Research Preprint

# Data Driven Estimation of Molecular Log-Likelihood using Fingerprint Key Counting

Esben Jannik Bjerrum[1*]

Chemical similarity between two molecules finds widespread use in drug discovery and material science, being utilized for similarity search, toxicological assessment, and as a foundation for QSAR models. This study describes models for the estimation of the log-likelihood for a given molecule to belong to a specific dataset, representing a form of similarity between a single molecule and a given dataset. Two different models are derived based on simple counting of fingerprint keys in the molecule and collected statistics for the total number of observations in the dataset. The AtomLL model is shown to be useful for detecting outliers with unusual keys and demonstrates the greatest baseline performance for class membership assignment. The MolLL model can detect outliers with an unusual number of repeats and is also beneficial for keeping de novo molecular generation and optimization in scope. Their performance is compared to a kernel density estimator model based on molecular descriptors. The model code and some precomputed models are available as open source on GitHub.

## Introduction

Chemical similarity is widely used in cheminformatics, but sometimes the question arises about how similar a compound is to a dataset. Given a set of molecules, e.g., actives, how likely is this other molecule to be part of that set? In other scenarios, it can be of interest to identify outliers and molecular structures that do not belong to a given set. QSAR and QSPR classifier models have been used for this purpose, but usually also utilize an alternative class, i.e., inactives or low-actives. The goal is to assign a given compound to either of these two classes. Moreover, in recent years, de novo generation and molecular optimization have been hot, but not without their criticism as generative models can have a tendency to generate out-of-scope compounds that are clearly not part of the applicability domain or intended aim of the generation, e.g., drug-like small molecules[1]. Adding additional restraints on the generation, such as SA-score[2] and QED score[3]can help ameliorate the out-of-scope generation as shown by Steinmann and Jensen[4]. SA-score is a compound score based on statistics about Morgan fingerprints of radius two, together with a complexity-based score, and assigns a score to molecules based on how easy they are to synthesize. SA-score has been used as a popular scoring component for de novo generation in recent years, with the aim of generating synthesizable compounds. SA-score has also been modified by researchers at Roche for use as a targeted restraint or post-generation filter[5]. As an alternative to using fingerprint keys, molecular properties or descriptors can be utilized. The "drug-beauty score," QED[3], quantifies drug-likeness by fitting desirability functions to a range of molecular properties: molecular weight, calculated lipophilicity, polar surface area, and the number of hydrogen bond donors and acceptors, rotatable bonds, aromatic rings, and structural alert counts.

This paper explores a simple fingerprint key counting-based approach to analyze existing datasets and then assigns likelihoods of molecules belonging to them based on the molecules' own keys. This provides a log-likelihood estimation that can be used to rank a given molecule's likelihood of belonging to the dataset. The MolLL code allows for easy analysis of custom datasets for specific needs using either fingerprint or property-based approaches. Only one dataset is needed, and the resulting models are demonstrated to be useful for outlier detection, exhibit basic performance for assigning class membership, and can efficiently support genetic algorithm optimization of molecules to generate reasonable molecules. The code is open-source and available on GitHub https://github.com/EBjerrum/molll

1. *Cheminformania.com, Mölndal, Sweden*
* Corresponding author: cheminformania@gmail.com

# Methods

## Log-Likelihood models

### *AtomLL model*

The AtomLL model utilizes a fingerprint key count-based approach to assign the likelihood of a given observation of a fingerprint, as shown in Equation 1. The basic premise is that the likelihood of observing a given key is proportional to the number of times it was observed in the dataset divided by the total number of observations. To smooth the counts, a smoothing count is added to all counts, and the denominator is adjusted by multiplying the smoothing count with the number of theoretical keys. This allows for the assignment of (low) likelihoods to keys that were not observed during the analysis of a dataset, thereby preventing prediction failures. Since the total number of keys depends on a complex combination of allowed atoms, bond types, and the radius of the fingerprint, it is simply estimated in practice as a number set to be larger than the number of observations.

$$\text{(1)} \qquad Likelihood\ of\ key\ observation = \frac{n_{key\ observations} + smoothing}{n_{total\ observations} + smoothing * n_{possible\ keys}}$$

The log-likelihood of a given molecule is then found as the sum of the logarithms of the likelihoods of its constituent atoms, as shown in equation (2).

$$\text{(2)} \qquad AtomLL\ of\ molecule = \sum_{i=0}^{n_{heavy\ atoms}} \log\left(Likelihood\ of\ key\ of\ atom\ i\right)$$

To correct for the size of molecules, a tunable correction term is included, which divides the log-likelihood of the molecule by the number of atoms, as shown in Equation (3).

$$\text{(3)} \qquad Corrected\ AtomLL\ of\ molecule = \frac{AtomLL\ of\ mol}{n_{heavy\ atoms}^{alpha}}$$

### *MolLL model*

The MolLL model not only takes into account how many times the key was independently observed in the dataset but also considers how many times it was observed in a given molecule. Therefore, observing a common key an unusual (e.g., large) number of times in a given molecule will still lead to a low likelihood. Apart from the more detailed accounting of counts, the smoothing and correction for size are done as for the AtomLL model, as shown in Equations 1 to 3. Additionally, a likelihood is assigned to the molecule given the total number of observed keys in a molecule, efficiently encoding the size of the molecules. After the analysis of the keys and counts in the dataset, the observed counts for each key for two observations and above are smoothed using the geometric mean of the neighboring count observations.

### *PropLL model*

The property-based model utilizes the kernel density estimator from scikit-learn[6], together with the MolecularDescriptorTransformer from scikit-mol[6]. A log-likelihood estimate can thus be obtained from any molecular descriptor supported by RDKit[7]. The descriptor(s) are first calculated with the scikit-mol transformer, then scaled with a scikit-learn robust scaler before being fitted to a kernel density estimation using the kernel density estimator from scikit-learn. The bandwidth parameter from the kernel estimator is tunable and set as default for 0.1. The bandwidth is used after the robust scaler, so the same bandwidth can be applied for all properties.

## Datasets

*Zinc:* A dataset of ZincDB[8] molecules was created by downloading 2D, Reaction Standard, In-Stock purchasable tranches from the ZincDB webpage[9], and sampling 100,000 compounds uniformly.

*ChEMBL:* The dataset of ChEMBL assay 3888429, targeting Tyrosine-protein kinase JAK2, has previously been processed[1] and was downloaded from GitHub[10].

*Dopamine receptor D2 (D2R):* ExcapeDB[11] was downloaded from Zenodo[12], and rows with the gene symbol DRD2 were extracted, checked for RDKit validity, non-isomeric canonical SMILES strings made with RDKit, and the dataset filtered for duplicates. A balanced dataset was then produced by taking all rows with an activity flag of "A" (actives) and sampling without replacement a similar number of rows with an activity flag "N".

*LibInvent:* LibInvent training data processed and filtered from ChEMBL as previously described[13] were downloaded from Zenodo[14]. The ChEMBL train files were processed by only loading the column with the full SMILES string and removing duplicates. An additional smaller dataset was obtained by sampling 10,000 SMILES strings randomly.

## Outlier detection

The Zinc dataset was analyzed with MolLL and AtomLL models with a radius of two, a smoothing factor of 0.1, an observation count correction (alpha) of 1.0, and an estimated keyspace of two million. Subsequently, all molecules from the Zinc dataset were scored with the models, and the score distributions were examined with histograms. All plots were created using Matplotlib[17]. The lowest scoring compounds and molecules closest to the median score were visualized using RDKit.

## Class-membership analysis

The D2R dataset was loaded and randomly split into a train set (75%) and a test set (25%). Afterwards, MolLL and AtomLL models with radii between 1 and 3, smoothing of 0.1, and alpha of 1.0 with estimated keyspaces of 2 million were created based on the train-set actives or inactives, respectively. The log-likelihood scores of both the test-set actives and inactives were calculated with all the models, and the resulting score distributions were examined with histograms and scatter plots. The difference between the test-set scores from the models based on the active and inactive train-set was calculated and also plotted with histograms. The overlap between the histograms was quantified using the Wasserstein distance (Earth mover distance) as implemented in the SciPy[15] stats module. All plots were created with Matplotlib[16].

## Molecular optimization with a genetic algorithm

The ChEMBL data was utilized to build a QSAR model predicting directly from SMILES strings by combining a Ridge regression model from scikit-learn[17] in a pipeline with the SmilesToMolTransformer and MorganFingerprintTransformer from scikit-mol[6]. Radius 3 and counts were used for featurization, and three-fold cross-validation and a grid search on logarithmically spaced values from 0.1 to 10,000 with fifty values were used to select an optimal alpha value based on the negative RMSE score. The model had a training score (coefficient of determination) of 0.68 and a test score of 0.36, indicating only moderate predictive performance.

Scoring functions for maximization were defined as listed in Table 1. The thresholds for the cutoff-based scoring functions were determined by eye-balling the distribution histograms of the ChEMBL data scored with the restraint models (SI figures 1 to 7) based on the LibInvent data. The radius was varied between 1 and 2 as noted in Table 1, but the smoothing was kept fixed at 0.1, and the alpha exponent at 1.0. The bandwidth for the PropLL model was set at 0.1. Weights in the scoring functions were adjusted by hand to roughly balance out the different response scales of the score components.

*Table 1: Scoring function variants used for the genetic algorithm optimization.*

| Scoring Function Label | Functional form |
|---|---|
| unrestricted | QSAR-score |
| atomll_r1 | QSAR-score + 3 * AtomLL(radius=1) |
| atomll_r1_cutoff | QSAR-score * (AtomLL(radius=1) > -5.75) |
| atomll_r2 | QSAR-score + 3 * AtomLL(radius=2) |
| atomll_r2_cutoff | QSAR-score * (AtomLL(radius=2) >  -8) |
| moll_r1 | QSAR-score + 3 * MolLL(radius=1) |
| moll_r1 | QSAR-score * (MolLL(radius=1) > -13) |
| moll_r2 | QSAR-score + 3 * MolLL(radius=2) |
| moll_r2 | QSAR-score * (MolLL(radius=2) > -11) |
| propll | QSAR-score + 3 * PropLL |
| propll_cutoff | QSAR-score * ( PropLL > -1.75 ) |
| atomll_r1_propll | QSAR-score + 3 * AtomLL(radius=1) + 3 * ProLL |
| atomll_r2_propll | QSAR-score + 3 * AtomLL(radius=2) + 3 * ProLL |

Mol-ga was downloaded from GitHub[18], and the code was modified to abort the run if the average number of heavy atoms in the offspring surpassed 100. The reporting statistics were furthermore expanded to also report on the properties of each batch of offspring in addition to the default reporting of the properties of the population. Optimization was performed in triplicate for each variant of the scoring function and started from three different sets of 1000 randomly picked Zinc compounds, using the sampling function as implemented in Mol-ga. The sets were kept the same for all triplicate runs of the variant scoring functions. The offspring size was set to 100, and the population size to 1000, with 1000 optimization steps.

The run data were analyzed for the progression in the number of heavy atoms in the offspring, the average QSAR score, as well as the fraction of molecules that would be excluded if the NIBR filters[19] had been used. The NIBR filters were applied using the implementation in the RDKit contrib folder. The progression was smoothed as noted using the Savitzky-Golay filter[20] as implemented in the SciPy[15] signal module with an exponent of one and a window of 50 on each series of data before computing the average and standard deviation between the triplicate runs.

## Results

### Outlier detection

One use-case for molecular likelihood estimation is to identify unusual molecules in a given dataset. After analyzing the Zinc dataset with the log-likelihood estimators, the molecules were self-evaluated using log-likelihood score estimation for belonging to the Zinc dataset. A histogram with inlaid molecules of the distribution obtained with a MolLL estimator for a sample from ZincDB is depicted in Figure 1, and the 8 molecules with the lowest log-likelihood are shown in Figure 2. The molecule with the lowest log-likelihood is a carbohydrate or close analogue, likely obtaining the low score due to an unusually high number of similar fingerprint keys from the hydroxy groups. Other molecules with low log-likelihood also have many repeats in substructures, such as lipids or PEG-chains. In contrast, the molecules around the median log-likelihood shown in Figure  3 seem less unusual, even though they display a variety of functional motifs. As ZincDB is a database of molecules that should be ready for docking or screening campaigns, one can judge if the molecules with the lowest LL should be included in the docking or virtual screening, or the MolLL could be used to filter the entire dataset based on a defined threshold.

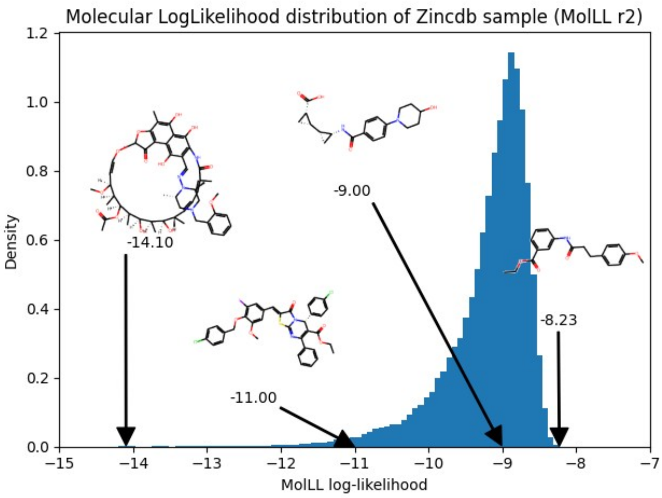


*Figure 1: Distribution of the log-likelihoods obtained from a random sample of 100,000 ZincDB compounds. Example molecular structures are inset with their molecular log-likelihood annotated.*

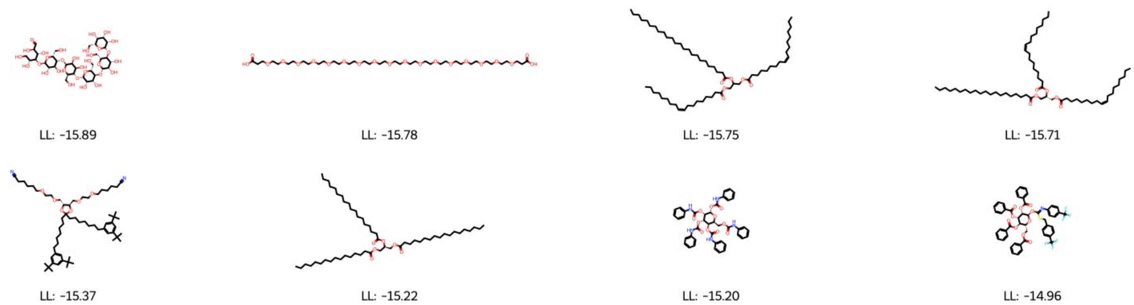


*Figure 2: Examples of molecules with the lowest log-likelihood using a MolLL model with a radius of two from the random sample of 100,000 molecules from ZincDB. Repeating structures and large sizes seem to be characteristic.*

Using the AtomLL estimator, the outliers look as shown in Figure 4. The dataset is the same as for the other figures, but the results are strikingly different from the ones obtained with the MolLL estimator. Unusual atoms or local combinations of atoms are evident. The two models can thus be used to highlight different unusual features of molecules, and can likely be used synergistically to identify outliers in datasets.

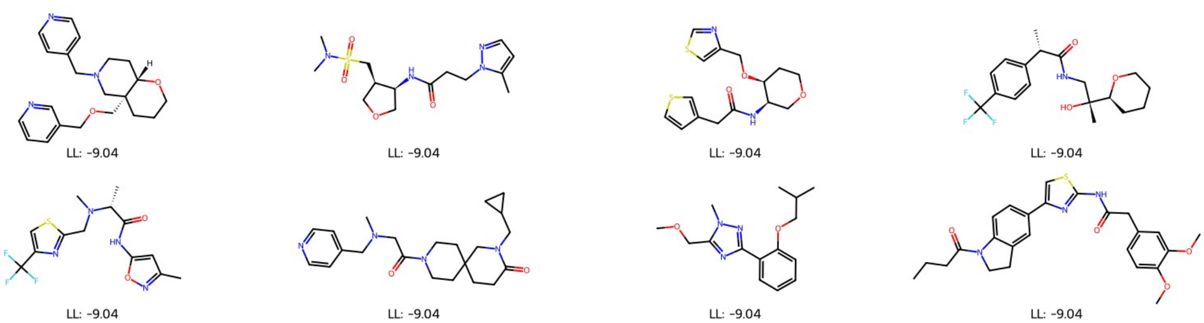


*Figure 3: Molecules with a median log-likelihood according to the MolLL estimator with a radius of two obtained from the random sample of 100,000 ZincDB compounds. A variety of heterocycles and functional groups are seen, even though the log-likelihood is the same.*

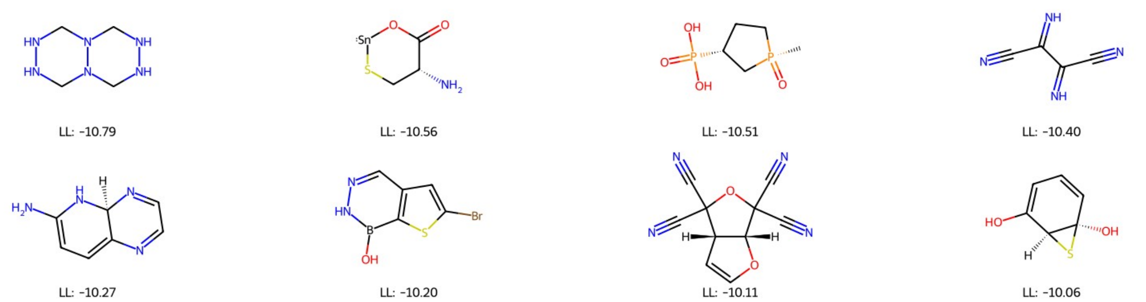


*Figure 4: Molecules with low log-likelihood according to the AtomLL estimator with a radius of two. Unusual local environments or combinations of atoms are striking features.*

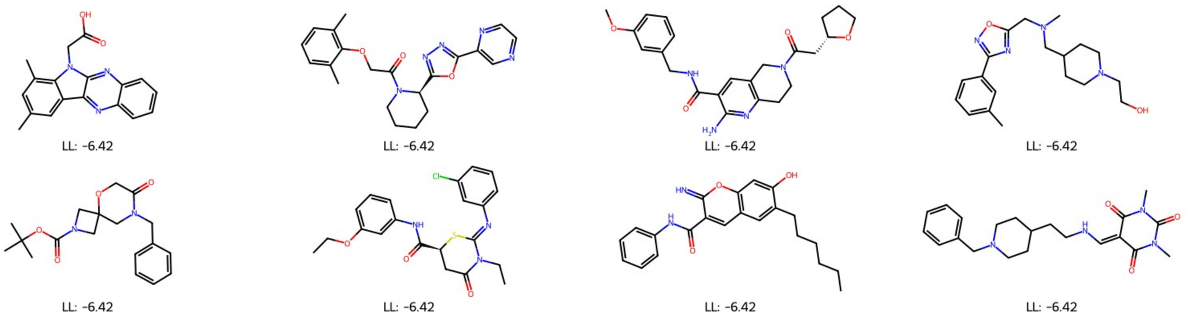


*Figure 5: Molecules with median log-likelihood according to the AtomLL estimator with a radius of two.*

## Class membership

Datasets often contain labeled classes of molecules, such as active and inactive, as measured in a given activity or binding assay. One such example is the D2R dataset from the ExcapeDB, which contains molecules labeled as either active or inactive. It was thus of interest to see if the log-likelihood estimation could be used as a way to gauge class membership in a given dataset. The dataset was prepared, and the classes were balanced via undersampling of the majority class (inactives). Figure 6 shows histograms of the test set log-likelihoods as predicted with either the estimator obtained by analysis of the train set of actives or inactives. Some separation is evident for the LL estimator based on the active train set, but the overlap between distributions is weaker for the LL estimator based on the inactive train set.

The direct comparison, however, hides the fact that log-likelihood estimations and thus the distributions are not independent of each other but rather correlated. Molecules with a low log-likelihood for one estimator are also likely to be low for other estimators. This is evident from the scatter plot shown in Figure 7, where the correlation is seen for both the active and inactive test sets. Instead, the difference between the two log-likelihood estimators can be used, and these difference distributions are shown in Figure 8, where the separation between the classes is much more pronounced. All histogram and scatter plots can be found in the Supplementatry information Figures 8 to 25.

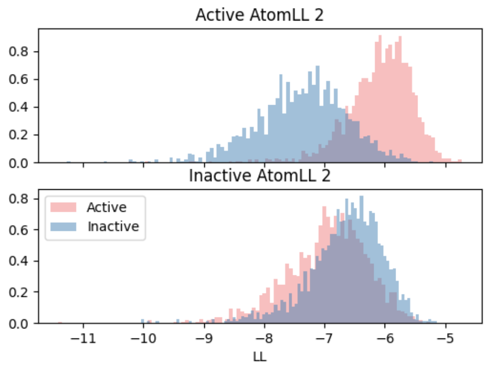


*Figure 6: Log-likelihood distributions for the test set classes using the AtomLL radius 2 estimators obtained from active and inactive train sets, respectively.*

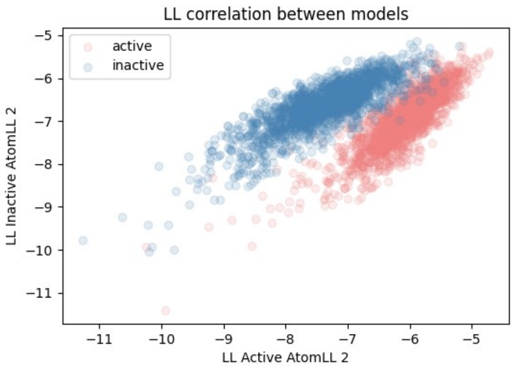

*Figure 7: Log-likelihood scatter plots for test set classes using the AtomLL radius 2 estimators obtained from active and inactive train sets, respectively.*

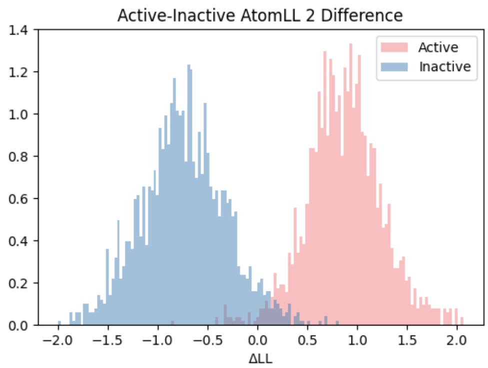

*Figure 8: Difference log-likelihood histograms for test set classes using the difference of AtomLL radius 2 estimators obtained from active and inactive train sets.*

The Wasserstein distances were calculated as a metric for the overlap between histograms for both AtomLL and MolLL models using different radii for the fingerprint, as shown in Table 2. Irrespective of the model and radius, the inactive likelihood estimators show the highest overlap in distributions, with the delta distribution having the lowest. Also listed in Table 2 are the recall, precision, and accuracy that can be found with the simple class prediction approach, where the class for a given sample is the same as the likelihood estimator with the highest log-likelihood. Higher key radii seem to benefit both the separation of the likelihood scores and the accuracy of the class membership prediction.

*Table 2: Distribution overlap and class classification metrics for different estimators.*

| | AtomLL 1 | AtomLL 2 | AtomLL 3 | MolLL 1 | MolLL 2 | MolLL 3 |
|---|---|---|---|---|---|---|
| WD active model | 0.764 | 1.296 | 1.641 | 0.331 | 0.621 | 0.830 |
| WD inactive model | 0.035 | 0.314 | 0.570 | 0.130 | 0.319 | 0.434 |
| WD delta models | 0.773 | 1.609 | 2.211 | 0.460 | 0.938 | 1.264 |
| Recall | 0.977 | 0.987 | 0.987 | 0.858 | 0.920 | 0.947 |
| Precision | 0.913 | 0.955 | 0.968 | 0.907 | 0.942 | 0.960 |
| Accuracy | 0.942 | 0.970 | 0.977 | 0.885 | 0.932 | 0.953 |

### Molecular optimization with a genetic algorithm

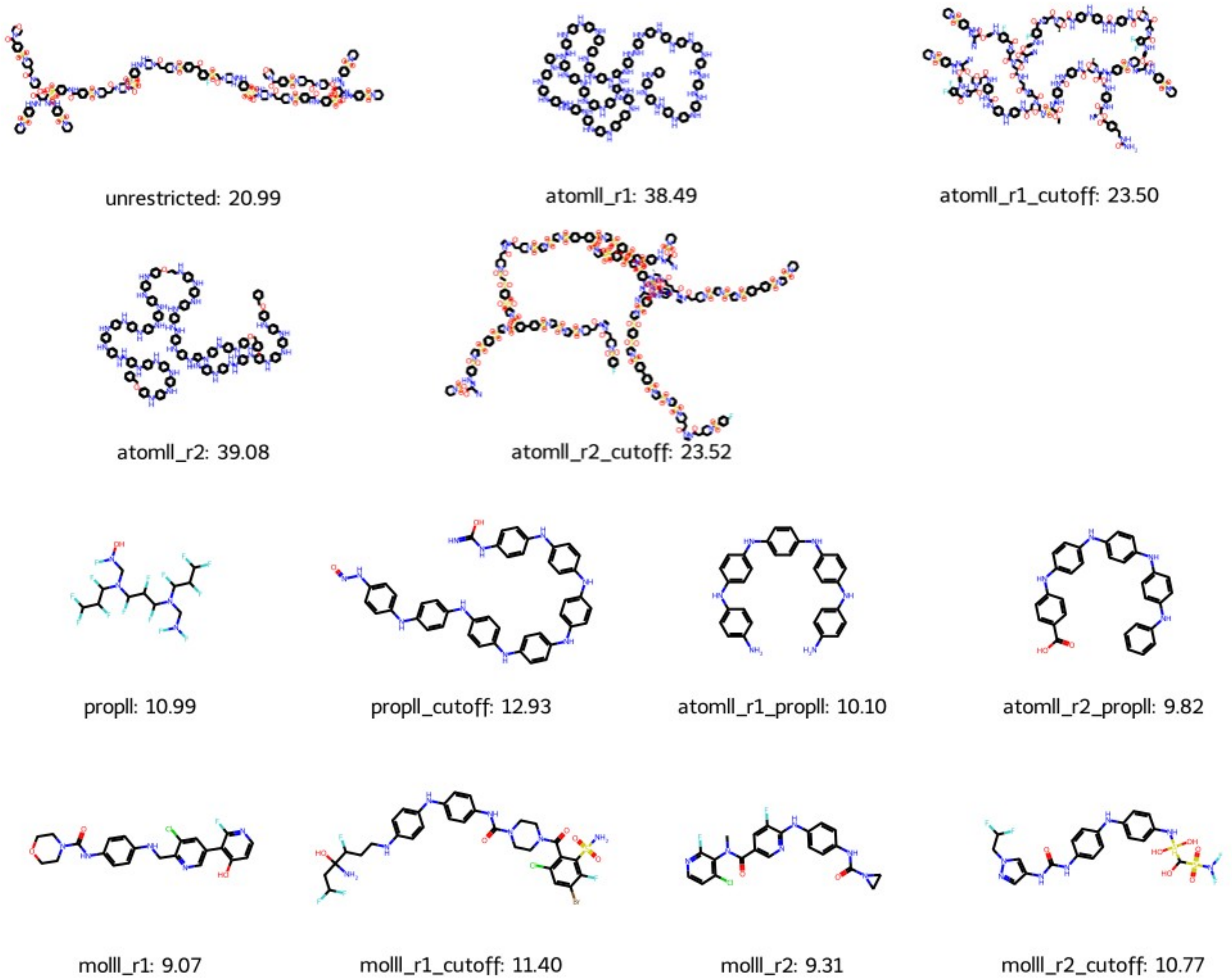


*Figure 9: Best scoring molecules obtained from using the different scoring functions for a genetic algorithm optimization. The score obtained with the QSAR model is noted under each molecule.*

A genetic algorithm was chosen as a baseline generative de novo design and molecular optimization algorithm with a propensity to generate out-of-scope molecules, and was tested with a basic QSAR model and different restraints added to the scoring function used for optimization.

Figure 9 shows the best scoring molecules obtained with the different scoring function modifiers used together with their score from the QSAR model. The unrestricted runs and runs with the AtomLL modifiers failed to constrain the molecular size of the optimization run, resulting in out-of-scope generation with large molecules with many repeats in just a few generations, as shown in Figure 10. The runs with an optimization score modified with property-based LL models seem to have generated more reasonably sized molecules, but with many repeating substructures. The property-based modification alone also has unusual substructures, such as -NF2, N(F)OH, and NN=O. The MolLL restrained optimization runs seem to be of reasonable sizes and moreover, without unusual repeats. Using the cutoff approach gave molecules with larger QSAR scores, but also molecules with more questionable substructural motifs, e.g., -C(OH)NH2 and the unusual sulfur-containing end group to the right in the moll_r2_cutoff constrained molecule. Top 8 scoring molecules from all runs can be found in the Supplementary information Figures 26 to 38.

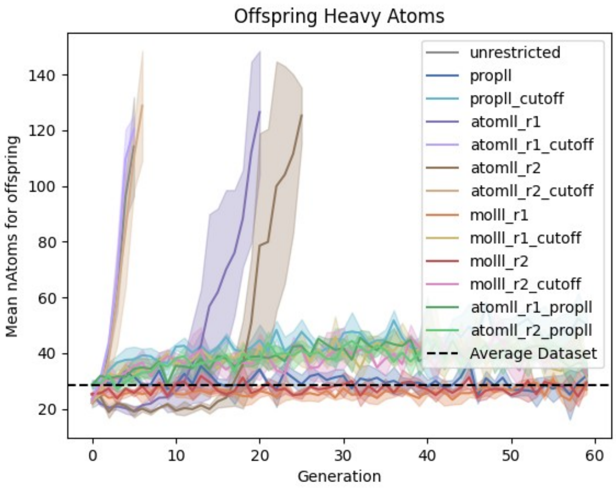


*Figure 10: Mean number of atoms for offspring for early generations of selected genetic algorithm runs. The unrestricted and AtomLL restrained scoring functions generated large molecules over the course of a few generations and had to be stopped prematurely. Curves are shown as the mean of triplicate runs with the standard deviation shown as a shaded area.*

Figure 10 and 11 show the development of the mean number of atoms in the offspring generations. As evident in Figure 10, unrestricted as well as AtomLL restrained scores failed to contain the tendency to create large molecules with larger

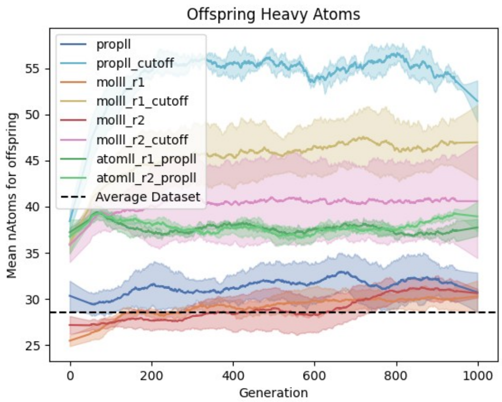


*Figure 11: Development of the mean number of atoms in offspring over the course of the genetic algorithm optimization. Curves are shown as the mean of triplicate runs with the standard deviation shown as a shaded area. The series was smoothed with a Savitzky-Golay filter with a window size of 50 and an exponent of one.*

QSAR scores and were stopped prematurely due to performance reasons. The cutoff variations had a similar performance as the unrestricted, whereas the direct inclusion of AtomLL restraints seemed to delay the tendency by a few generations.

Moreover, as evident from Figure 11, the other forms of restraints were able to constrain the development of large molecules to a different degree. Property-based as well as the MolLL models used as direct restraints kept the generations closer to the average of the train data used to compute the statistics for the models, albeit all ending with an average above the train data average. The cutoff variants of the restraints all allowed for the generation of larger molecules than their counterparts. The combination of AtomLL with the property-based restraint showed intermediate sizes, even though they contained the same property-based restraint as the property restrained run alone.

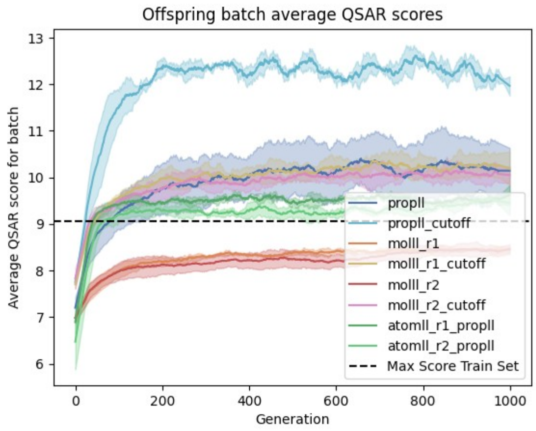


*Figure 12: Average QSAR score of offspring for the completed genetic algorithm optimization runs. Curves are shown as the mean of triplicate runs with the standard deviation shown as a shaded area. The series was smoothed with a Savitzky-Golay filter with a window size of 50 and an exponent of one. The stippled black line shows the max QSAR score for the training set.*

Figure 12 shows the development of average QSAR scores for the offspring over the course of the GA optimization for the various restraint options tested. Most of the runs end up extrapolating the QSAR scores, which could implicate that they are outside the applicability domain of the model and the optimization has found ways to exploit the QSAR scoring function. Figure 9 suggests this may be by repeating well-scoring elements. The MolLL-based restraints are kept well within the QSAR score range with a gradual increase throughout the course of the optimization. It should be noted that the standard deviation shown in the plot is between the averages of each triplicate run and not the variation of scores in each offspring generation. All the cut-off-based restraints get higher scores than their directly restrained counterparts, and the combination of AtomLL and property restraints keeps the scores closer to the max score observed for the training set.

Figure 13 shows the fraction of molecules that would be excluded based on the publicly available NIBR medicinal chemistry filters for small molecule drug discovery. Most runs end up generating molecules that would almost all be excluded. The MolLL-based model with direct integration gives the lowest fractions, around 15-25% excluded. The cutoff variants give a higher proportion of molecules in the offspring that would end up being excluded by the filters.

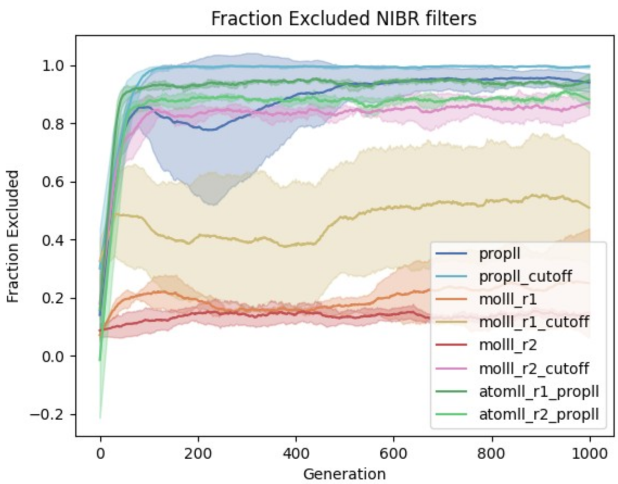


*Figure 13: Fraction of molecules that would be excluded according to NIBR filters. Curves are shown as the mean of triplicate runs with the standard deviation shown as a shaded area. The series was smoothed with a Savitzky-Golay filter with a window size of 50 and an exponent of one.*

Overall, the results suggest that the MolLL-based model with direct integration into the scoring function for optimization and generative algorithms will both ensure an adequate size, no exploitation by repeating substructures, and fewer unusual substructural motifs. This overall also leads to a lower fraction of the generated molecules which would have to be filtered away by subsequent application of medicinal chemistry rules in the form of the NIBR filters.

## Discussion

Although the log-likelihood estimations have shown success in various applications, some improvements could be envisioned. One aspect to consider is that the log-likelihood estimation is dependent on parameters such as the fingerprint key radius. Increasing the fingerprint key radius makes the keys more specific and thus less frequently observed, while also expanding the number of observations as a fingerprint with radius 2 includes the keys for radius 1 and 0. This effectively lowers the numerator and increases the denominator in Equation 1, leading to a shift towards lower log-likelihoods for molecules with increasing radius, despite being from the same dataset. Additionally, differences in log-likelihood distributions between the AtomLL and MolLL models exist. Therefore, it may be beneficial to investigate formulations that are independent of the underlying key or property definition, or find a way to normalize and scale the calculated log-likelihood estimates. Alternatively, it may be an inherent property of the approach that the more detailed the comparison and the more properties are added to an evaluation between items, the higher the likelihood that differences appear between the molecules, thereby decreasing the likelihood of them being similar.

This dependency of the outcome on the formulation and model choice is also evident in the qualitative differences observed by the two proposed models, AtomLL and MolLL. A comparison of the molecules in Figure 2 and 4 illustrates this. The low likelihood of belonging to a dataset depends on the way we compute the features of the molecules, and thus our choice and biases as an observer cannot be excluded from the calculated statistics. However, this flexibility allows us to choose a model based on the aspects we want to use to identify outliers. For instance, if we are interested in identifying unusual substructures and local atom environments, we can use the AtomLL model. Conversely, if we are also interested in the distribution and count of repeats, we can choose the MolLL model.

There is some correlation between the models, as investigated for the ChEMBL datasets using the log-likelihood models derived from the LibInvent and plotted in Figure 14. Unsurprisingly, the models with the same functional form but different settings are mostly correlated, with weaker correlation between the AtomLL and MolLL models. The property models are least correlated with the other two. This finding is somewhat surprising for the MolLL model, as it was shown to be quite efficient at balancing the genetic algorithm optimization tendency to create large molecules. Instead, it might indicate that its in-scope effect of the restraint is more likely to come from preventing exploitation via repeats of similar sub-structures, a scoring function exploit that seems common and may be inherent to the crossover algorithm of the genetic algorithm.

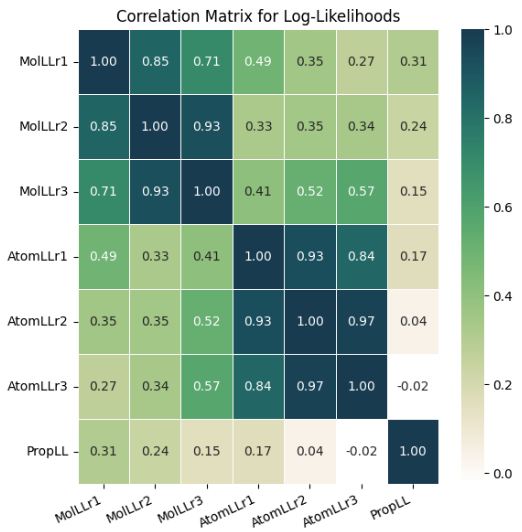


*Figure 14: Correlation matrix heatmap for the ChEMBL data log-likehoods as estimated with models derived from the LibInvent data.*

Taking the next illustrative use with class membership estimation, the performance of the class membership analysis seems good from the statistics shown in Table 2. However, the performance of this simple approach is not as good as a directly fitted model, here exemplified with a simple Ridge classifier. To compare with a dedicated class model, a RidgeClassifierCV model was tuned and trained on the same train-set, and the performance was analyzed. The features used were Morgan Fingerprints as implemented in RDKit[7] and integrated into the Scikit-Learn model via Scikit-Mol[6]. Table 3 shows that this simple multiple linear regression model obtains even better results than the unfitted models based on MolLL and AtomLL (cf. Table 2). The results from the class-membership analysis should not be interpreted as MolLL and AtomLL being good choices for classifier models, but rather be understood that MolLL and AtomLL models capture differences between molecular classes on a basis that is also relevant for biological activity. The example class membership was included as it illustrates properties of and allows us to understand the AtomLL and MolLL model better, rather than being a recommended way of doing classification models.

*Table 3: Classification metrics for the Ridge Classifier using the D2R dataset test set.*

| | Bits | Bits | Bits | Counts | Counts | Counts |
|---|---|---|---|---|---|---|
| FP Radius | 1 | 2 | 3 | 1 | 2 | 3 |
| Recall | 0.988 | 0.992 | 0.992 | 0.989 | 0.994 | 0.992 |
| Precision | 0.962 | 0.983 | 0.984 | 0.962 | 0.982 | 0.983 |
| Accuracy | 0.975 | 0.988 | 0.988 | 0.975 | 0.988 | 0.988 |

One attractive aspect of the model is that it works using only a single class, as exemplified by actives, and can also provide insights into the datasets. The inactives of the D2R dataset are more diverse, and the log-likelihood models derived from these contain chemistry that is also found in the actives, as the actives also receive reasonable log-likelihood estimates from these models. In general, the inverse is less true; the models derived from the actives assign low log-likelihood to the inactives, likely reflecting that there are specific ways to obtain active molecules, but there are many more ways to have inactive molecules.

When it comes to keeping generative algorithms in scope, the genetic algorithm was chosen as an illustrative example as it's easy to obtain molecules that are easily recognized as out-of-scope. However, other generative models have been prone to over-exploiting scoring functions[1]. One way to counteract this is to ensure that QSAR models are indeed predictive[21], or to impose restraints or filters on the scored molecules based on descriptors and molecular properties[22]. The AtomLL and MolLL can be seen as being in the latter category, but maybe focusing more on substructures than overall properties.

A caveat when interpreting the differences between the different restraint models is that no exhaustive effort has been made to tune the scoring function variants individually. It could be that with different weights in the scoring function, the results would have been better for some of the variants of the scoring function. For example, we observed undesirable molecules and molecular motifs, but also higher QSAR scores, for the cutoff variants. The cutoff had been chosen to be in the lowest percentile of scores obtained from the sample of the dataset, and it could have been set higher, which would better allow filtering out the undesirable molecules at a probable tradeoff of obtaining lower QSAR scores. This balance is, of course, entirely up to the individual designer of the desirability score for the generative algorithm and the desired outcome of the run.

The choice of Morgan Fingerprint as the basis for the models was based on previous experience with these fingerprints, but it would be easy to imagine other models based on different fingerprints. These could have other beneficial properties for outlier detection, classification, or as restraints on generative algorithms, as some fingerprints have been shown to be more beneficial in certain domains, such as for the classification of natural compounds[23], [24]

## Conclusion

In conclusion, MolLL and AtomLL offer valuable models for estimating the log-likelihood of molecular structures. Despite their simple fingerprint key count approach, they prove useful for outlier detection in datasets and for ensuring in-scope control in generative de novo design and molecular optimization. While the models can also be applied to assign class membership, they do not outperform even simple multiple regression-based classifiers. Each model highlights different aspects: the AtomLL model excels at identifying molecules with unusual singular motifs, whereas the MolLL model is better suited for in-scope control. It considers the number of key observations in the molecule, thus mitigating score exploitation in the form of repeated substructures and maintaining reasonable molecular sizes. The open-source code is readily available for download and use, facilitating analysis of custom datasets. Additionally, a few precomputed models based on a prefiltered dataset of drug-like molecules are provided for convenience.

## References


[1] P. Renz, D. Van Rompaey, J. K. Wegner, S. Hochreiter, and G. Klambauer, "On failure modes in molecule generation and optimization," *Drug Discov. Today Technol.*, vol. 32–33, pp. 55–63, Dec. 2019, doi: 10.1016/j.ddtec.2020.09.003.

[2] P. Ertl and A. Schuffenhauer, "Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions," *J. Cheminformatics*, vol. 1, no. 1, p. 8, Dec. 2009, doi: 10.1186/1758-2946-1-8.

[3] G. R. Bickerton, G. V. Paolini, J. Besnard, S. Muresan, and A. L. Hopkins, "Quantifying the chemical beauty of drugs," *Nat. Chem.*, vol. 4, no. 2, pp. 90–98, Jan. 2012, doi: 10.1038/nchem.1243.

[4] C. Steinmann and J. H. Jensen, "Using a Genetic Algorithm to Find Molecules with Good Docking Scores," 2021, doi: 10.26434/chemrxiv.13525589.v2.

[5] M. Awale, "Modifying the Synthetic Accessibility Score to Identify Undesirable Virtual Compounds," RDKit UGM 2021, okt 2021. Accessed: Apr. 02, 2024. [Online]. Available: https://github.com/rdkit/UGM_2021/blob/main/Presentations/Awale_SAScoreModifications.pdf

[6] E. J. Bjerrum *et al.*, "Scikit-Mol brings cheminformatics to Scikit-Learn," *ChemRxiv*, Dec. 2023, doi: 10.26434/chemrxiv-2023-fzqwd.

[7] "RDKit: Open source cheminformatics." Accessed: Sep. 08, 2022. [Online]. Available: http://www.rdkit.org

[8] B. Tingle *et al.*, "ZINC-22 - A Free Multi-Billion-Scale Database of Tangible Compounds for Ligand Discovery," Oct. 2022, doi: 10.26434/chemrxiv-2022-82czl.

[9] "ZINC." Accessed: Apr. 03, 2024. [Online]. Available: https://zinc.docking.org/tranches/home/

[10] "mgenerators-failure-modes/assays/processed at master · ml-jku/mgenerators-failure-modes." Accessed: Apr. 03, 2024. [Online]. Available: https://github.com/ml-jku/mgenerators-failure-modes/tree/master/assays/processed

[11] J. Sun *et al.*, "ExCAPE-DB: an integrated large scale dataset facilitating Big Data analysis in chemogenomics," *J. Cheminformatics*, vol. 9, no. 1, p. 17, Mar. 2017, doi: 10.1186/s13321-017-0203-5.

[12] J. Sun *et al.*, "ExcapeDB: An integrated large scale dataset facilitating Big Data analysis in chemogenomics." Zenodo, Nov. 29, 2016. doi: 10.5281/zenodo.173258.

[13] V. Fialková *et al.*, "LibINVENT: Reaction-based Generative Scaffold Decoration for in Silico Library Design," *J. Chem. Inf. Model.*, p. acs.jcim.1c00469, Aug. 2021, doi: 10.1021/acs.jcim.1c00469.

[14] V. Fialková *et al.*, "LibINVENT: Reaction-based Generative Scaffold Decoration for in Silico Library Design." Zenodo, Aug. 30, 2021. doi: 10.5281/zenodo.6627127.

[15] P. Virtanen *et al.*, "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nat. Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2020, doi: 10.1038/s41592-019-0686-2.

[16] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90–95, May 2007, doi: 10.1109/MCSE.2007.55.

[17] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. null, pp. 2825–2830, Nov. 2011.

[18] "mol-ga: Simple genetic algorithms for 2D molecular design." Accessed: Apr. 04, 2024. [Online]. Available: https://github.com/AustinT/mol_ga

[19] A. Schuffenhauer *et al.*, "Evolution of Novartis' Small Molecule Screening Deck Design," *J. Med. Chem.*, vol. 63, no. 23, pp. 14425–14447, Dec. 2020, doi: 10.1021/acs.jmedchem.0c01332.

[20] Abraham. Savitzky and M. J. E. Golay, “Smoothing and Differentiation of Data by Simplified Least Squares Procedures.,” *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, Jul. 1964, doi: 10.1021/ac60214a047.

[21] M. Langevin, R. Vuilleumier, and M. Bianciotto, “Explaining and avoiding failure modes in goal-directed generation of small molecules,” *J. Cheminformatics*, vol. 14, no. 1, p. 20, Apr. 2022, doi: 10.1186/s13321-022-00601-y.

[22] M. Langevin *et al.*, “Impact of Applicability Domains to Generative Artificial Intelligence,” *ACS Omega*, vol. 8, no. 25, pp. 23148–23167, Jun. 2023, doi: 10.1021/acsomega.3c00883.

[23] D. Boldini, D. Ballabio, V. Consonni, R. Todeschini, F. Grisoni, and S. A. Sieber, “Effectiveness of molecular fingerprints for exploring the chemical space of natural products,” *J. Cheminformatics*, vol. 16, no. 1, p. 35, Mar. 2024, doi: 10.1186/s13321-024-00830-3.

[24] A. Capecchi, D. Probst, and J.-L. Reymond, “One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome,” *J. Cheminformatics*, vol. 12, no. 1, p. 43, Jun. 2020, doi: 10.1186/s13321-020-00445-4.