EvoFlopa: Evolutionary Fast Lead OPtimization Algorithm

Preprint, compiled January 4, 2025

Tomáš Jelínek[1*] **and Marek Tobiáš, Anna Gregušová, Aneta Tranová, Tomáš Preisler**[2]

[1]Main Author, Charles University, Faculty of Science
[2]TrmD project collaborators, Charles University, Faculty of Science

Abstract

This study introduces an iterative molecular optimization framework that integrates modified STONED algorithm [1] for de novo molecular design with UniDock for binding affinity assessment, and RDKit for property calculation. The framework uses a weighted loss function, combining docking scores, synthetic accessibility (SA), and quantitative estimate of drug-likeness (QED), to guide molecular exploration towards molecules with improved binding affinity and favorable physicochemical properties. By leveraging multithreading, CUDA and Docker containerization or Conda environment, the approach facilitates efficient exploration of chemical space and is easy to deploy on various computational platforms. We demonstrate the effectiveness of the framework by identifying novel compounds with favorable binding affinities to a TrmD target, a promising target for the development of novel antibiotics. All of the code has been made available on GitHub: https://github.com/Desperadus/EvoFLOPA.

1 Introduction

The sheer scale of the synthesizable drug-like chemical universe, estimated to encompass approximately 10^{27} molecules, makes exhaustive exploration impossible; thus, we employ a targeted approach focusing on structure-based drug design (SBDD) guided by evolutionary algorithms to efficiently navigate this vast space.

We contrasted our approach with existing methods, such as the AutoGrow4 [2] program, which also uses an evolutionary algorithm. However, our approach differs by using the SELFIE string representation of molecules, allowing us to directly manipulate molecular structures without reliance on predefined fragments or reaction libraries. This representation enables the exploration of a wider range of chemical space. Our approach incorporates a multi-objective scoring function that combines docking scores, synthetic accessibility (SA), and quantitative estimates of drug-likeness (QED) to evaluate molecule fitness. We further improved the efficiency by leveraging multithreading and the computational capabilities of GPUs.

While deep learning methods, such as those utilizing recurrent neural networks (RNNs), Graph Neural Networks (GNNs), Variational AutoEncoders (VAEs) or transformers, are being used to explore molecular design, they typically require training on large datasets of known active molecules. Subbranch of deep learning methods are models using reinforcement learning, such as REINVENT4 [3]. They have been used to generate novel molecules with desired properties. However, these methods are computationally expensive and very often use SMILES representation of molecules, which is not as sturdy as SELFIE representation, moreover REINVENT4 uses OpenEye toolkit, which is not open-source.

Our approach has the ability to generate diverse molecules using one or few starting molecules. To evaluate the performance, our method was applied to the discovery of novel molecules targeting the TrmD protein, a promising target for the development of novel antibiotics. The results demonstrate that our strategy is effective for generating drug-like molecules with improved binding affinity and other desired properties.

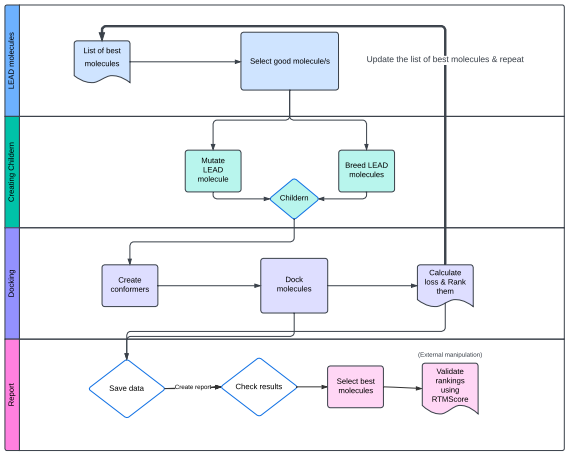

Figure 1: Pipeline overview of EvoFLOPA.

2 Methodology of EvoFLOPA

2.1 *Molecular Representation: SELFIES*

Our method, EvoFLOPA, utilizes Simplified Molecular-Input Line-Entry System (SELFIES) strings for molecular representation [4]. SELFIES are a robust, string-based encoding of molecules that allow for direct manipulation of molecular graphs via string operations, ensuring syntactic validity. This representation enables the generation of diverse and chemically valid molecules throughout the evolutionary optimization process.

2.2 *Docking and Affinity Assessment: UniDock*

EvoFLOPA leverages UniDock for assessing binding affinity. UniDock is a highly optimized docking program that uses CUDA acceleration to greatly improve speed and efficiency.

*correspondence: tomgolf.jelinek@gmail.com

2.3 Evolutionary Optimization: Mutation and Breeding

2.3.1 Mutation

Our algorithm introduces a mutation step that modifies the SELFIE string representation of the parent molecule. Three types of mutations are implemented: atom addition, atom deletion, and atom replacement. These mutations are performed with a 33% probability each. This is done so that the SELFIE string is tokenized and then the mutation is done on the tokens. As SELFIES are surjective onto the set of all possible molecules - one can in theory explore the whole chemical space using these mutations.

2.3.2 Breeding

EvoFLOPA implements breeding by taking 2 parent molecules, computing the shortest edit path between them using Levenshtein distance and changing one molecule to other by doing these edits - this is not done the same way as in the original STONED paper, where they just changed tokens of the SELFIE string without doing any alignment. This allows for the generation of a new molecule that contains features from both parent molecules. This process, coupled with mutations, allows the algorithm to explore the chemical space effectively.

2.4 Docking and Loss Function

After the mutation and breeding step, conformers are generated and all of the molecules are docked using UniDock. After docking, each molecule is scored using a weighted loss function. The loss function L is calculated as:

$$L = w_{SA} \cdot (1 - \frac{SA}{10}) + w_{QED} \cdot QED - w_{Dock} \cdot DockingScore$$

where SA is the synthetic accessibility score, QED is the quantitative estimate of drug-likeness, and $DockingScore$ is the UniDock docking score. The weights, w_{SA}, w_{QED}, and w_{Dock} are user-defined parameters that control the contribution of each term to the overall loss. This multi-objective scoring ensures that the algorithm not only identifies molecules with high binding affinity but also those that are synthetically feasible and drug-like. The algorithm tries to maximize the loss function (I have just realised that more appropiate name would be fitness function, but who cares) - the higher the loss, the better the molecule.

2.5 Iterative Optimization

The optimization process proceeds iteratively. Molecules from each generation are first ranked based on their loss scores and appended to list of best molecules, then the top n molecules are selected from that list for seeding the next generation. These molecules are not directly selected; instead, their loss scores are converted to probabilities through a softmax function, defined as:

$$P(x_i) = \frac{e^{x_i/T}}{\sum_{j=1}^{n} e^{x_j/T}}$$

where x_i represents the loss score of the i-th molecule, n is the number of top molecules under consideration, and T is the temperature parameter. This temperature controls the exploration-exploitation balance: a lower temperature biases selection towards molecules with the highest loss scores, whereas a higher temperature increases the probability of selecting lower-scoring molecules. With these probabilities, the next generation is seed for mutation and/or breeding.

3 DESIGNING TRMD INHIBITOR

TrmD is a tRNA methyltransferase that catalyzes the methylation of guanine at position 37 (m^1G37) in tRNA. This modification is essential for maintaining proper tRNA structure and function, directly influencing the efficiency of translation and reading frame maintenance during protein synthesis. TrmD is highly conserved across various species of bacteria reflecting its critical role in cellular processes. Eucariotic cells do not have TrmD (Trm5 has the orthological function), making it an attractive target for the development of novel antibiotics with minimal off-target effects.

For the TrmD target, we have chosen the PDB crystal structure 4YVG from *H. influenzae*. This structure is monomer with a resolution of 1.55 Å and contains a bound ligand, S-AdenosylMethionine (SAM). The ligand and solvent was removed, and the protein was prepared for docking using the UniDock program.

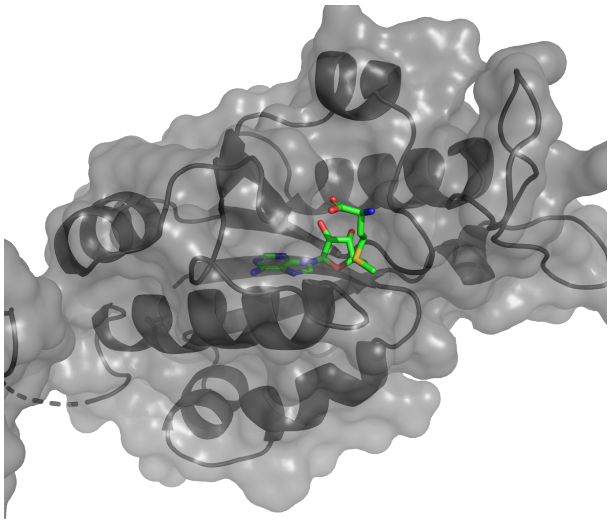

Figure 2: 4YVG TrmD protein structure with bound SAM ligand.

Wilkinson et al. [5] synthesized and tested a variety of nicotinamide and azaindole with the aim to inhibit the TrmD protein. Their molecules such as *Compound 23* have reached nanomolar (9 nM) affinities. Therefore we have selected *Compounds 21, 23, 35, 36, 37* which have displayed great binding affinity as a starting points for our EvoFLOPA optimization.

3.1 Results

EvoFlopa was run on AMD Ryzen 5 5600x (12) CPU alongside with NVIDIA GeForce RTX 4070Ti SUPER GPU. The algorithm was run for circa 500 iterations and took about 1 hour to finish. Further another one was with additional compound (the best one found in the first run) and was run for 1000 iterations and took again about 2 hour to finish.

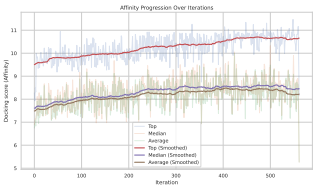

Figure 3: Bidning affinity progression of the run no.1

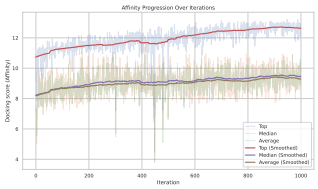

Figure 4: Bidning affinity progression of the run no.2

By running the *gen_report.py* script, one can see the progression of the EvoFLOPA algorithm as well get the best molecules found in the run.

We have named the best molecule found by EvoFLOPA *Compound Y*.

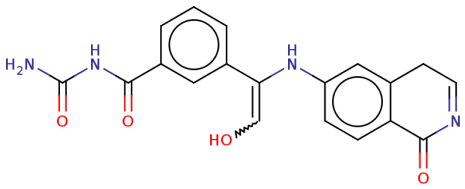

Figure 5: Compound Y found by EvoFLOPA. Having: **UniDock energy**: -10.626 kcal/mol, **RTMScore**: 40.704, **SA**: 3.01, **QED**: 0.617

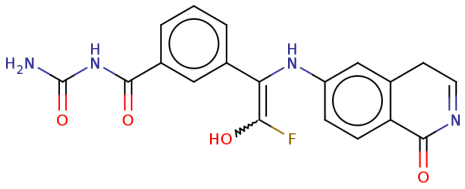

Figure 7: Compound YF found by EvoFLOPA. Having: **UniDock energy**: -10.437 kcal/mol, **RTMScore**: 45.427, **SA**: 3.05, **QED**: 0.603

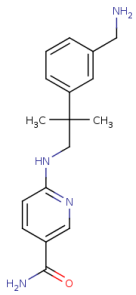

Figure 6: Compound 23 from Wilkinson et al. [5]. Having: **UniDock energy**: -7.322 kcal/mol, **RTMScore**: 32.792, **SA**: 2.30, **QED**: 0.760

3.1.1 RTMScoring

While Vina/Vinardo scoring function were is used for pose scoring, it can be limited. We used RTMScore [6], a knowledge-based deep-learning scoring function, for rescoring and verifying binding poses. Unlike Vina's empirical approach, RTMScore learns from experimental protein-ligand complexes, calculating likelihoods of observed residue-atom distances. This leads to improved accuracy and generalization, outperforming Vina in both binding pose prediction and virtual screening. RTMScore also provides valuable information about specific residue-atom contributions to binding, offering insights not available from Vina alone.

The original repository of RTMScore is provided with dependency issues, so we have created a fork that provides a Dockerfile and Conda enviroment for easy installation and usage. It can be found at: https://github.com/Desperadus/RTMScore

The *gen_report.py* script also creates .sdf file of best *n* scoring molecules outputed by EvoFLOPA, which can be then used as input for the rtmscoring script.

This rescoring using RTMScore has enabled us to gain more insight into favorable chain additions to the molecule. For example RMTScore highly preferes the addition of a fluorine atom to the part of molecule where it folds in the pocket to fill it more. As oposed to UniDock with VINArdo scoring function - EvoFLOPA has given this molecule only the 13th spot (We denote this molecule as *Compound YF*).

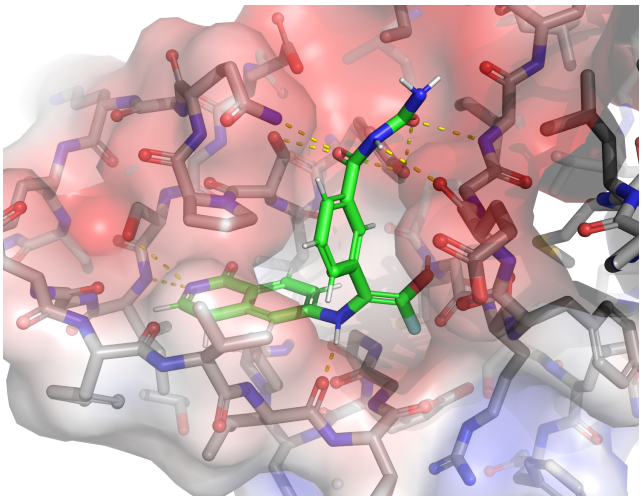

Figure 8: Compound YF in the TrmD protein pocket.

Furthermore, similarly the RTMScore has shown that addition of OH group to the *Compound Y* might be favorable for the binding as well.

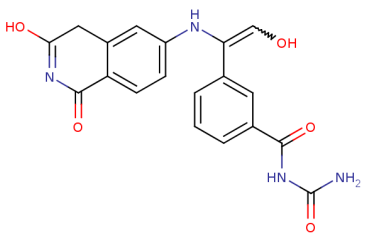

Figure 9: Compound YOH found by EvoFLOPA. Having: **UniDock energy**: -10.429 kcal/mol, **RTMScore**: 42.160, **SA**: 3.06, **QED**: 0.512

However, UniDock completely fails to dock Compound YFOH (Compound Y with added F and OH group) into the TrmD protein pocket.

Therefore we turned to SwissDock, which has successfully docked the Compound YFOH into the TrmD protein pocket. Then we have redocked it using UniDock and obtained score of -10.922 kcal/mol and RTMScore of 42.04.

4 CONCLUSIONS

We introduced EvoFLOPA, an evolutionary algorithm framework that efficiently explores chemical space using SELFIES, UniDock, and RDKit. By optimizing a multi-objective fitness function combining docking scores, synthetic accessibility, and drug-likeness, the algorithm identifies molecules with favorable binding and physicochemical properties. Our results demonstrate the discovery of novel TrmD inhibitors which appear to have the best in class binding affinity to TrmD in the know literature. All code is publicly available at https://github.com/Desperadus/EvoFLOPA.

Future Work

The biggest problem of EvoFLOPA is the stochastic nature of the mutation and breeding process, which also greatly depends on the seed molecules. This can cause problems such as the algorithm getting stuck in local minima or taking long time to converge. However, the SELFIES data generated by EvoFLOPA alongside with their binding affinities can be used to finetune a deep learning model generating SELFIES by using reinforcement learning. Resulting model could be then used further to generate new molecules with high binding affinity.

REFERENCES

[1] AkshatKumar Nigam, Robert Pollice, Mario Krenn, Gabriel dos Passos Gomes, and Alan Aspuru-Guzik. Beyond generative models: Superfast traversal, optimization, novelty, exploration and discovery (stoned) algorithm for molecules using selfies. *ChemRxiv*, January 2021. doi: 10.26434/chemrxiv.13383266.v2. URL http://dx.doi.org/10.26434/chemrxiv.13383266.v2.

[2] Jacob O. Spiegel and Jacob D. Durrant. Autogrow4: an open-source genetic algorithm for de novo drug design and lead optimization. *Journal of Cheminformatics*, 12(1), April 2020. ISSN 1758-2946. doi: 10.1186/s13321-020-00429-4. URL http://dx.doi.org/10.1186/s13321-020-00429-4.

[3] Hannes H. Loeffler, Jiazhen He, Alessandro Tibo, Jon Paul Janet, Alexey Voronov, Lewis H. Mervin, and Ola Engkvist. Reinvent 4: Modern ai-driven generative molecule design. *Journal of Cheminformatics*, 16(1), February 2024. ISSN 1758-2946. doi: 10.1186/s13321-024-00812-5. URL http://dx.doi.org/10.1186/s13321-024-00812-5.

[4] Mario Krenn, Florian Hase, AkshatKumar Nigam, Pascal Friederich, and Alán Aspuru-Guzik. Self-referencing embedded strings (selfies): A 100 2019. doi: 10.48550/ARXIV.1905.13741. URL https://arxiv.org/abs/1905.13741.

[5] Andrew J. Wilkinson, Nicola Ooi, Jonathan Finlayson, Victoria E. Lee, David Lyth, Kathryn S. Maskew, Rebecca Newman, David Orr, Keith Ansell, Kristian Birchall, Peter Canning, Peter Coombs, Lucia Fusani, Ed McIver, João Pisco, Philip M. Ireland, Christopher Jenkins, Isobel H. Norville, Stephanie J. Southern, Richard Cowan, Gareth Hall, Catherine Kettleborough, Victoria J. Savage, and Ian R. Cooper. Evaluating the druggability of trmd, a potential antibacterial target, through design and microbiological profiling of a series of potent trmd inhibitors. *Bioorganic; Medicinal Chemistry Letters*, 90:129331, June 2023. ISSN 0960-894X. doi: 10.1016/j.bmcl.2023.129331. URL http://dx.doi.org/10.1016/j.bmcl.2023.129331.

[6] Chao Shen, Xujun Zhang, Yafeng Deng, Junbo Gao, Dong Wang, Lei Xu, Peichen Pan, Tingjun Hou, and Yu Kang. Boosting protein–ligand binding pose prediction and virtual screening based on residue–atom distance likelihood potential and graph transformer. *Journal of Medicinal Chemistry*, 65(15):10691–10706, August 2022. ISSN 1520-4804. doi: 10.1021/acs.jmedchem.2c00991. URL http://dx.doi.org/10.1021/acs.jmedchem.2c00991.

APPENDIX

4.1 Run 1

Parameters used:

Parameter	Value
receptor	examples/TrmD/raw_4yvg.pdbqt
ligand	null
ligands	Compound*.sdf
scoring_function	vinardo
center_x	45.0
center_y	5.0
center_z	10.0
size_x	22.5
size_y	22.5
size_z	22.5
max_step	40
exhaustiveness	384
num_iterations	1000
num_confs	3
num_modes	3
experiment_name	experiments/main_run_1000_iter
config	config.json
batch_size	512
num_variants	32
docking_threads	3
verbose	false
top_n_history	128
temperature	0.8
breed	true
breeding_prob	0.3
min_allowed_cycle_size	3
max_allowed_cycle_size	11
seed	42

Note that the Run 1 was killed forcibly after circa 500 iterations.

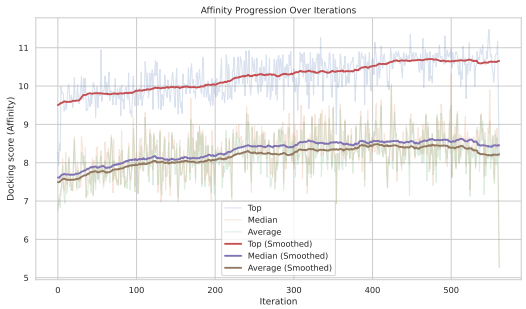

Figure 10: Bidning affinity progression of the run no.1

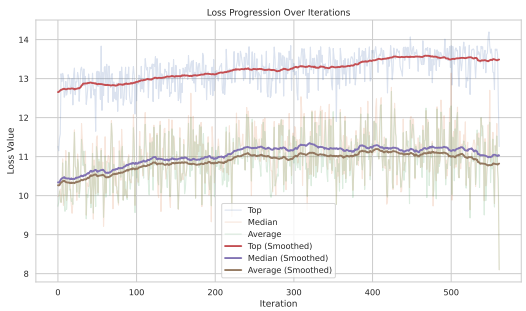

Figure 11: Loss progression of the run no.1

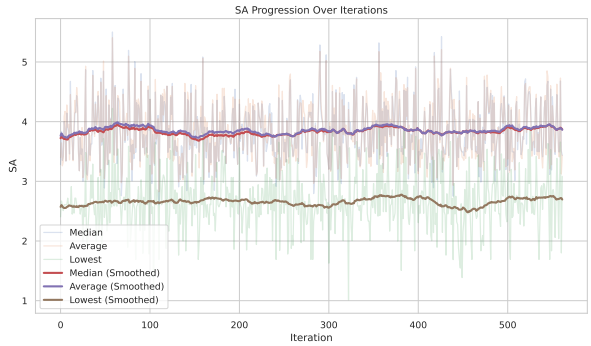

Figure 12: SA score progression of the run no.1

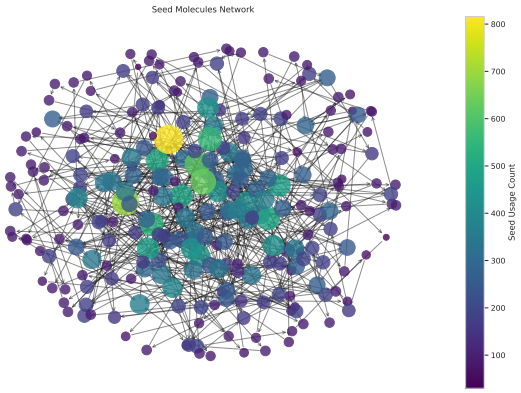

Figure 13: Network of molecules that were used as seed and their usage in the breeding process. Color by the amount molecule has been used as seed.

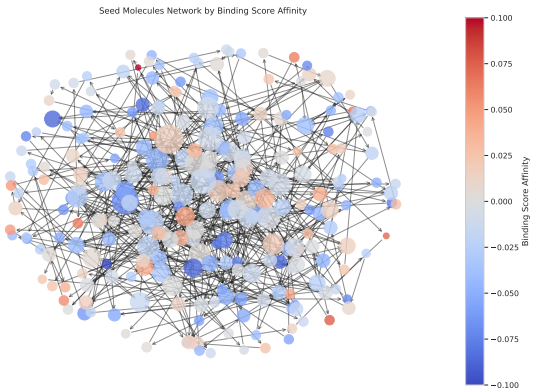

Figure 14: Network of molecules that were used as seed and their usage in the breeding process. Color by the affinity of the molecule.

4.2 Run 2

Note that in run 2, the best molecule from run 1 was added to the starting molecules. Parameters used:

Parameter	Value
receptor	examples/TrmD/raw_4yvg.pdbqt
ligand	null
ligands	Compound*.sdf
scoring_function	vinardo
center_x	45.0
center_y	5.0
center_z	10.0
size_x	22.5
size_y	22.5
size_z	22.5
max_step	40
exhaustiveness	384
num_iterations	1000
num_confs	3
num_modes	3
experiment_name	experiments/main_run_1000_iter2
config	config.json
batch_size	512
num_variants	32
docking_threads	3
verbose	false
top_n_history	128
temperature	0.8
breed	true
breeding_prob	0.3
min_allowed_cycle_size	3
max_allowed_cycle_size	11
seed	42

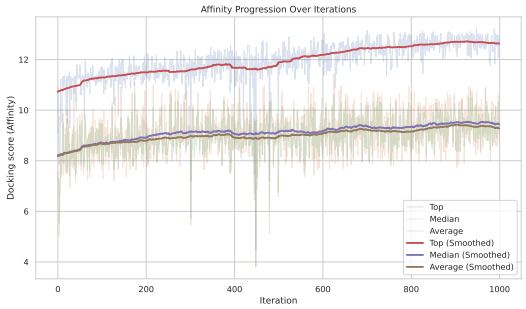

Figure 15: Bidning affinity progression of the run no.1

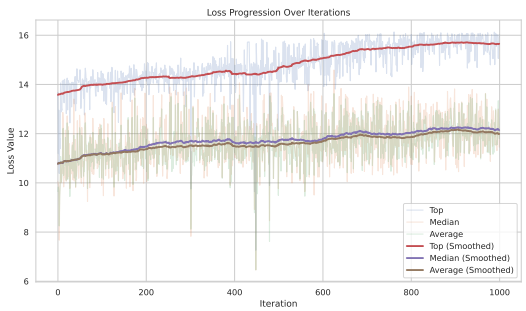

Figure 16: Loss progression of the run no.1

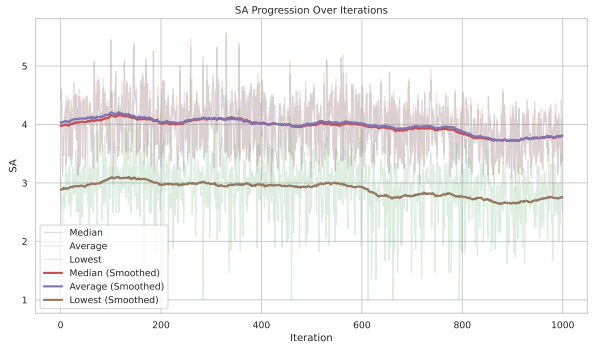

Figure 17: SA score progression of the run no.1

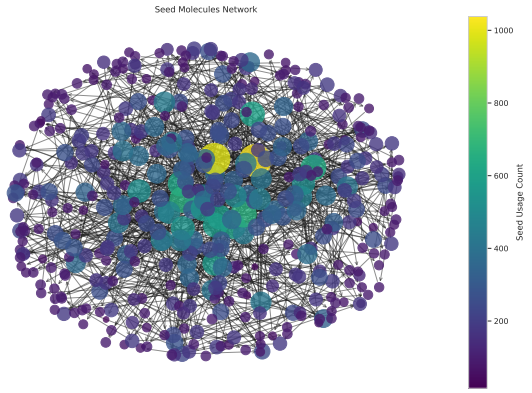

Figure 18: Network of molecules that were used as seed and their usage in the breeding process. Color by the amount molecule has been used as seed.

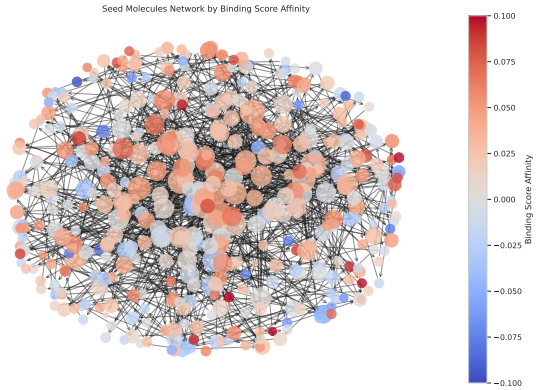

Figure 19: Network of molecules that were used as seed and their usage in the breeding process. Color by the affinity of the molecule.