Chebifier 2: An Ensemble for Chemistry

Simon Flügel[1,*], Martin Glauer[2], Janna Hastings[3,4,5], Till Mossakowski[1], Christopher J. Mungall[6], Charlotte Tumescheit[3,5] and Fabian Neuhaus[2]

[1]Institute for Computer Science, University of Osnabrück, Neuer Graben 29, 49074 Osnabrück, Germany
[2]Institute for Cooperating Systems, Otto von Guericke University Magdeburg, Universitätsplatz 2, 39106 Magdeburg, Germany
[3]Institute for Implementation Science in Health Care, University of Zurich, Universitätsstrasse 84, 8006 Zürich, Switzerland
[4]School of Medicine, University of St. Gallen, (HSG), St. Jakob-Strasse 21, 9000 Gallen, Switzerland
[5]Swiss Institute of Bioinformatics, Lausanne, Switzerland
[6]Division of Environmental Genomics and Systems Biology, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

Abstract

Accurately classifying chemical structures is a central task in cheminformatics. Different approaches to classification exist, but they all come with their own drawbacks. Manual classification is time-intensive and hard to scale up to large databases. Rule-based methods are specific to some classes and require a high level of maintenance. Deep learning models lack reliability and explainability.

In this work, we present an ensemble learning method that combines generative artificial intelligence, classical deep learning and symbolic approaches to classify chemicals into the Chemical Entities of Biological Interest (ChEBI) ontology.

Our approach covers 1,722 classes with vastly different properties. The ensemble makes use of the strengths of each model and aligns their predictions with the OWL axiomatisation of ChEBI. We provide both a Python library as well as a web front-end which allow users to classify arbitrary molecules with the ensemble.

Keywords

ChEBI, chemical classification, ensemble learning, ontology extension

1. Introduction

Over the past decades, the amount of knowledge available in the life sciences has grown exponentially. Making use of this knowledge requires organisation and the connection of new findings to existing knowledge. For example, the Chemical Entities of Biological Interest (ChEBI) ontology [1, 2] provides a manually curated classification hierarchy for chemicals with 222,172 entries (as of June 2025). While manual curation allows ChEBI to enforce a high level of quality, it cannot keep up with other chemistry databases such as PubChem [3] which contains 122 million compounds (as of June 2025) and grows by roughly 300,000 entries per month.

There is therefore an urgent need for an automated extension of ChEBI. So far, different methods have been proposed for this task. Rule-based methods [4, 5, 6, 7] are successful if applied to specific classes. Since the rules are developed by chemical experts, they can achieve a similar level of accuracy as human curators and even reduce errors. ClassyFire [4] in particular has become an essential tool assisting ChEBI development. However, setting up rules still requires a significant amount of human labour – in most cases, it is not possible to trivially extend a rule-based approach to new classes. In addition, not all classes in ChEBI are defined by structural features that can easily be captured with rules that are based on pattern matching against the chemical structure or easily computable chemical properties. An alternative approach is provided by machine learning [8, 9, 10], which does not require hand-crafted rules. One advantage of machine learning methods is that existing models may be retrained

SymGenAI4Sci 2025: First International Workshop on Symbolic and Generative AI for Science co-located with Semantics-2025, September 3–5, 2025, Vienna, Austria

*Corresponding author.
simon.fluegel@uni-osnabrueck.de (S. Flügel)
0000-0003-3754-9016 (S. Flügel); 0000-0001-6772-1943 (M. Glauer); 0000-0002-3469-4923 (J. Hastings); 0000-0002-8938-5204 (T. Mossakowski); 0000-0002-6601-2165 (C. J. Mungall); 0000-0002-7563-5575 (C. Tumescheit); 0000-0002-1058-3102 (F. Neuhaus)

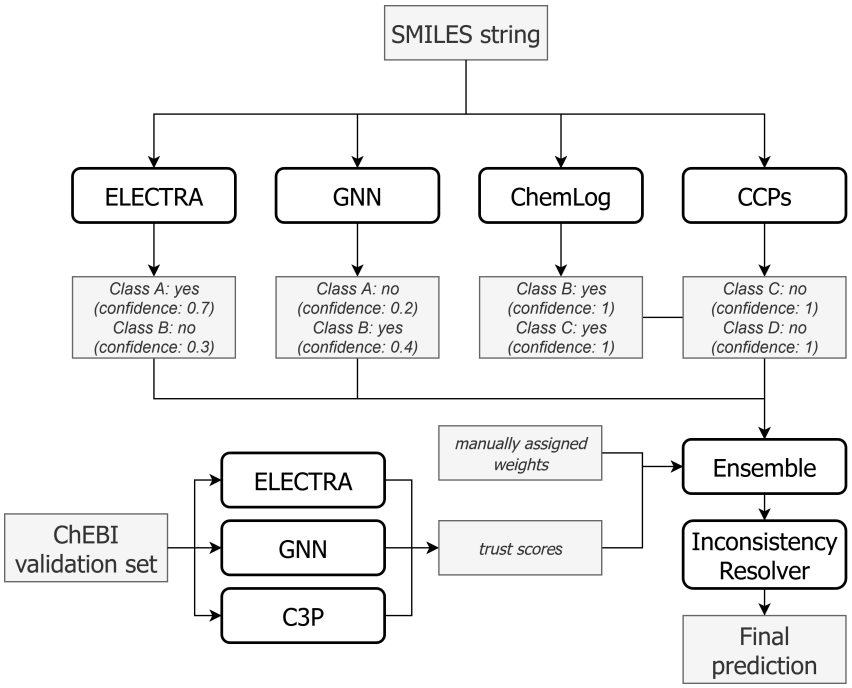

Figure 1: Architecture of our ensemble. The Classes A-D are exemplary and do not refer to actual ChEBI classes.

on new classes with minimal additional effort. However, one disadvantage of many machine learning approaches is their data-dependency. Especially deep learning models fail to learn complex chemical classes with few available samples. Another disadvantage is that they lack interpretability: deep learning models inherently are black-boxes. While efforts towards a more human-understandable classification have been made [10], this still remains an open research topic. A third approach, based on generative AI has been suggested in [11], which uses a large language model to generate Python programs that classify chemicals. This approach has a better interpretability since it provides natural-language explanations for each classification.

Since the different approaches are typically most performant for different kinds of chemical classes [8], we developed Chebifier 2, an ensemble that integrates different models for classifying chemical entities (represented as SMILES strings) into classes from ChEBI. Chebifier 2 integrates the results from different models that represent all three aforementioned approaches. To combine the different predictions, we have developed a weighting mechanism and apply automatic inconsistency resolution. In the following, we give a short overview over the models that are part of the ensemble. Then, we describe the ensemble aggregation mechanism before moving on to the inconsistency resolution.

2. Methodology

The architecture of our proposed ensemble model is shown in Figure 1. As an input, we use SMILES strings [12], a compact linear representation of molecules. The SMILES string is then passed to the 4 models which independently make predictions and report their decisions and confidence scores to the ensemble. For the learned models, an additional reliability score ("trust") has been calculated on a ChEBI validation set. This, as well as manually assigned weights, are used by the ensemble to make predictions for each class for which at least one model has made a decision (cf. Section 2.2). In the final step, these class-wise predictions are verified against the ChEBI axiomatisation and corrected if necessary (cf. Section 2.3). In total, the ensemble covers 1,722 classes, 1,528 of which are predicted by at least two different models.

2.1. Integrated models

ELECTRA. ELECTRA [13] is a transformer model trained on SMILES strings with two tasks: A self-supervised pre-training task in which a discriminator network has to decide whether a given token is part of the original SMILES string or has been generated by a generator network. For this task, we use SMILES strings from PubChem. In the second task, the discriminator is trained on a ChEBI classification task. Here, all SMILES-annotated classes of ChEBI (version 241) are taken as samples and all ChEBI classes with at least 50 SMILES-annotated subclasses as labels. This results in a dataset with 1,528 labels and 187,293 samples which is split into a training, validation and test set with an 80/10/10 ratio. A previous version of the ELECTRA model has been described in [9]. Training has been conducted with the ChEB-ai library.[1]

Graph Neural Network (GNN). In addition to ELECTRA, we use another deep learning model, more specifically a Residual Gated Graph Convolutional Network [14]. In contrast to the transformer model, the GNN works on a graph representation of the molecule instead of the SMILES string. Fur this purpose, SMILES strings are converted into a graph representation using RDKit.[2] Skipping the pre-training step, the GNN is trained on the same ChEBI classification task as ELECTRA (1,528 labels). Training has been conducted with ChEB-ai's graph extension.[3]

ChemLog. ChemLog[4] is a rule-based classification tool described in [7] It is based on a monadic second-order logic formalisation of 169 classes of molecules. The logical definitions are used to generate (and validate) code that classifies molecules automatically. ChemLog covers 18 peptide-related classes discussed in [7]. In this work, we extend ChemLog by 155 classes that are defined by either the presence of a chemical element (e.g., CHEBI:51143 *nitrogen molecular entity*) or by a bond between a carbon atom and a given chemical element (e.g., CHEBI:51185 *organoiron compound*). Out of the 173 classes predicted by ChemLog in total, 128 are not predicted by any of the other models.

Chemical Classifier Programs (CCPs). Based on the natural language definitions of ChEBI, large language models (LLMs) are used to generate CCPs, which are customized Python scripts that use the RDKit library to classify chemicals [11]. CCPs are generated in an iterative process, where molecules sampled from ChEBI are used to prompt the LLM to refine its original classification program. Alongside the predictions, natural-language explanations for each possible classification result are returned. The implementation is available on Github.[5] The CCPs integrated in Chebifier 2 cover 338 chemical classes, 66 of which are unique to this approach.

2.2. Ensemble aggregation

The ensemble aggregates all model predictions into a single ensemble prediction. Given an input molecule (i.e., a SMILES string) and models $m_1, m_2, \ldots, m_n$, the following steps are performed: First, the predictions $\mathbf{p^m} \in \{0, 1, -1\}^{C_m}$ from each model m are collected. Here, C_m is the number of classes predicted by m. Note that this number may vary depending on the SMILES string. For the deep learning models, the confidence scores $\mathbf{f^m}$ for each prediction are computed as well. Given the direct model output $o_c^m \in [0, 1]$, the prediction is defined via a threshold θ and the confidence via the distance to this threshold (usually, $\theta = 0.5$ is used):

$$p_c^m = \begin{cases} 1 & \text{if } o_c^m > \theta \\ 0 & \text{otherwise} \end{cases} \qquad f_c^m = \begin{cases} \frac{o_c^m - \theta}{1-\theta} & \text{if } o_c^m > \theta \\ \frac{\theta - o_c^m}{\theta} & \text{otherwise} \end{cases} \tag{1}$$

[1] https://github.com/ChEB-AI/python-chebai
[2] https://www.rdkit.org/
[3] https://github.com/ChEB-AI/python-chebai-graph
[4] ChemLog peptides: https://github.com/sfluegel05/chemlog-peptides, extension: https://github.com/ChEB-AI/chemlog-extra
[5] https://github.com/chemkg/c3p

For ChemLog and CCPs, the confidence scores are set to 1 as these models make binary decisions and don't judge their own confidence depending on a given SMILES string. In a second step, predictions are aggregated for each class c, resulting in an ensemble prediction p_c^e. $p_c^{m_i} = -1$ is used for errors and therefore ignored during the aggregation. However, if all models return -1 for all classes, an error message is displayed. The aggregation is based on three weights. The *confidence* $f_c^{m_i}$ discussed above which is reported by the models themselves for a given sample. The performance of models on a validation set is measured as well, which is independent of the current sample. We call this *trust* $t_c^{m_i}$. For the deep learning models and the CCPs trust is calculated as an F1-score-based metric

$$t_c^m = 1 + \frac{2TP_c^m}{2TP_c^m + FP_c^m + FN_c^m} \tag{2}$$

where TP_c^m, FP_c^m and FN_c^m are the numbers of true positives, false positives and false negatives across the validation set. In addition, it is possible to manually assign a weight w^{m_i} to specific models. The final ensemble prediction combines all the factors discussed:

$$p_c^e = \begin{cases} 1 & \text{if } \sum_m [p_c^m \cdot f_c^m \cdot t_c^m \cdot w^m] > \sum_m [(1 - p_c^m) \cdot f_c^m \cdot t_c^m \cdot w^m] \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

2.3. Inconsistency resolution

After a decision has been made for each class independently, we use the logical axioms of ChEBI to check for inconsistencies. There are two types of inconsistencies. (1) Our ensemble predicts that a molecule belongs to class A, but not B, in spite of the fact that A is a subclass of B according to ChEBI. (2) Our ensemble predicts that a molecule belongs both to class A and class B, in spite of the fact that A and B are disjoint classes in ChEBI. In both cases, we revisit two predictions of the ensemble and change them with the goal of achieving logical consistency. However, a change which resolves a logical conflict with some axioms in ChEBI may lead to new conflicts with other axioms in ChEBI. Thus, we implemented an iterative approach which ensures the consistency of predictions with ChEBI.

3. Summary

We have introduced Chebifier 2, an ensemble learning methodology for chemical classification. Chebifier 2 combines generative AI-based classifiers with rule-based and deep learning classifiers. This results in a powerful ensemble model that can predict 1722 ChEBI classes, harnessing the strengths of each approach and carefully weighting their contributions. The ensemble can be easily expanded to new classes and is available both as a Python library as well as via a web frontend.

In future work, we aim to add more models to the ensemble. For instance, a range of classical AI methods has been applied to ChEBI in [8]. Using ELECTRA, we will try ensemble learning strategies such as bagging and boosting and add specialised models trained on subgraphs of the ChEBI hierarchy. Both ChemLog and the CCPs have a significant potential for the inclusion of new classes as well. Here, we will harness the existing ensemble to identify classes and molecule types which might benefit from specialised approaches. While many of the individual models of the ensemble have already been evaluated, future work will include a comparison between models and an evaluation of the full ensemble.

Acknowledgments

This work has been funded by the Deutsche Forschungsgesellschaft (DFG, German Research Foundation) - 522907718 and 456666331 and by the Swiss National Science Foundation (SNF) - 215906. Additionally, CJM was funded by the Genomic Science Program in the U.S. Department of Energy, Office of Science, Office of Biological and Environmental Research (BER) under contract numbers DE-AC02–05CH11231.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] K. Degtyarenko, P. De Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, R. Alcántara, M. Darsow, M. Guedj, M. Ashburner, ChEBI: a database and ontology for chemical entities of biological interest, Nucleic acids research 36 (2007) D344–D350.

[2] J. Hastings, G. Owen, A. Dekker, M. Ennis, N. Kale, V. Muthukrishnan, S. Turner, N. Swainston, P. Mendes, C. Steinbeck, ChEBI in 2016: Improved services and an expanding collection of metabolites, Nucleic acids research 44 (2016) D1214–D1219.

[3] S. Kim, J. Chen, T. Cheng, A. Gindulyte, J. He, S. He, Q. Li, B. A. Shoemaker, P. A. Thiessen, B. Yu, L. Zaslavsky, J. Zhang, E. E. Bolton, PubChem 2025 update, Nucleic Acids Res. 53 (2025) D1516–D1525.

[4] Y. Djoumbou Feunang, R. Eisner, C. Knox, L. Chepelev, J. Hastings, G. Owen, E. Fahy, C. Steinbeck, S. Subramanian, E. Bolton, R. Greiner, D. S. Wishart, ClassyFire: Automated chemical classification with a comprehensive, computable taxonomy, Journal of Cheminformatics 8 (2016) 61. URL: https://doi.org/10.1186/s13321-016-0174-y. doi:10.1186/s13321-016-0174-y.

[5] S. Flügel, M. Glauer, F. Neuhaus, J. Hastings, When one logic is not enough: Integrating first-order annotations in OWL ontologies, Semantic Web 16 (2025) SW–243440.

[6] O. Kutz, J. Hastings, T. Mossakowski, Modelling highly symmetrical mmolecules: Linking ontologies and graphs, in: International Conference on Artificial Intelligence: Methodology, Systems, and Applications, Springer, 2012, pp. 103–111.

[7] S. Flügel, M. Glauer, T. Mossakowski, F. Neuhaus, ChemLog: Making MSOL viable for ontological classification and learning, in: 5th International Joint Conference on Learning and Reasoning (accepted), preprint available on arXiv, 2025. URL: http://arxiv.org/abs/2507.13987. doi:10.48550/arXiv.2507.13987.

[8] J. Hastings, M. Glauer, A. Memariani, F. Neuhaus, T. Mossakowski, Learning chemistry: Exploring the suitability of machine learning for the task of structure-based chemical ontology classification, Journal of Cheminformatics 13 (2021) 1–20.

[9] M. Glauer, F. Neuhaus, S. Flügel, M. Wosny, T. Mossakowski, A. Memariani, J. Schwerdt, J. Hastings, Chebifier: Automating semantic classification in ChEBI to accelerate data-driven discovery, Digital Discovery 3 (2024) 896–907.

[10] M. Glauer, A. Memariani, F. Neuhaus, T. Mossakowski, J. Hastings, Interpretable ontology extension in chemistry, Semantic Web 15 (2024) 937–958.

[11] C. J. Mungall, A. Malik, D. R. Korn, J. T. Reese, N. M. O'Boyle, J. Hastings, Chemical classification program synthesis using generative artificial intelligence, arXiv preprint arXiv:2505.18470 (2025).

[12] D. Weininger, SMILES, a chemical language and information system, Journal of Chemical Information and Computer Sciences 28 (1988) 31–36.

[13] K. Clark, M.-T. Luong, Q. V. Le, C. D. Manning, ELECTRA: Pre-trainin text encoders as discriminators rather than generators, 2020. URL: http://arxiv.org/abs/2003.10555. doi:10.48550/arXiv.2003.10555, arXiv:2003.10555 [cs].

[14] X. Bresson, T. Laurent, Residual gated graph convnets, 2018. URL: http://arxiv.org/abs/1711.07553. doi:10.48550/arXiv.1711.07553, arXiv:1711.07553 [cs].

A. Online Resources

The implementation of our ensemble can be found at https://github.com/ChEB-AI/python-chebifier. An interactive website for chemical classification can be found at https://chebifier.hastingslab.org/. The

website also provides explanations of individual classifications for ChemLog and CCPs. Trained models and ensemble weights are available at https://zenodo.org/records/16263057.