StreaMD: the toolkit for high-throughput molecular dynamics simulations

Aleksandra Ivanova[1], Olena Mokshyna[1,2], Pavel Polishchuk[1*]

[1] Institute of Molecular and Translational Medicine, Faculty of Medicine and Dentistry, Palacky University, Hnevotinska 5, 77900 Olomouc, Czech Republic

[2] Institute of Organic Chemistry and Biochemistry of the Czech Academy of Sciences, Flemingovo náměstí 542/2, 160 00 Praha 6, Czech Republic

pavlo.polishchuk@upol.cz

Abstract

Molecular dynamics simulations serve as a prevalent approach for investigating the dynamic behaviour of proteins and protein-ligand complexes. Due to its versatility and speed, GROMACS stands out as a commonly utilized software platform for executing molecular dynamics simulations. However, its effective utilization requires substantial expertise in configuring, executing, and interpreting molecular dynamics trajectories. Existing automation tools are constrained in their capability to conduct simulations for large sets of compounds with minimal user intervention, or in their ability to distribute simulations across multiple servers. To address these challenges, we developed a Python module that streamlines all phases of molecular dynamics simulations, encompassing preparation, execution, and analysis. This module minimizes the required knowledge for users engaging in molecular dynamics simulations and can efficiently operate across multiple servers within a network or a cluster. Notably, the tool not only automates trajectory simulation but also facilitates the computation of free binding energies for protein-ligand complexes and generates interaction fingerprints across the trajectory. Our study demonstrated the applicability of this tool on several benchmark datasets. Additionally, we provided recommendations for end-users to effectively utilize the tool.

Keywords: molecular dynamics, high-throughput molecular dynamics, distributed simulations, GROMACS

Introduction

Molecular dynamics (MD) simulations and the computation of binding free energies represent pivotal methodologies within computational chemistry and molecular biology[1-3]. MD simulations facilitate the exploration of atomic and molecular motion and study of intermolecular interactions. Concurrently, the calculation of binding free energies associated with ligand-protein interactions can unveil the most plausible binding modes by virtue of ranking docking poses[4-6]. Moreover, it enables the prioritization of compounds for subsequent experimental evaluation by ranking ligands[4, 7]. This capacity to discern binding affinities and interactions has become increasingly pivotal in contemporary structure-based virtual screening pipelines, owing to the expanding availability of high-performance computing resources, thereby establishing the calculation of binding free energies as an integral component of such workflows.

The setup of MD simulations and the computation of binding free energies demands a certain level of expertise and knowledge. This process can be susceptible to errors (e.g. setting up force field, solvent box, simulation parameters, etc) when executed manually, especially when dealing with multiple ligands and complexes. Structure preparation necessitates a series of steps, each requiring careful parameter selection to yield valid results. Consequently, the automation of these intricate procedures and the development of simplified, user-friendly pipelines for MD simulations and free energy calculations are imperative to facilitate structure-based virtual

screening pipelines and enable the easy assessment of hundreds or thousands of ligands within a single screening campaign.

Several endeavors have been made to streamline MD protocols for end-users, thereby reducing the demand for specialized knowledge in the domain of molecular simulations. We are not going to address here multiple existing in-house solutions to MD automation due to their inaccessibility to the wider scientific community. Among accessible solutions, OpenMM, for instance, provides a versatile framework for constructing customized pipelines for MD simulations[8]. Building upon OpenMM, the OpenMMDL tool (available at https://github.com/wolberlab/OpenMMDL) has been designed to simplify the preparation of protein and ligand structures for MD simulations. It offers a web-based interface to generate a set of scripts using input files, facilitating the execution of MD simulations. Additionally, tools like HTMD[9] and ACEMD[10] enable the creation of customized pipelines and the execution of MD simulations on single servers and clusters. However, these tools require the development of tailored pipelines suitable for processing multiple protein-ligand complexes in a single execution. Galaxy is the data analysis platform, which incorporates multiple tools (including MD) and provides a web-based interface to execute MD simulations within a distributed environment[11]. A notable advantage lies in the ability to perform MD simulations involving multiple ligands bound to the same protein target through a straightforward process. However, this necessitates the installation and configuration of the tool on a cluster. Other difficulties may arise with cofactor-dependent system simulations or automatic continuation of interrupted runs since the default workflows do not support such functionalities. Also the tool does not support so far Gaussian and MCPB.py parametrization. Recent developments include Uni-GBSA[12] and ChemFlow[13], both primarily focused on the calculation of binding free energies using the MM-GBSA/PBSA approaches and the implementation of simplified, user-friendly pipelines. While Uni-GBSA supports not only the calculation of binding free energies but also conventional MD simulations of proteins or protein-ligand complexes, it is not inherently designed for high-throughput simulations, requiring users to establish their own pipelines for execution in a distributed environment. On the other hand, ChemFlow can be executed on distributed systems operating under SLURM or PBS schedulers, but its ScoreFlow module is primarily geared towards the re-scoring of docking poses using the MM-GBSA/PBSA approaches and is not very suitable for conventional MD simulations. Hence, an evident gap persists in the availability of tools that can automate the most common MD simulations and are amenable to execution on distributed systems without necessitating specialized knowledge in their operation.

We have established an automated pipeline designed to facilitate explicit-solvent MD simulations across various systems, including proteins, protein-cofactors, protein-ligand complexes, and protein-ligand-cofactors systems. Notably, our pipeline distinguishes itself by accommodating simulations involving cofactors, which are often intrinsic components of proteins and are of critical significance for obtaining accurate simulation results. The key feature of this pipeline lies in its comprehensive automation, encompassing all stages of the simulation workflow, commencing from system preparation and extending through to the execution of production simulations.

It is noteworthy that our developed pipeline seamlessly supports systems necessitating customized atom types and force fields, such as cases involving specific metal ions within a binding site or ligands containing boron atoms. Importantly, this support is integrated and does not impose additional burdens on the user. Furthermore, our tool permits the easy continuation or extension of simulations as required.

Additionally, we have integrated MD simulation pipelines with the computation of binding free energies utilizing the MM-GBSA/PBSA methodology and the analysis of protein-ligand contacts. These simulations and calculations can be executed on both single servers and distributed systems. The incorporation of distributed systems has been achieved through the utilization of the Dask library, which removes the need for a dedicated scheduler and enables operation across a network of computers. This development empowers the performance of high-throughput MD simulations and the calculation of binding free energies for a substantial number of ligands, all achieved with minimal user efforts.

Implementation

The module has been implemented using Python 3 and is designed to operate within the UNIX operating system environment. Illustrated in Figure 1 the general workflow delineates the operational sequence. Users are required to supply a prepared protein structure in PDB format, ensuring its completeness by addressing any missing residues and side chains, while also ensuring protonation and, in particular, explicitly setting histidine protonation states. Furthermore, users have the option to submit one or more ligands and/or cofactors in MOL or SDF formats, with coordinates aligned with those of the submitted protein.

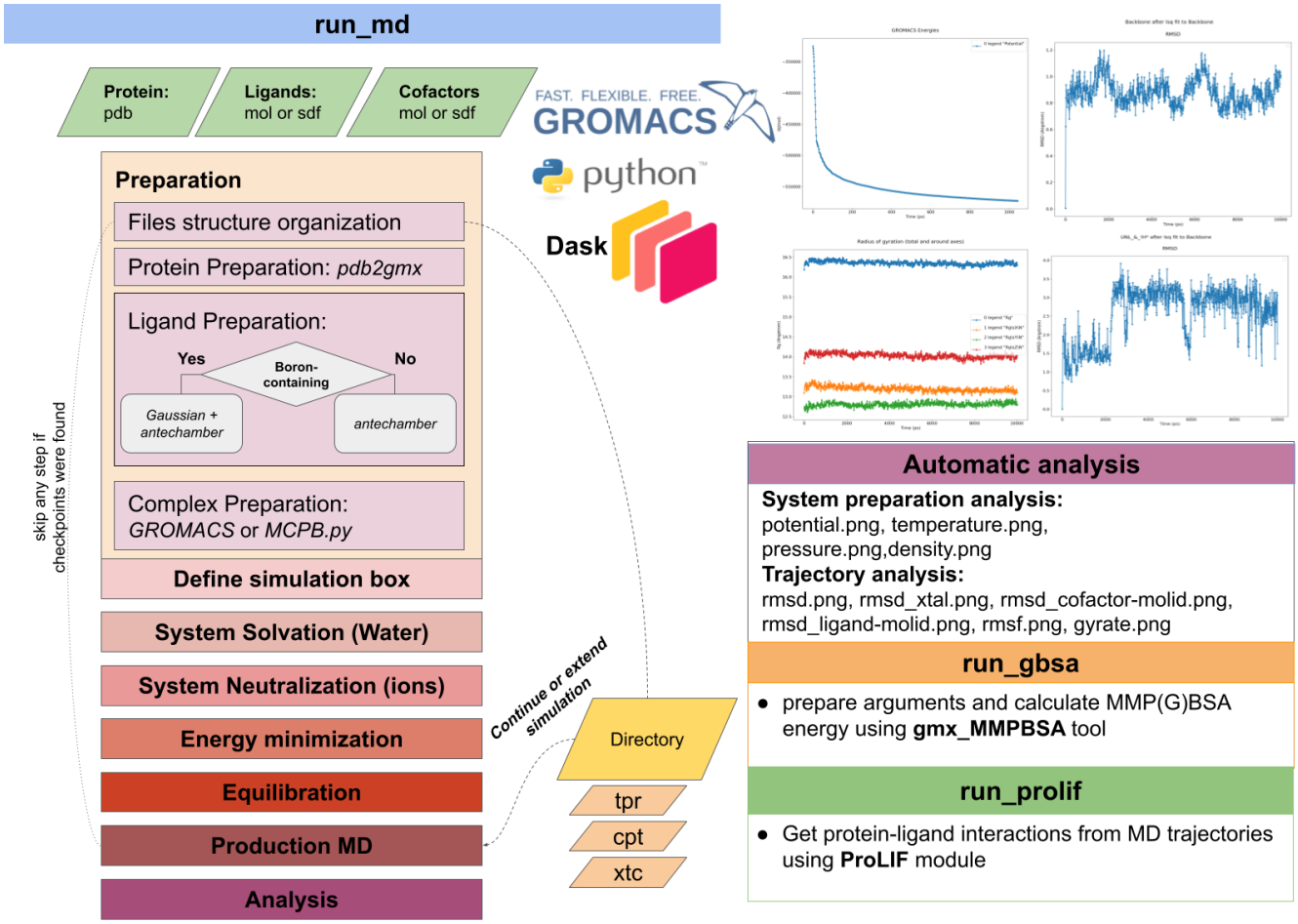


Figure 1. Overview of the StreaMD pipeline.

The tool relies on a strict hierarchy of files and directories within the user specified output directory – the *root* directory. This directory structure will be created automatically upon running of corresponding simulations. All MD files will be stored in the *root/md_files* directory and all log files will be stored in the *root* directory directly. Files created during protein, ligand and cofactors preparation are stored in *root/md_files/md_preparation/{protein, ligands, cofactors}*. Complex preparation and production run MD files are stored into *root/md_files/md_run/${protein-name}_${ligand-id}/*. In the case if such directories already exist, the tool will search for checkpoint files to skip previously completed steps and will continue an interrupted run.

StreaMD offers two operational modes: conducting simulations and extending existing simulations. In the latter mode, users are required to submit either a directory containing the preceding run generated by StreaMD or

external files in *tpr*, *cpt*, and *xtc* formats. Additionally, users must specify the desired extension duration for the simulation in nanoseconds.

Ligand and complex preparation stages, as well as MD simulation and subsequent analysis, are conducted individually for each submitted system (complex). These tasks, with the exception of MD production simulations, are parallelized based on CPU core allocation. Meanwhile, MD production simulations are parallelized also on a per-node basis, with the user retaining the option to restrict the maximum number of CPU cores utilized per node. The parallel execution is facilitated through the utilization of the Dask library, which has previously demonstrated efficacy in our EasyDock tool for distributed docking[14]. Dask, a Python library tailored for parallel and distributed computing, supports execution across various clusters or a network of servers via SSH connections. To activate parallel processing, users are required to submit a text file containing the node addresses to be utilized by a Dask SSH cluster.

Protein Preparation

Before the start of simulations, a user should prepare the protein structure:

1. Complete missing residues and reconstruct missing loops
2. Resolve alternative residue locations
3. Remove co-crystallizated ligands and water molecules, if any
4. Protonate the protein at a chosen pH value
5. Check protonation states of amino acids, in particular for histidines to put proper aliases HIE, HID or HIP (otherwise protonation may be changed during MD preparation stage)

StreaMD provides automatic processing of the submitted protein structure by executing the command *gmx pdb2gmx*, which reads a pdb file, reassign hydrogens according to amino acid residue names and writes coordinates and a topology in GROMACS format. By default, the tool employs TIP3P water model and AMBER99SB-ILDN forcefield[15]. If checkpoint files *${protein-name}.gro* and *topol.top* already exist in the working directory (*root/md_files/md_preparation/protein/*) the preparation step will be skipped.

Ligand/Cofactor Preparation

If a user supplies 3D structures of ligands or cofactors, the tool initiates a molecular preparation step, generating *mol2* files containing coordinates and atomic charges, along with corresponding *${ligand-id}.itp* files encompassing force field constants and *posre_${ligand-id}.itp* files specifying restraints for equilibration. Molecule preparation starts with addition of hydrogens according to the charged states of atoms and the total formal charge. For molecules incorporating boron atoms lacking force field parameters, a special workflow for geometry optimization and electrostatic potential computation was implemented, utilizing Gaussian software (http://signe.teokem.lu.se/ulf/Methods/resp.html, https://www.x-mol.com/groups/Dong/news/816). Gaussian output files are transformed into *mol2* format with calculated RESP charges by the antechamber tool. To employ the Gaussian parameterization approach, users are required to submit the path to the Gaussian executable file and an activation string for the Gaussian module (if computations are to be conducted on a cluster). For other molecules, antechamber is utilized to compute bcc charges and generate *mol2* files.

The generated *mol2* files serve as input for Amber parmchk2, facilitating the creation of force field modification files (*frcmod*) containing requisite force field parameters. Subsequently, the LEaP program (*tleap*) is employed to generate AMBER topology and coordinate files, which are subsequently converted into GROMACS topology and coordinate files using ParmED. Finally, *gmx genrestr* is utilized to generate position restraints for each prepared molecule.

Any encountered issues during the preparation of individual ligands will not impact others, as unprepared ligands are simply omitted from the process. Conversely, any complication arising with a cofactor will halt program execution, as a system cannot undergo simulation without all cofactors present.

The presence of *${ligand-id}.itp* and *posre_${ligand-id}.itp* files in the corresponding directory will trigger the bypassing of the molecule preparation step. If *mol2* files exist without accompanying *itp* files, the preparation workflow exclusively skips the *mol2* generation step, including the Gaussian-based process for molecules containing boron atoms. This may also work for molecules containing other atoms, but we did not investigate this possibility.

Complex preparation

Following the prior steps, all prepared files including those for the protein, ligands, and cofactors are seamlessly merged into corresponding complex.gro and topology files, which are then stored within a designated *md_run* directory. The solvation process is executed by *gmx solvate*, configuring a cubic box with a 1 nm distance between the solute and the box. To neutralize the system, Na+ and Cl- ions are introduced via *gmx genion*. A checkpoint file, *solv_ions.gro*, is generated accordingly. In cases where this file exists, both the solvation and neutralization steps are automatically skipped.

For protein-ligand complexes involving metal ions, a distinct preparation protocol utilizing the MCPB.py module[16] was implemented. Application of the MCPB.py parametrization necessitates user provision of metal residue names, alongside specification of the Gaussian executable file path and the Gaussian module activation string (particularly for cluster-based computations).

System minimization proceeds until the maximum force value reaches 1000.0 kJ/mol/nm or less, but not exceeding 50000 steps. Following this, consecutive 1000 ps NVT and NPT equilibrations are executed (the time duration can be customized by a user). Minimization and equilibration phases yield respective system analysis files, such as *potential.png* detailing potential energy variations during minimization, and *temperature.xvg*, *pressure.xvg*, and *density.xvg* from the equilibration phase. These files serve to visually assess system stability and facilitate further analysis. Throughout these procedures, the tool generates checkpoint files to expedite subsequent runs by skipping completed minimization, NVT, or NPT equilibration steps.

MD simulations

Users have the option to define the simulation duration in nanoseconds, with a default value of 1 ns, as this is a minimum reasonable trajectory length to perform some analysis and identify issues. The outcome of this phase comprises *md_out.tpr* (topology), *md_out.xtc* (trajectory), and *md_out.cpt* (checkpoint) files. If these files exist the system processing will be skipped accompanied with the corresponding warning message. To resume an interrupted simulation or extend a completed one, users can specify the path (or paths) to the directory containing *xtc*, *tpr*, and *cpt* files from previous simulations. Additionally, they can supply a new simulation duration in nanoseconds.

Replicas

Repeating the simulation multiple times allows for better statistical sampling of the space, providing more reliable averages and insights into the system's behavior. By default, StreaMD does not support multiple repetition within the same run. Although a user can perform multiple separate runs by applying the same command with different working directory argument (--wdir).

MD analysis

In this phase, the tool undertakes system centering, alignment, and elimination of periodic boundary conditions to yield a trajectory amenable for subsequent MM-GBSA/PBSA calculations and for the retrieval of protein-ligand fingerprints. Consequently, the tool generates a *frame.pdb* file containing the tenth frame of the trajectory for the entire system, alongside *md_short_forcheck.xtc*, which constitutes a subset of the complete trajectory (every 50th frame if the trajectory length is 10 ns or less, and every 100th frame if the trajectory is longer). These files serve for fast visual inspection of the obtained trajectory. Furthermore, the tool calculates root-mean-square fluctuation (RMSF) and radius of gyration for the protein, and root-mean-square deviation (RMSD) values for both the protein and the ligand, individually assessing each cofactor as well. The computed data is saved in PNG format (by seaborn module), facilitating the subsequent analysis.

MM-GBSA/PBSA calculation

The *run_gbsa* module offers a straightforward interface for computing binding free energy using the gmx_MMPBSA tool[17]. To start calculations a user should supply the directories containing simulation outputs generated by StreaMD or external trajectory (*xtc*), topology (*tpr*) and *index.ndx* files. Users have the option to either customize a file containing parameters for MM-GBSA/PBSA calculations (*mmpbsa.in*) or supply their own input file. Upon completion of calculations for all ligands, the module automatically parses and merges outputs in a unified aggregated output file. To facilitate efficient parallel processing, *run_gbsa* utilizes Dask library, dynamically determining the number of processes allocated for each calculation based on the number of frames utilized in the trajectory.

The accuracy of binding free energy calculations depends on multiple factors. The most important are continuum solvation model, interior dielectric constant or entropy treatment. In the present study Interaction Entropy (IE) was used to approximate the binding entropy. IE is computationally very efficient and relatively accurate approach.[18] However, accuracy of entropy estimation can vary substantially for complex and highly flexible systems, therefore, some authors prefer to not perform entropy calculation at all.[19] Meanwhile the correct value of interior dielectric constant may also have significant impact on the estimation of solvation energy especially for simulations of polar or charged molecules. The solute interior dielectric constant value equals 1 is usually used by default, although some works show that it can result in an overestimation of the ligand–receptor electrostatic interaction for some systems and values 2-4 often perform better especially in large data sets of diverse proteins or charged systems.[6] In our pipeline we set up the value of interior dielectric constant to 4 by default, although a user should take into account that the best dielectric constant is system-dependent and some parameter scanning may be required to achieve the highest accuracy.

Protein-ligand fingerprint analysis

The *run_prolif* module facilitates the extraction of protein-ligand contacts through utilization of the ProLIF python library[20]. To start the analysis, users are required to supply directories containing simulation outputs generated by StreaMD or external trajectory (*xtc*) and topology (*tpr*) files. Leveraging Dask for parallel processing, the module enhances computational efficiency. The primary output consists of a text file (*plif.csv*) within each simulation directory, documenting all identified contacts for each trajectory frame. This default behaviour can be customized by adjusting the step parameter to select every n-th frame for analysis. Subsequently, the extracted data is visualized in a 2D plot (*plif.png*) by plotnine module. Additionally, an interactive 2D interaction network (*plif.html*) is generated, showing detected protein-ligand contacts. By default, all contacts will be visualized. This may be misleading in cases if a ligand moves a lot and some contacts cannot be actually established simultaneously. However, users have the flexibility to modify the minimum frequency of occurrence of displayed contacts. Further, protein-ligand interaction fingerprints for all complexes are consolidated into a single file (*prolif_output.csv*), along with a 2D plot (*prolif_output_occupancy0.6.png*) illustrating protein-ligand contacts with a specified minimum occupancy. Users can adjust the default occupancy threshold of 0.6 to suit their preferences.

Logging of calculations

To facilitating identification of issues and tracking the progress StreaMD provides two levels of logging. The general information about each step (e.g. the passed arguments, running and finished steps) is collected in a single log-file placed in the *root* directory of the project. The outputs of individual programs (e.g GROMACS, Antechamber, Gaussian) are collected in separate log-files individual for every processing system and they are located in the corresponding directories of simulating systems. Additional tools (MM-GBSA/PBSA and ProLIF) also produce log-files: one in the *root* directory of the project (or a directory from where the script was launched) and separate log-files for individual systems which are stored in the corresponding directories. Therefore, if there are any errors reported in the general log-file, a user may look at particular log-files to identify an issue.

Results and discussion

The wide functionality of the tool makes it useful for different practical tasks. The tool has been successfully applied in a number of studies, however only few of them have been published so far[21, 22]. Below we will demonstrate the utility of StreaMD on several benchmark datasets and study the computational performance.

GBSA energy calculation

To assess the functionality of the implemented tool, we conducted 10 ns single run simulations and computed Generalized Born Surface Area (GBSA) energies for complexes sourced from the Greenidge dataset[13]. Due to the errors in provided protein and ligand structures simulations were executed successfully for only 556 out of the total 626 complexes. Molecules underwent automatic preparation and pre-processing using the default StreaMD protocol.

To investigate the influence of molecular dynamics simulation duration on the accuracy of calculated binding energy, we analyzed various time frames within the 10 ns trajectories for energy computation. The correlation coefficient remained moderate and slightly dependent in the chosen time frame, ranging from -0.64 to -0.73 (Table S1). The highest correlation was observed when utilizing the first nanosecond of trajectories for binding free energy calculation (Figure 2). However, we suggest to use longer trajectories as they should bring more robust estimates. Omitting the interaction entropy (IE) term, as done in the reference work by Gomes et al. [13], yielded insignificant improvements in correlations (Table S1). The correlation achieved by Gomes et al. for the same set of 556 compounds was comparable at -0.71. It's noteworthy that in their study, free energies were computed from docking poses since ChemFlow integrates docking and MM-GBSA within a unified pipeline, and the trajectory length was 20 ns.

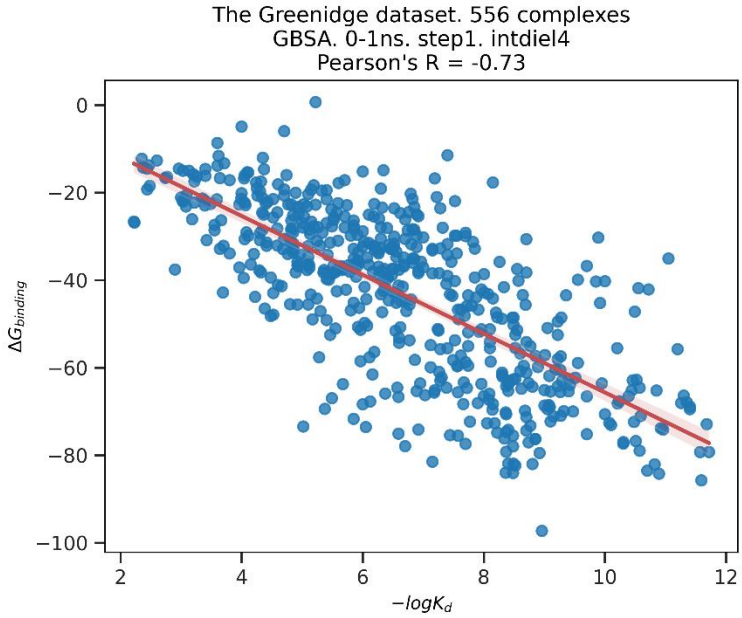


Figure 2. Correlation between calculated MM-GBSA free energies and observed pK_d (Pearson R = -0.73) for 556 protein–ligand complexes from the Greenidge data set[13]. Free energies were calculated from the first nanosecond of the trajectories using all frames and internal dielectric 4.

To further validate the default protocol, we selected three curated datasets of high-quality PDB complexes sourced from the work of Bahia et al.[23]. These datasets encompassed 166 protein-ligand complexes for human β-secretase 1 (UniProt ID: P56817), 63 complexes for human α-thrombin (UniProt ID: P00734), and 51 complexes for bovine trypsin (UniProt ID: P00760). Within each dataset, we identified a reference complex characterized by minimal root-mean-square deviation (RMSD) to all other complexes and high resolution (β-secretase 1: PDB 3UFL, α-thrombin: PDB 4AYY, and bovine trypsin: PDB 1O2I). Subsequently, all other complexes were aligned to their respective reference structures to obtain initial ligand coordinates. Clashes of ligands after alignment were automatically solved during the equilibration and minimization steps, thus no explicit intervention was required. Subsequently, we conducted 1 ns molecular dynamics (MD) simulations for each complex. To compute Generalized Born Surface Area (GBSA) binding free energies, we varied dielectric constants (intdiel = 1 or 4) and considered or disregarded the interaction entropy term.

Based on the analysis (Figure 3), employing a higher internal dielectric parameter (intdiel 4 vs. 1) was generally advantageous, whereas considering the interaction entropy term yielded negligible effects. Notably, for the thrombin dataset, optimal results were achieved with a low internal dielectric parameter (intdiel = 1) and without considering the interaction entropy term (Figure 3). These findings suggest that inclusion of the interaction entropy term may not be necessary for ranking a large set of compounds, as it does not significantly enhance ranking.

For comparative purposes, we conducted molecular docking utilizing Vina[24] and Gnina[25] (dense_ensemble model) integrated in EasyDock[14]. In both cases, the exclusiveness parameter was set to 32. Notably, for the trypsin dataset, both docking programs surpassed the MM-GBSA approach in their ability to rank compounds. In the case of the β-secretase dataset, docking with Gnina exhibited comparable performance to MM-GBSA, while Vina demonstrated inferior performance. Conversely, for the thrombin dataset, a particular setup of MM-GBSA (intdiel = 1 and without interaction entropy) yielded superior ranking capability, followed by Gnina and Vina. While Gnina demonstrated commendable performance, it's worth noting that this might be attributed to the inclusion of some of these compounds in the training of Gnina models. Thus, despite its higher computational demands, the

MM-GBSA approach may offer advantages in certain scenarios, outperforming state-of-the-art docking tools. However, it may necessitate parameter tuning for optimal performance.

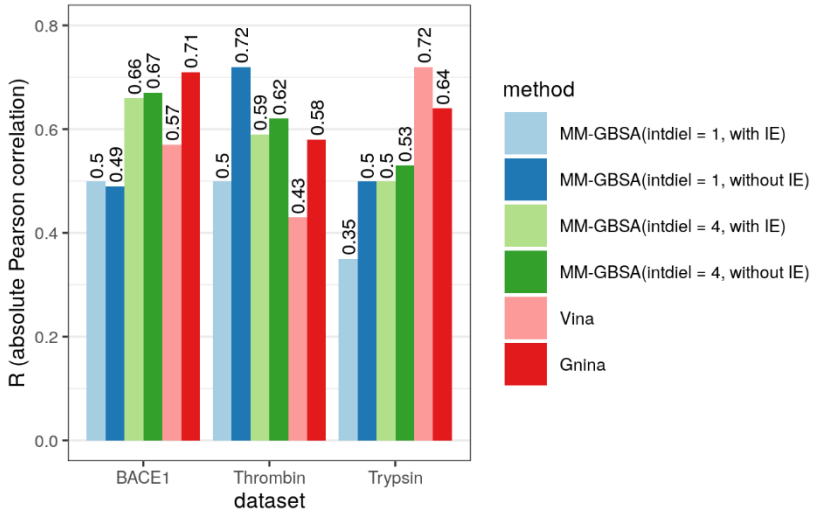


Figure 3. Correlation between docking scores or calculated MM-GBSA free energies for three benchmark data sets. MM-GBSA free binding energies were calculated for different dielectric constants (1 or 4) and considering or ignoring the interaction entropy term (with or without IE). Scatterplots between docking scores or calculated free energies and experimental pK_d values are available in Figures S1-S3.

Scalability and general performance

To assess the scalability of StreaMD, we conducted 51 simulations of the Trypsin dataset, with each simulation comprising 1 ns for NVT and NPT equilibration steps, followed by an additional 1 ns for the production simulation. These simulations were executed in both single-node and multiple-node modes, utilizing a total of 13 nodes, each equipped with 128 CPU cores.

In the single-node mode, the entire process, including preparation, 1 ns MD simulation, and analysis, required 1026 minutes for the 51 complexes. In contrast, the multiple-node mode completed the same tasks in 90 minutes. The calculated overhead was 14%, primarily attributed to the fact that during the preparation and analysis stages, a single molecule is processed on a single CPU core. Given that there were only 51 ligands, not all nodes were fully occupied during these stages, resulting in the observed overhead. However, the simulation stage demonstrated perfect parallelization, efficiently utilizing all cores on all nodes as expected.

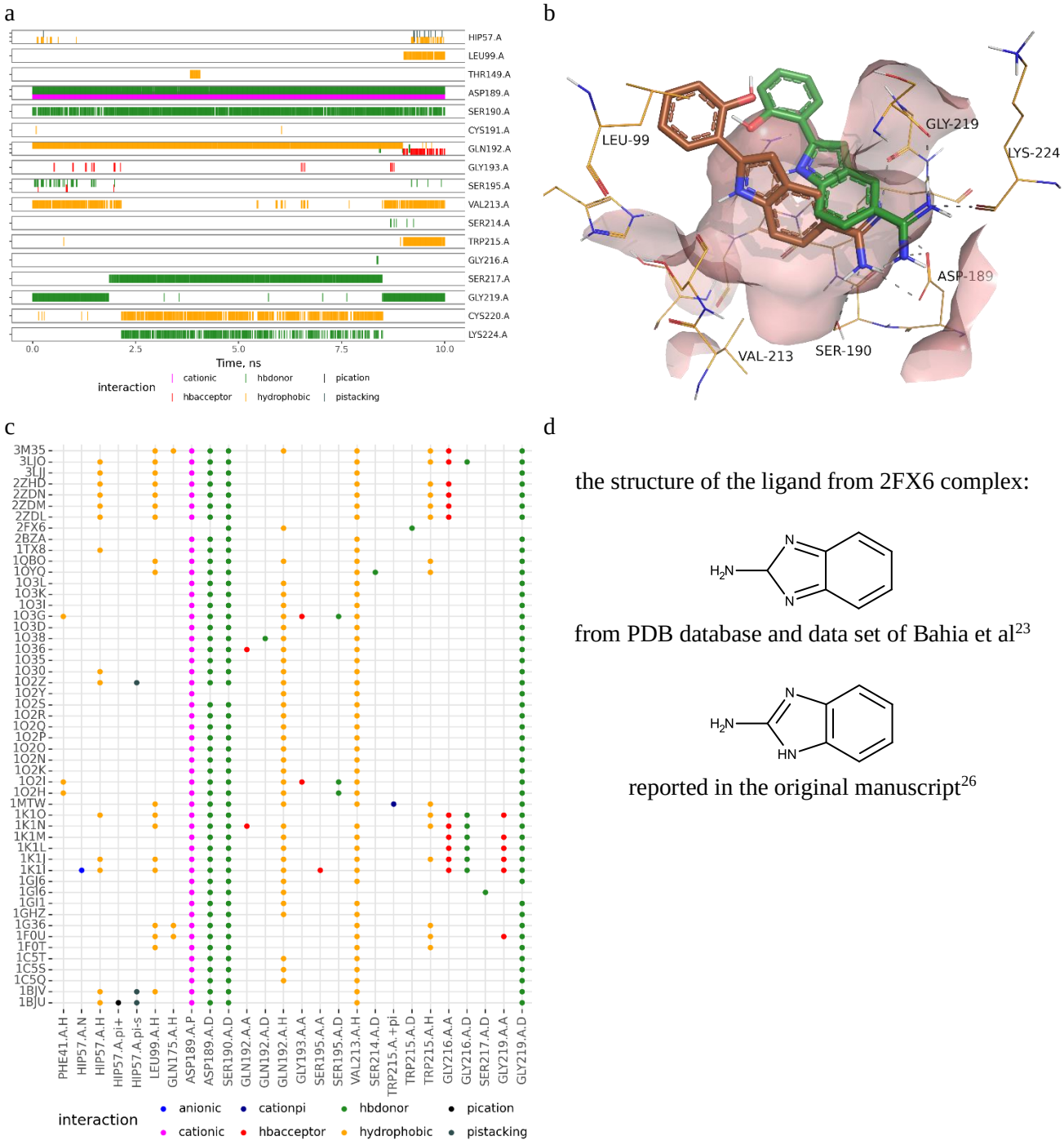
To address this issue, we introduced a specific argument to the program interface, allowing users to selectively execute one of three stages (preparation, simulation, analysis). This flexibility enables users to conduct the preparation step separately on a single server, while simulations can be concurrently executed on all servers in a separate run. By default, all steps are sequentially executed, commencing from input structures of proteins and ligands and concluding with the analysis of obtained trajectories.

Analysis of protein-ligand interactions

An additional analysis of protein-ligand contacts can be performed using ProLIF. The outputs can be visualized for individual protein-ligand systems as well as for a set of systems. We demonstrated these outputs for the dataset of trypsin inhibitors. The analysis of individual protein-ligand systems may show which contacts are co-occurred and how these groups of contacts change during the simulation that may suggest ligand moving or pose changing.

There is an example of the analysis of an individual trajectory in :Figure 4a. The ligand 1GI6 complex forms typical strong interactions with Asp189 and Ser190 of trypsin. Additionally, there are an H-bond with Gly219 and a hydrophobic interaction with Val213. These additional contacts are broken after 2 ns and new contacts are established with Ser217, Cys220 and Lys224. However, after 8 ns the ligand again creates contacts with Val213 and Gly219 along with new ones (Leu99, Trp215). These changes in contacts indicate changes in ligand poses. The ligand after starting of the simulation goes deeper into the binding site and afterwards returns back to the initial pose (:Figure 4b).

The analysis of contacts observed for multiple ligands may help to identify the most frequently observed contacts and interaction patterns and identify ligands which do not follow them, that may indicate their unique binding modes or issues in a simulation setup. The analysis of the whole set of trypsin inhibitors revealed as expected the common interaction pattern. The majority of ligands have charged interaction with Asp189, H-bonds with Ser190 and Gly219 and hydrophobic interactions with Gln192 and Val213 (:Figure 4c). However, a ligand from 2FX6 complex did not follow this pattern. Visual inspection of a ligand MD trajectory revealed that the structure of the ligand was wrongly annotated in the PDB database and was not fixed in the dataset collected by Bahia et al (:Figure 4d). The bond orders were incorrectly interpreted, that results in wrong geometry of the structure and that the ligand started to move away from its initial pose and could not form expected contacts. These simple examples demonstrate how the analysis of protein-ligand interactions may be used to retrieve important and useful information about simulated systems.



:Figure 4. Protein-ligand interactions detected for the trypsin dataset. (a) Interaction fingerprints detected for 1GI6 protein-ligand complex during 10 ns MD simulation. (b) Starting and finishing poses (orange) and the pose in the middle of the simulation (green) of 1GI6 protein-ligand complex during 10 ns MD simulation. (c) Interaction fingerprints for the whole trypsin dataset occurred in at least 60% of frames of 10 ns MD trajectories. (d) Structures of the ligand from 2FX6 complex annotated in PDB and the dataset of Bahia et al and in the original manuscript.

StreaMD options and features:

- default set of optimal parameters to run molecular dynamics, which can be customized
- support of simulations of different molecular systems in explicit water solvent:
    - protein
    - protein-cofactor(s)

    - protein-ligand
    - protein-ligand-cofactor(s)
- support of modeling of boron-containing molecules (using the Gaussian program)
- MCPBPY support to simulate proteins with specific metal ions not parametrized in commonly used force fields
- the ability to continue interrupted simulations or to extend finished ones
- support of distributed computing using Dask library across a network of severs (not necessary a cluster)
- automatic analysis of simulation:
    - separate RMSD plots for protein, ligand and cofactors objects
    - a plot of flexibility of side chains of amino acids (RMSF)
    - a plot and a pdb file with radius of gyration
    - a single frame pdb file for the topology and a short subset of the trajectory for the quick visual inspection
    - a fitted trajectory (with removed periodic boundary conditions, aligned and centered on the first frame) to use for energy or protein-ligand interaction calculations
- support of analysis of MD trajectories by additional instruments:
    - ProLIF: Ligand-Protein interactions
    - MM(PB)GBSA: Calculation of Binding Energy
- logging of every calculation running

StreaMD limitations and remarks:

- preparation of boron-containing molecules and the MCPBPY protocol requires a Gaussian license;
- running a protocol on the number of molecules less than the total number of cores on multiple servers can be inefficient due to inability to distribute the antechamber ligand preparation tasks among more than 1 computational core per ligand;
- StreaMD, as well as GROMACS, can be run only on Linux.

Conclusions

We have implemented a comprehensive automated pipeline capable of conducting molecular dynamics (MD) simulations utilizing GROMACS, calculating binding free energies employing the MM-GBSA/PBSA methodology, and generating protein-ligand interaction fingerprints using ProLIF. The main feature of the developed tool is that it does not require deep knowledge of molecular dynamics and GROMACS. The tool accommodates simulations involving proteins, protein-ligand complexes, and cofactors, with seamless handling of complexes containing specific metal ions (via MCPB.py) and boron-containing ligands (via Gaussian). Furthermore, computations can be efficiently distributed across servers within a network or cluster, facilitated by the Dask Python library with minimal overhead.

Through testing on number of benchmark datasets to evaluate binding free energies using the Generalized Born Surface Area (GBSA) method, we have identified default parameters: employing a dielectric constant of 4 and disregarding the entropy term. The exclusion of the entropy term was recommended due to its marginal impact on enhancing ranking performance, while imposing a computational burden.

Our developed tool holds versatile applicability across diverse scenarios, with particular potential for performing large-scale simulations, such as the calculation of binding free energies utilizing the MM-GBSA/PBSA approach for a substantial number of ligands.

Availability and requirements

Project name: StreaMD

GitHub: https://github.com/ci-lab-cz/streamd

Operating system(s): Linux

Programming language: Python 3

Other requirements: GROMACS, RDKit, Dask

License: MIT

Any restrictions to use by non-academics: no

Competing interests

The author declares no competing interests.

Funding

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through INTER-EXCELLENCE II LUAUS23262, the e-INFRA CZ (ID:90254), ELIXIR-CZ (LM2018131, LM2023055), CZ-OPENSCREEN (LM2018130, LM2023052) grants and by European and Regional Fund project ENOCH (No. CZ.02.1.01/0.0/0.0/16_019/0000868).

Author contributions

A.I. software development, draft manuscript writing and editing; O.M. software development, manuscript editing; P.P. project supervision, draft manuscript writing and editing.

Acknowledgments

The authors thank Alessandra Gilda Ritacca from Department of Chemistry and Chemical Technologies, University of Calabria for consulting with simulations of boron-containing molecules.

References:

(1) King, E.; Aitchison, E.; Li, H.; Luo, R. Recent Developments in Free Energy Calculations for Drug Discovery. *Frontiers in Molecular Biosciences* **2021**, *8*, Review. DOI: 10.3389/fmolb.2021.712085.

(2) De Vivo, M.; Masetti, M.; Bottegoni, G.; Cavalli, A. Role of Molecular Dynamics and Related Methods in Drug Discovery. *J. Med. Chem.* **2016**, *59* (9), 4035-4061, doi: 10.1021/acs.jmedchem.5b01684. DOI: 10.1021/acs.jmedchem.5b01684.

(3) Limongelli, V. Ligand binding free energy and kinetics calculation in 2020. *WIREs Computational Molecular Science* **2020**, *10* (4), e1455. DOI: https://doi.org/10.1002/wcms.1455.

(4) Liao, J.; Nie, X.; Unarta, I. C.; Ericksen, S. S.; Tang, W. In Silico Modeling and Scoring of PROTAC-Mediated Ternary Complex Poses. *J. Med. Chem.* **2022**, *65* (8), 6116-6132, doi: 10.1021/acs.jmedchem.1c02155. DOI: 10.1021/acs.jmedchem.1c02155.

(5) Åqvist, J.; Medina, C.; Samuelsson, J.-E. A new method for predicting binding affinity in computer-aided drug design. *Protein Engineering, Design and Selection* **1994**, *7* (3), 385-391. DOI: 10.1093/protein/7.3.385 (acccessed 3/28/2024).

(6) Wang, E.; Sun, H.; Wang, J.; Wang, Z.; Liu, H.; Zhang, J. Z. H.; Hou, T. End-Point Binding Free Energy Calculation with MM/PBSA and MM/GBSA: Strategies and Applications in Drug Design. *Chem. Rev.* **2019**, *119* (16), 9478-9508, doi: 10.1021/acs.chemrev.9b00055. DOI: 10.1021/acs.chemrev.9b00055.

(7) Chai, X.; Sun, H.; Zhou, W.; Chen, C.; Shan, L.; Yang, Y.; He, J.; Pang, J.; Yang, L.; Wang, X.; et al. Discovery of N-(4-(Benzyloxy)-phenyl)-sulfonamide Derivatives as Novel Antagonists of the Human Androgen Receptor Targeting the Activation Function 2. *J. Med. Chem.* **2022**, *65* (3), 2507-2521, doi: 10.1021/acs.jmedchem.1c01938. DOI: 10.1021/acs.jmedchem.1c01938.

(8) Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L.-P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; et al. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Comput. Biol.* **2017**, *13* (7), e1005659, doi:10.1371/journal.pcbi.1005659.

(9) Doerr, S.; Harvey, M. J.; Noé, F.; De Fabritiis, G. HTMD: High-Throughput Molecular Dynamics for Molecular Discovery. *Journal of Chemical Theory and Computation* **2016**, *12* (4), 1845-1852, doi: 10.1021/acs.jctc.6b00049. DOI: 10.1021/acs.jctc.6b00049.

(10) Harvey, M. J.; Giupponi, G.; Fabritiis, G. D. ACEMD: Accelerating Biomolecular Dynamics in the Microsecond Time Scale. *Journal of Chemical Theory and Computation* **2009**, *5* (6), 1632-1639, doi: 10.1021/ct9000685. DOI: 10.1021/ct9000685.

(11) Bray, S. A.; Senapathi, T.; Barnett, C. B.; Grüning, B. A. Intuitive, reproducible high-throughput molecular dynamics in Galaxy: a tutorial. *J. Cheminf.* **2020**, *12* (1), 54. DOI: 10.1186/s13321-020-00451-6.

(12) Yang, M.; Bo, Z.; Xu, T.; Xu, B.; Wang, D.; Zheng, H. Uni-GBSA: an open-source and web-based automatic workflow to perform MM/GB(PB)SA calculations for virtual screening. *Briefings in Bioinformatics* **2023**. DOI: 10.1093/bib/bbad218 (acccessed 6/20/2023).

(13) Barreto Gomes, D. E.; Galentino, K.; Sisquellas, M.; Monari, L.; Bouysset, C.; Cecchini, M. ChemFlow–From 2D Chemical Libraries to Protein–Ligand Binding Free Energies. *J. Chem. Inf. Model.* **2023**, *63* (2), 407-411, doi: 10.1021/acs.jcim.2c00919. DOI: 10.1021/acs.jcim.2c00919.

(14) Minibaeva, G.; Ivanova, A.; Polishchuk, P. EasyDock: customizable and scalable docking tool. *J. Cheminf.* **2023**, *15* (1), 102. DOI: 10.1186/s13321-023-00772-2.

(15) Lindorff-Larsen, K.; Piana, S.; Palmo, K.; Maragakis, P.; Klepeis, J. L.; Dror, R. O.; Shaw, D. E. Improved side-chain torsion potentials for the Amber ff99SB protein force field. *Proteins: Structure, Function, and Bioinformatics* **2010**, *78* (8), 1950-1958. DOI: https://doi.org/10.1002/prot.22711.

(16) Li, P.; Merz, K. M., Jr. MCPB.py: A Python Based Metal Center Parameter Builder. *J. Chem. Inf. Model.* **2016**, *56* (4), 599-604. DOI: 10.1021/acs.jcim.5b00674.

(17) Valdés-Tresanco, M. S.; Valdés-Tresanco, M. E.; Valiente, P. A.; Moreno, E. gmx_MMPBSA: A New Tool to Perform End-State Free Energy Calculations with GROMACS. *Journal of Chemical Theory and Computation* **2021**, *17* (10), 6281-6291. DOI: 10.1021/acs.jctc.1c00645.

(18) Duan, L.; Liu, X.; Zhang, J. Z. H. Interaction Entropy: A New Paradigm for Highly Efficient and Reliable Computation of Protein–Ligand Binding Free Energy. *J. Am. Chem. Soc.* **2016**, *138* (17), 5722-5728, doi: 10.1021/jacs.6b02682. DOI: 10.1021/jacs.6b02682.

(19) Ekberg, V.; Ryde, U. On the Use of Interaction Entropy and Related Methods to Estimate Binding Entropies. *Journal of Chemical Theory and Computation* **2021**, *17* (8), 5379-5391. DOI: 10.1021/acs.jctc.1c00374.
(20) Bouysset, C.; Fiorucci, S. ProLIF: a library to encode molecular interactions as fingerprints. *J. Cheminf.* **2021**, *13* (1), 72. DOI: 10.1186/s13321-021-00548-6.
(21) Rehulka, J.; Subtelna, I.; Kryshchyshyn-Dylevych, A.; Cherniienko, A.; Ivanova, A.; Matveieva, M.; Polishchuk, P.; Gurska, S.; Hajduch, M.; Zagrijtschuk, O.; et al. Anticancer 5-arylidene-2-(4-hydroxyphenyl)aminothiazol-4(5H)-ones as tubulin inhibitors. *Archiv der Pharmazie* **2022**, e2200419. DOI: https://doi.org/10.1002/ardp.202200419.
(22) Jurášek, M.; Řehulka, J.; Hrubá, L.; Ivanová, A.; Gurská, S.; Mokshyna, O.; Trousil, P.; Huml, L.; Polishchuk, P.; Hajdúch, M.; et al. Triazole-based estradiol dimers prepared via CuAAC from 17α-ethinyl estradiol with five-atom linkers causing G2/M arrest and tubulin inhibition. *Bioorganic Chemistry* **2023**, *131*, 106334. DOI: https://doi.org/10.1016/j.bioorg.2022.106334.
(23) Bahia, M. S.; Kaspi, O.; Touitou, M.; Binayev, I.; Dhail, S.; Spiegel, J.; Khazanov, N.; Yosipof, A.; Senderowitz, H. A comparison between 2D and 3D descriptors in QSAR modeling based on bio-active conformations. *Mol. Inf.* **2023**, *42* (4), 2200186. DOI: https://doi.org/10.1002/minf.202200186.
(24) Trott, O.; Olson, A. J. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* **2009**, *31* (2), 455-461. DOI: https://doi.org/10.1002/jcc.21334.
(25) McNutt, A. T.; Francoeur, P.; Aggarwal, R.; Masuda, T.; Meli, R.; Ragoza, M.; Sunseri, J.; Koes, D. R. GNINA 1.0: molecular docking with deep learning. *J. Cheminf.* **2021**, *13* (1), 43. DOI: 10.1186/s13321-021-00522-2.
(26) McGrath, M. E.; Sprengeler, P. A.; Hirschbein, B.; Somoza, J. R.; Lehoux, I.; Janc, J. W.; Gjerstad, E.; Graupe, M.; Estiarte, A.; Venkataramani, C.; et al. Structure-Guided Design of Peptide-Based Tryptase Inhibitors. *Biochemistry* **2006**, *45* (19), 5964-5973. DOI: 10.1021/bi060173m.