that are statistically indistinguishable from the "best" model, a threshold is defined using the repeatability of the performances across a moderate repetition of the cross-validation procedure.

For both pGA-SVM and pGA-GTM, we reduced as much as possible the number of disk accesses, relying on function calls instead of system calls, in order to speed up the calculations.

a. pGA-SVM

For SVM or SVR, using libsvm software [7], the goal of the pGA-SVM is to find the chromosome leading to the best classification or regression model. The chromosome encodes for the DS to use, the value of the cost parameter, the kernel to use, the gamma, coeff0 values, and the epsilon value for the regression. When the chromosome needs to be evaluated, the value of the gene needs to be "translated" to libsvm parameters.
As the DS chosen may modify highly the vector representation of each compound of the dataset, we needed to address the variability induced to the gamma parameter. The gamma is a kernel parameter scaling as the standard deviation of the instances in the data space. Therefore, to be integrated in a GA, it needs to be standardized. As in libsvm-GA [8], a preliminary "gamma factor" is encoded in the gene. To translate it to libsvm parameter, it is divided by the average Euclidean distance or dot product values over training set pairs, depending on the kernel used, allowing for a reasonable range of gamma. These metrics are computed for each DS at the initialization step of the GA. The epsilon parameter for regression tasks is in the same units as the target property to model. For this reason, the actual epsilon value is calculated by multiplying the epsilon parameter by the standard deviation of the training property values. The cost parameter is encoded using a log-scale conversion and the actual cost value is obtained by taking the exponential of the cost parameter from the chromosome.

The objective of this GA is to simultaneously optimize the meta-parameters of SVM/SVR problems, alongside determining the DS, with SVM parameters being contingent upon the chosen DS. If any model fails to fit during training (R2 XV_M < minperfstop), it is immediately discarded (the solution gets a FS of 0.00), to speed the computation process, by discarding "bad" models. By default, minperfstop is equal to 0.

For that reason, the FS is calculated as follows:
- For each M N-fold XV: shuffle the training set and N-fold XV (default: M=3, N=3)
- Compute R^2/BA for this XV step.
- At the end of the M N-fold XV: Calculate the mean $\langle BA \rangle$ or $\langle R^2 \rangle$ and standard deviation σ of the M M N-fold XV: $\sigma(BA)$ or $\sigma(R^2)$.
- The FS is defined as $\langle R^2 \rangle - 2\sigma$ for the SVR and $\langle BA \rangle - 2\sigma$ for the SVC.

b. pGA-GTM

For GTM, using the in house GTM implementation in Pascal [4], the goal is to find the chromosome leading to the manifold best representing the dataset. The chromosome encodes for the DS to use, the number of RBFs, the width of the RBFs, and the regularization parameter. As for the pGA-SVM, the genes need to be translated to GTM parameters. The number of RBFs is taken from equation (1) where nb_{mol} is the

number of molecules in the frameset, and $gene_{RBF}$ is the value of the gene encoding for the number of RBFs.

$$nRBF = Min(floor\left(\frac{nb_{mol}}{2}\right), gene_{RBF} \times 10) \quad (1)$$

In the fitness function of the pGA-GTM, the descriptor sets *'descrset.svm'* (one for each of the descriptor set present in the *.rng* file), are used as frameset (data set used for unsupervised training of the manifold) to train the manifold using GTMapTool.

The regularization parameter is equal to 10 to the power of the gene value, and the width of the RBFs is taken directly from the value of the gene. The number of nodes is taken as 25 times the number of RBFs. Unlike in the previous implementation of the in-house GA, we compute the PCA that are used to initialize the manifold once at the initialization step of the GA for a given DS, and not for each FS calculation, saving computation time.
The difference with the pGA-SVM lies in the fact that the pGA-GTM is a multi-objective optimization. Indeed, to give a FS to each manifold, they are evaluated on each property given by the user (on each of the landscape created).

If the option ***-q, --scoringset*** is set to true, the 'external' set '*scoringset_descrset.svm'* is projected on the manifold and used for the calculation of the FS.
If the option ***-q, --scoringset*** is not used, the frameset will be projected on the manifold and used for the calculation of the FS.

The value of the gene encoding for the DS is used as frameset for unsupervised training of the manifold. To create a landscape, the user has the option to provide a scoring set of annotated compounds, projected on the manifold and used for the calculation of the FS. Otherwise, the compounds used for manifold creation will be projected on it for landscape creation.

For that reason, the FS is calculated as follows:

- The manifold is initialized and trained using the frameset.
- Frameset or scoring set are projected on the trained manifold.
- For each property given by the user: M x N-fold XV (by default: M=3, N=3).
- Compute R^2/BA for this XV step.
- At the end of the M N-fold XV: Calculate the mean $\langle BA \rangle$ or $\langle R^2 \rangle$ and standard deviation σ of the M $\sigma(BA)$ or $\sigma(R^2)$.
- For each P property compute a score $S = \langle R^2 \rangle - 0{,}5\sigma$ for the regression and $S = \langle BA \rangle - 0{,}5\sigma$ for the classification.
- The FS is defined as $\langle S \rangle$.

4. Conclusions

This document aims to explain the new flexible GA for optimization of GTM and SVM/SVR tasks, offering a wide range of strategies and parameters to fine-tune the GA process according to specific requirements and objectives.
To summarize, we have covered several aspects: we discussed the fitness functions employed in both pGA-GTM and pGA-SVM, how to use the command line, defined the options, and depicted the output of both GA.

5. List of abbreviations.

GA Genetic Algorithm
EA Evolutionary Algorithm
GTM Generative Topographic Mapping
SVM Support Vector Machine
SVR Support Vector Regression
DS Descriptor Spaces
PCA Principal Component Analysis
FS Fitness Score
XV Cross-Validation

6. Bibliography

1. Holland JH (1975) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. Univ. of Michigan Press, Ann Arbor

2. De Jong KA (1975) Analysis of the behavior of a class of genetic adaptive systems. University of Michigan

3. Bishop CM, Svensén M, Williams CKI (1998) GTM: The Generative Topographic Mapping. Neural Computation 10:215–234. https://doi.org/10.1162/089976698300017953

4. Kireeva N, Baskin II, Gaspar HA, et al (2012) Generative Topographic Mapping (GTM): Universal Tool for Data Visualization, Structure-Activity Modeling and Dataset Comparison. Mol Inf 31:301–312. https://doi.org/10.1002/minf.201100163

5. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory. ACM, Pittsburgh Pennsylvania USA, pp 144–152

6. Drucker H, Burges CJC, Kaufman L, et al (1996) Support Vector Regression Machines. In: Advances in Neural Information Processing Systems 9, NIPS, Denver, CO, USA, December 2-5, 1996. pp 155–161

7. Chang C-C, Lin C-J (2011) LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol 2:1–27. https://doi.org/10.1145/1961189.1961199

8. Horvath D, Brown J, Marcou G, Varnek A (2014) An Evolutionary Optimizer of libsvm Models. Challenges 5:450–472. https://doi.org/10.3390/challe5020450

9. Srinivas M, Patnaik LM (1994) Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans Syst, Man, Cybern 24:656–667. https://doi.org/10.1109/21.286385

10. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: Ijcai. Montreal, Canada, pp 1137–1145

11. Rodriguez JD, Perez A, Lozano JA (2010) Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation. IEEE Trans Pattern Anal Mach Intell 32:569–575. https://doi.org/10.1109/TPAMI.2009.187

12. Kim J-H (2009) Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. Computational Statistics & Data Analysis 53:3735–3745. https://doi.org/10.1016/j.csda.2009.04.009

Louis PLYER

Construction d'outils pédagogiques pour la Chimie par des approches Chémoinformatique

Résumé

Cette thèse est consacrée au développement et à la mise en œuvre d'outils open-source innovants visant à améliorer l'enseignement de la chimie via la plateforme Moodle. Le projet ChemMoodle comprend quatre plugins : deux pour la notation automatique des questions à l'aide d'un système de notation doux, et deux pour l'affichage d'informations chimiques telles que les structures 2D et 3D et les spectres pour les étudiants. De plus, un algorithme génétique a été développé pour simplifier la sélection des valeurs optimales pour les paramètres libres de la cartographie topographique générative (GTM) et de la machine à vecteur de support (SVM). Cet algorithme permet d'optimiser les méta-paramètres sans l'intervention d'un expert.

Résumé en anglais

This thesis is dedicated to the development and implementation of innovative open-source tools aimed at enhancing chemical education through the Moodle platform. The ChemMoodle project includes four plugins: two for automatic grading of questions using a smooth grading system, and two for displaying chemical information such as 2D and 3D structures and spectrums to students. Additionally, a genetic algorithm was developed to simplify the selection of optimal values for the free parameters of Generative Topographic Mapping (GTM) and Support Vector Machine (SVM). This algorithm allows for a meta-parameter optimization without the need for expert intervention.