BMC Bioinformatics

SOFTWARE

Open Access

# M01 tool: an automated, comprehensive computational tool for generating small molecule-peptide hybrids and docking them into curated protein structures

Mahsa Sheikholeslami[1*], Mohammad Hasan Nazari[1] and Afshin Fassihi[1*]

*Correspondence:
mahsaa.sheikholeslami@gmail.com; fassihi@pharm.mui.ac.ir

[1] Department of Medicinal Chemistry, School of Pharmacy and Pharmaceutical Science, Isfahan University of Medical Science, Hezar Jerib, Isfahan 817416 - 73461, Iran

## Abstract

**Background:** The field of computational drug design is undergoing rapid advancements, highlighting the need for innovative methods to enhance the efficiency and accuracy of calculating ligand-receptor interactions. In this context, we introduce the M01 tool, a comprehensive computational package designed to facilitate the generation and docking of small molecule-peptide hybrids. M01 integrates several established tools, such as RDKit and EasyDock, into a user-friendly platform that automates the workflow from hybrid generation to docking simulations. This tool is particularly beneficial for researchers with limited chemistry expertise, helping them leverage advanced computational techniques.

**Results:** The M01 tool features an intuitive interface for visualizing molecules and selecting connection points in generating new ligands. It also offers automated receptor preparation using UniProt or PDB IDs and generates default docking configuration files. Furthermore, it includes ligand preparation and docking capabilities through EasyDock and calculates molecular descriptors relevant to drug-likeness properties. Validation studies with peptide-alkoxyamine hybrids demonstrated the tool's effectiveness, generating over 14,000 unique hybrid molecules and showcasing its versatility in drug design applications.

**Conclusions:** The M01 tool represents a significant advancement in computational drug design, streamlining the process of creating hybrid molecules and conducting docking studies. Its ability to automate complex workflows and provide essential molecular insights can empower researchers and enhance the development of novel therapeutics, ultimately contributing to more efficient drug discovery efforts.

**Keywords:** Computational drug design, Hybrid molecules, Docking simulations, Ligand-receptor interactions, ADMET properties, Peptide–drug conjugates

## Introduction

The field of computational drug design is rapidly evolving and plays a crucial role in identifying and optimizing potential drug candidates. A significant number of compounds that undergo drug development fail in clinical trials, necessitating structural modifications to improve their efficacy, pharmacokinetic properties, and effectiveness [1]. Despite extensive efforts and resources spent on the matter, the cost and time to introduce novel drug candidates have increased over the last three decades [2]. In response to these challenges, hybrid generation is one popular strategy for enhancing the pharmacologic properties of drugs. Hybrid compounds consisting of small molecule and peptide components, also known as peptide–drug conjugates, are attractive complex molecules that have emerged as promising candidates in drug design [3–5].

These hybrids may combine the beneficial properties of endogenous peptide ligands and small-molecule antagonists targeting the corresponding receptors. Peptides themselves play a crucial role in regulating numerous biological processes, including those involved in the pathological states of pain [6] and various behavioral processes [6–8], and they also exhibit antibiotic properties [9]. The hybridization of molecules can offer numerous advantages, such as overcoming drug resistance [10], improving the solubility of other drugs [11], and modulating multiple targets [12]. Thus, the rational binding of peptides to small molecules can enhance their properties. Despite these potential advantages, small oligopeptides face several intrinsic challenges, such as biodegradation, increased clearance, and poor bioavailability [13]. To address these issues, pseudo-peptides have been developed [14, 15]. Strategies for creating pseudo-peptides include substituting L-amino acids with D-amino acids, modifying amide bonds, and cyclizing the peptides [16].

Hybrid molecule generation can benefit from computational drug design methods, including molecular design methods and docking simulations, which have greatly enhanced the efficiency and accuracy of ligand-receptor interaction studies [17]. However, the selection of the correct number, type, and order of peptides for hybridization can be complicated. Additionally, these computational methods often require specialized expertise and involve complex procedures, creating barriers for researchers who may not have a strong chemistry background. Therefore, the development of integrable, automated, and user-friendly software is highly desirable for researchers.

One semi-automated tool for docking small molecules is EasyDock [18], designed by Minibaeva and colleagues for customizable docking that incorporates proper ligand preparation and optimization for subsequent docking processes. It primarily uses RDKit modules for three-dimensional (3D) embedding [19], substituting boron atoms, in case found in the ligand structure, with carbons for achieving the possibility of docking calculations through Vina and creating a PDBQT file via the Meeko module [20]. It also enables users to protonate ligands with the pkasovler toolkit, which predicts protonation state with graph convolutional networks. In addition to EasyDock, there are also other necessary modules for docking processes, including PDBFixer for receptor file curation and MGLTools [21] for PDBQT file creation. Additionally, there needs to be an option for AlphaFold structures because there is no access to all of the 3D macromolecule structures through experimental methods. These packages need to be pipe-lined to be used effortlessly.

To address these challenges, we developed the M01 tool, an automated computational package designed for generating small molecule-peptide hybrids and docking them into a curated protein structure within a comprehensive pipeline. The M01 tool integrates several established software packages, including RDKit, EasyDock, PDBFixer, MGLTools, Autodock Vina, and Smina [22–24] into a cohesive platform that streamlines the workflow from hybrid generation to docking simulation. This tool is intended for researchers with limited expertise in chemistry. Compared to existing tools, the M01 tool enables the generation of a list of possible hybrids consisting of all possible combinations of the input peptides, and complete automation through features like receptor download, preparation, and curation. Receptor curation is managed by the PDBFixer package, which scans the protein structure for any missing or non-standard residues and atoms, replacing them according to SEQRES records that contain the sequences of the reported protein tertiary structures. A default docking configuration is then applied, minimizing the need for user interaction and docking expertise.

In addition to the main Jupiter notebook-based tool, we provide a user-friendly web-based graphical user interface (GUI) for ligand generating and an easy way to perform molecular descriptor calculations for pharmacokinetic and drug-likeness properties predictions, facilitating further analysis and predictions.

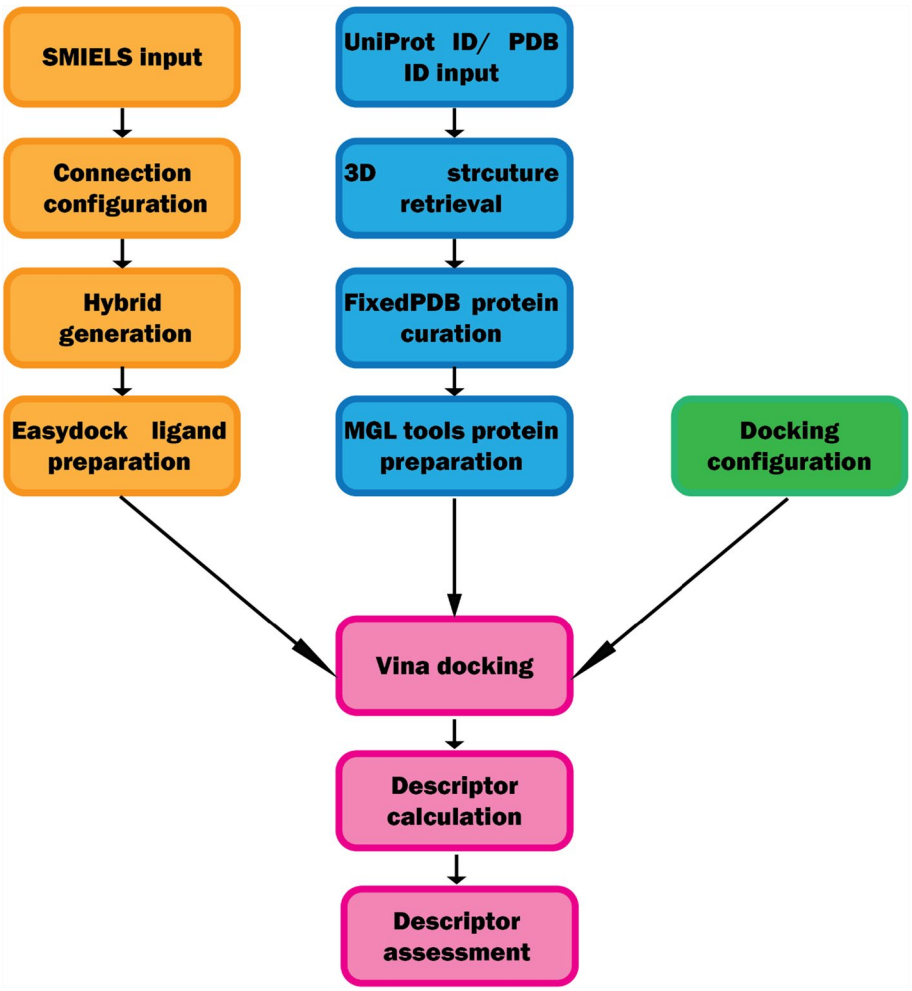
The workflow designed in the M01 tool software is illustrated in Scheme 1.

## Implementation

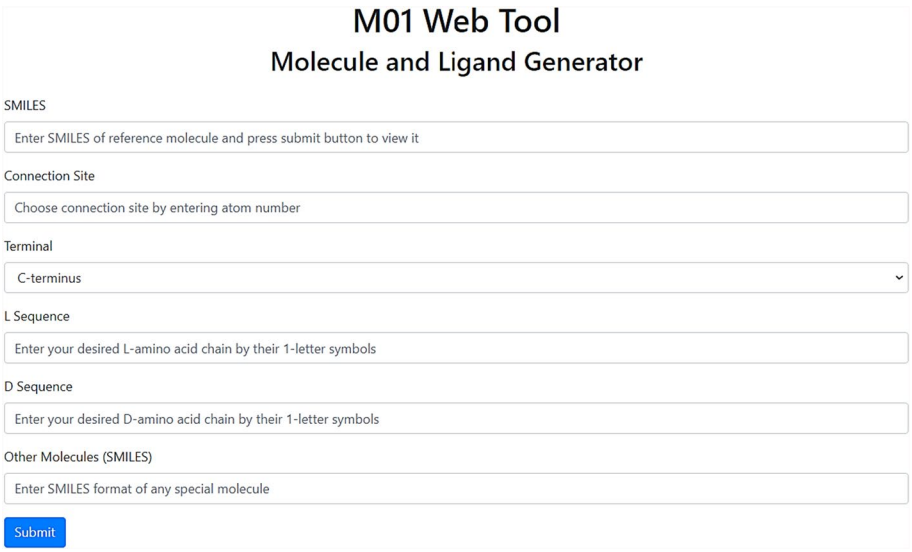
### Hybrid generation

The hybrid generation method involves the creation of small molecule-peptide hybrids using the RDKit module. This library is used for molecular manipulation and for hybridizing small molecules with peptides. The core component of this method is the LigandBuilder class, which manages the formation of peptide bonds and the removal of the hydrogens necessary for this process. A depiction of the graphical online web tool is shown in Scheme 2.

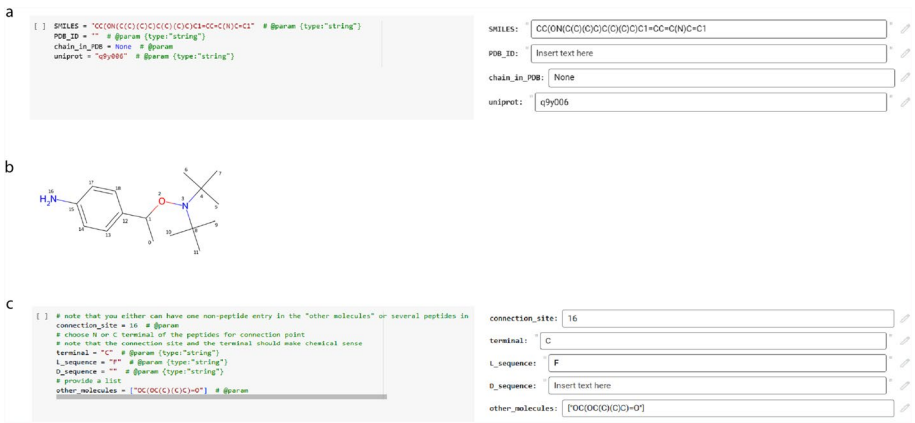
- The *input_amino_acids* function allows users to input sequences of L- and D-amino acids, along with other custom molecules in SMILES format, to create diverse molecular combinations. L- and D-amino acid sequences are converted into RDKit molecule objects using FASTA parsing methods, ensuring compatibility with stereoisomeric configurations. The "other molecules" input handles non-standard amino acids and bulky protecting groups that are capable of forming peptide bonds. In this step, all the unique combinations and permutations of the amino acids are generated and connected to the core ligand to ensure a thorough search for the desirable peptide. RDKit's molecule sanitization and hydrogen addition steps guarantee structural validity, enabling robust hybrid ligand generation. A depiction of the Jupiter notebook is shown in Scheme 3.
- The *create_peptide_bond* method employs predefined SMARTS reaction patterns to form peptide bonds between two molecular fragments. It specifically targets molecules containing carboxylic acid, carbonate, or amine substructures. To maximize bond formation success, the method iterates through four potential reactions, two of which involve hydroxyl groups on the carboxylic terminal.




**Scheme 1** Workflow of hybrid generation and docking using M01 tool




**Scheme 2** A view of the webpage M01 tool




**Scheme 3** **a** Input fields for ligand and receptor information in the M01 notebook tool. **b** Visualization of the input molecule for selecting the connection site. **c** Input fields for specifying the connection site and amino acid sequences

**Table 1** Summary of SMARTS Patterns and Their Corresponding Chemical Reactions for Peptide Bond Formation

| Reaction Type | Reactant 1 | Reactant 2 | Product | Chemical Reaction |
|---|---|---|---|---|
| Reaction with OH (1) | [C:1](= [O:2])[OH:3] | [N:4] | [C:1](= [O:2])[N:4] | Carboxylic acid + Amine → Amide |
| Reaction with OH (2) | [N:4] | [C:1](= [O:2])[OH:3] | [C:1](= [O:2])[N:4] | Amine + Carboxylic acid → Amide |
| Reaction without OH (1) | [N:4] | [C:1](= [O:2])[O:3] | [C:1](= [O:2])[N:4] | Amine + Ester → Amide |
| Reaction without OH (2) | [C:1](= [O:2])[O:3] | [N:4] | [C:1](= [O:2])[N:4] | Ester + Amine → Amide |

Each reaction is implemented as a SMARTS-based transformation using RDKit's ChemicalReactions module. Products of successful reactions undergo validation and are returned as RDKit molecule objects. If no suitable bond is formed, the function provides feedback on the failure. A Summary of SMARTS patterns used in peptide bond formation is detailed in Table 1.

- The *remove_extra_hydrogens* method is designed to fine-tune the hydrogen count on nitrogen atoms, particularly in aromatic heterocycles and other complex molecular structures. The method manually adjusts the explicit hydrogen count to prevent kekulization errors during molecule sanitization. This is crucial for preserving chemical accuracy in nitrogen-containing heterocycles commonly used in drug design.
- The *generate_ligands* method systematically explores peptide-ligand combinations by attaching peptide sequences to a specified connection site of a target small molecule. The method takes two inputs: the connection_site, which identifies the atom index for ligand attachment, and the terminal, which specifies whether the attachment occurs at the carboxyl (C) or amino (N) terminus. For carboxyl-terminal attachments, the method directly replaces hydroxyl groups with the input molecule. For amino-terminal attachments, it first modifies the ligand by adding a

cesium atom to enable precise attachment points. The connection sites are determined using SMARTS queries, ensuring accurate molecular linkage at functional groups. Each generated ligand undergoes validation steps, including hydrogen removal, handling nitrogen heterocycles, structural sanitization, and conversion to canonical SMILES, making them suitable for downstream computational docking simulations.

### Ligand preparation

Ligand preparation is implemented via the EasyDock module, to which ligands can be provided as SMILES strings. Then, RDKit's EmbedMolecule and UFFOptimizeMolecule modules create optimized structures. Protonation is implemented at pH 7.4 or any desired pH with the pkasolver module. Although automated protonation applications are not optimal [25], their use in high-throughput practices is widely accepted [26]. Then, the molecule is converted to PDBQT format via the Meeko module and can be used for docking simulation.

### Target preparation

Either UniProt [27] or PDB IDs [28] can be used as input to prepare a receptor for further docking calculations. In the event of receiving a UniProt ID, the program connects to the UniProt application programming interface (API) provided by the database to query and access its data programmatically. Data retrieved from the API is then parsed to search for PDB IDs and chains related to the desired protein. In cases of multiple PDBs per entry, the chain with the largest amino acid count and highest resolution is selected. After the PDB ID and chain are chosen, the PDB file is downloaded with the Biopython module [29], and extra chains are removed from the file. In this step, the centroid of the chain is also calculated and added to the PDB file for further docking calculations. The PDBFixer module is also integrated with the target preparation module, which is used to fix common problems in PDB files, such as missing atoms, missing residues, non-standard residues, etc. Additionally, all heteroatoms, including water molecules, are removed in this step. PDB files with multiple chains are accessible through PDB IDs.

To address the challenge of docking simulations for proteins without experimentally determined structures, the M01 tool now incorporates a robust feature for fetching and utilizing AlphaFold-predicted structures. This functionality is implemented through the AlphaFoldStructureFetcher class, which automates the retrieval of AlphaFold CIF files from the AlphaFold Database and converts them into PDB format for use in docking simulations. Additionally, the tool calculates the centroid of the protein structure and appends this information to the PDB file, facilitating grid box placement in docking workflows. This enhancement significantly expands the tool's applicability, enabling researchers to perform docking studies on proteins with no available experimental structures while maintaining high-quality input data. The integration of AlphaFold-predicted structures is optional, allowing users to choose between experimental and predicted structures based on their specific needs.

In the M01 tool, the MGLtools tool was used to check hydrogens, add Gasteiger charges [30] to the structure, and remove non-polar hydrogens by merging them into the adjacent carbon atom.

#### Docking configuration

This tool automatically generates default configuration files for docking, which minimizes the required input files. The latest version of Autodock Vina (1.2.5) and Smina (2020.12.10) are used for the docking simulations based on user preferences. The input PDBQT files and chain centroid are prepared as described in the previous sections, while the protein setup and configuration files still need to be created. The chain centroid serves as the center of the grid box, and a box measuring 126 ×126 ×126 Å is selected to encompass the entire chain, allowing for a blind docking process. A high exhaustiveness value of 32 is preferred to compensate for the relatively large grid box size [22, 23]. A maximum of 10 binding modes per ligand will be generated, and in the subsequent analysis, only the mode with the highest binding affinity will be considered.

#### Molecular descriptor calculation

For each ligand, the M01 tool computes several key descriptors related to drug-likeness properties that are crucial for evaluating the drug-likeness and pharmacokinetic behavior of the compound. These descriptors include:

1. *Crippen-Wildman Partition Coefficients (logP)*: This is a measure of the compound's lipophilicity, or hydrophobicity, which reflects its ability to partition between water and an organic solvent, typically octanol. LogP is crucial for predicting a compound's membrane permeability and absorption. The calculation is performed using the Crippen-Wildman partition model, which is based on experimentally derived coefficients for each atom type in the molecule. RDKit's CalcCrippenDescriptors function is used to estimate the logP value [31].
2. *Hydrogen Bond Donor (HBD) Count*: The number of hydrogen bond donors (typically N–H or O–H groups) in a molecule. Hydrogen bonding is a key interaction that influences a compound's solubility, membrane permeability, and receptor binding affinity. A high number of HBD groups can limit the absorption and distribution of a compound, as they increase polarity and reduce lipophilicity. This is calculated by RDKit's CalcNumLipinskiHBD function [32].
3. *Hydrogen Bond Acceptor (HBA) Count*: The number of hydrogen bond acceptors (typically lone pairs on electronegative atoms such as nitrogen, oxygen, or fluorine) in the molecule. Like hydrogen bond donors, hydrogen bond acceptors play a critical role in a molecule's ability to interact with biological targets. This descriptor is calculated using RDKit's CalcNumLipinskiHBA function [32].
4. *Molecular Weight (MW)*: The molecular weight of the compound is calculated as the sum of the atomic weights of the constituent atoms. Molecular weight is a fundamental descriptor for drug-likeness, as it influences various pharmacokinetic properties such as absorption and clearance. RDKit's CalcExactMolWt function is used to compute this property [32].

5. *Topological Polar Surface Area (TPSA)*: TPSA is a descriptor that quantifies the surface area of a molecule that is polar. It is an important predictor of drug absorption, as compounds with high TPSA tend to have poor membrane permeability and are less likely to cross the blood–brain barrier. RDKit's CalcTPSA function is used to compute this descriptor.
6. *Quantitative Estimate of Drug-likeness (QED) Score*: The QED score is a measure of how closely the compound's physicochemical properties align with those of known drugs. It takes into account multiple properties such as LogP, molecular weight, hydrogen bond donors/acceptors, and TPSA, providing a numerical estimate of drug-likeness. The QED score is calculated using RDKit's QED.qed function, where higher values indicate more drug-like compounds [33].

Once the descriptors are computed, they are stored and processed in a structured format, allowing for easy analysis and visualization. These data are then accessible via our drug-likeness_assessment notebook, a tool designed to further analyze and visualize the calculated properties. This notebook provides users with the ability to explore the relationships between the descriptors and assess the potential drug-likeness of the ligands in a comprehensive and interactive manner. The drug-likeness_assessment notebook facilitates the identification of promising compounds based on their drug-likeness properties, aiding in the selection of ligands for further experimental or computational validation.

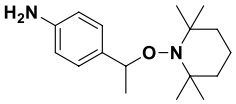
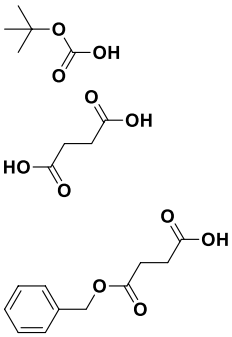
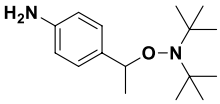
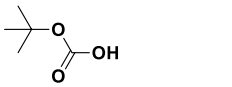
## Results and discussion

In this section, we evaluate the tool's performance by regenerating a set of previously generated ligands and checking the validity, diversity, and drug-likeness properties of the generated molecules. In this article, we refer to the small-molecule substructure of the hybrid as the "core ligand" and to the amino acids and/or other hybridizing molecules as "hybridants."

### Performance

To confirm the accurate performance of the M01 tool software in generating hybrid structures, a set of peptide-alkoxyamine hybrids was utilized in the validation step. These hybrids were previously studied for their antiplasmodial activity by Embo-Ibouanga et al. [34]. The validation process involved selecting a core molecule, choosing the hybridants, and identifying the expected hybrids among the results. The hybrid generation process used three D-amino acids (D-valine, D-phenylalanine, and D-valine) along with three L-amino acids (L-valine, L-phenylalanine, and L-valine). These amino acids were inputted as one-letter symbol sequences. This process resulted in 14,074 unique hybrid molecules, representing all possible combinations of the hybridants. The generated hybrids included the molecules tested in the study by Embo-Ibouanga et al. as well as new hybrid combinations. A list of the amino acids and other molecules used in this hybridization process is provided in Table 2. All generated molecules can be found in Additional File 1. The regenerated hybrid molecules used in the work of Embo-Ibouanga et al. with their given names and IDs are presented in Additional File 2. For more detailed instructions on how to use the tool, please refer to our user manual.

**Table 2** List of Amino Acids and Other Molecules Used for Hybridization

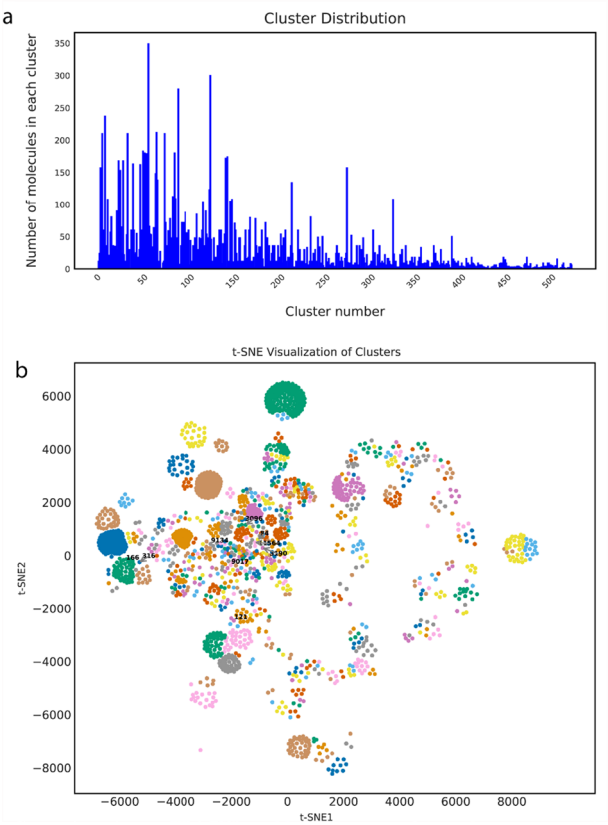
| Core SMILES structure | D-amino acids | L- amino acids | Other molecules |
|---|---|---|---|
|  | Val<br>Phe<br>Val | Val<br>Phe<br>Val |  |
|  | Val<br>Phe<br>Val | Val<br>Phe<br>Val |  |

### Molecular clustering and descriptor visualization

To organize and analyze the generated hybrid molecules, we employed a deep clustering approach that combines the power of Variational Autoencoders (VAE) and K-means clustering as introduced by Hamid Hadipour et al. [35]. Initially, we encoded molecule-specific atom and bond information using a Principal Component Analysis (PCA)-based method, which allowed us to extract both local and global chemical properties of the molecules. Using the embeddings obtained from the VAE, which captures both the global chemical properties and the local atom and bond features, we proceeded to cluster the molecules into 525 distinct groupsThe number of clusters was optimized using the Silhouette, Calinski-Harabasz and Davies-Bouldin indexes, which helped identify the optimal number of clusters based on the similarity between the molecules [36–38].

We then applied the K-means algorithm to the VAE embeddings to group the molecules effectively. K-means, a widely used unsupervised machine learning algorithm, helped us partition the high-dimensional embeddings into clusters based on molecular similarity. After clustering, we visualized the results using t-SNE, a dimensionality reduction technique, which projected the high-dimensional data into two dimensions, making it easier to interpret and analyze the distribution of molecules across the clusters. The visualization of the molecules proposed by Embo-Ibouanga et al. highlights a unique region in the scatter plot where the compounds are clustered, indicating a strong correlation between their structural characteristics and biological activities. This suggests that this clustering method has effectively identified and grouped compounds with similar properties, reinforcing the experimental evidence of their structure and activity relationship.

The successful reproduction of these relationships using our tool suggests its robustness and reliability in identifying potential drug candidates. Furthermore, exploring additional compounds within this specific region of the scatter plot may yield promising candidates for future drug development.



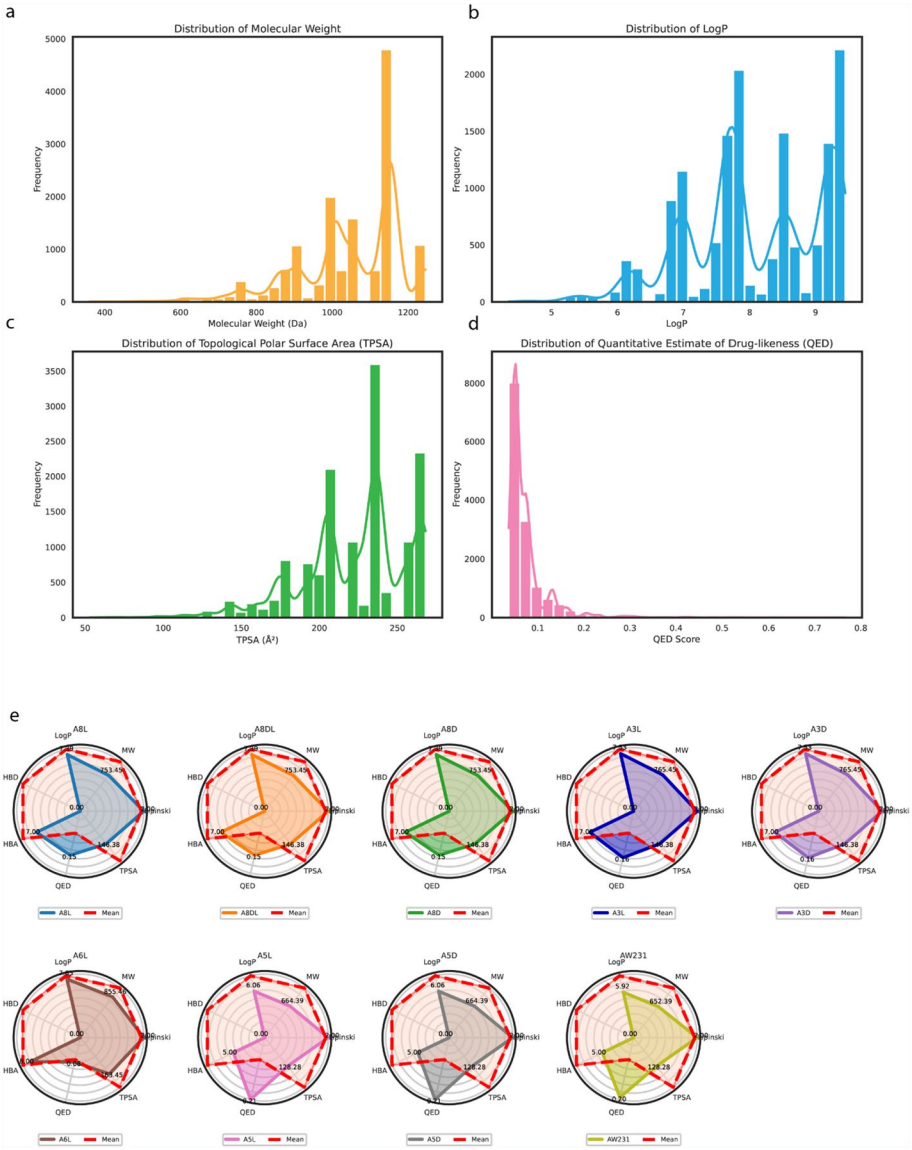

**Scheme 4** **a** Distribution of molecules across 500 clusters, illustrating the grouping of data points based on their feature similarities. **b** t-SNE scatter plot of the dataset, visualizing the two-dimensional embedding of the clustered data points, with key molecules annotated for reference (available in Embo-Ibouanga et al. study)

By leveraging both the clustering technique and our drug-likeness assessment tool, researchers can efficiently prioritize the most promising ligands. This approach allows for the focused evaluation of a selected number of compounds from each cluster, based on their significant structural and activity similarities, thereby streamlining the process of identifying potential therapeutic agents through docking studies. A detailed visualization of the molecular distribution and clustering, including both the embedding space and the cluster-specific molecule groups, can be seen in Scheme 4.

Finally, the drug-likeness properties of the generated ligands were calculated and visualized using the drug-likeness_assessment notebook. This approach systematically evaluates key molecular properties, including MW, LogP, TPSA, HBD, HBA, and QED. The properties of the experimental compounds were then compared to the mean values of the entire batch of generated molecules to assess their drug-likeness potential. The results of these evaluations are summarized in Scheme 5.

### Tool comparison for molecular docking and hybrid generation

Various computational tools have been developed to support molecular docking, hybrid generation, and pharmacokinetics evaluation, each offering different features




**Scheme 5** **a** Distribution of molecular weight in the generated molecules. **b** Distribution of LogP in the generated molecules. **c** Distribution of topological polar surface area (TPSA) in the generated molecules. **d** Distribution of quantitative estimate of drug-likeness (QED) in the generated molecules. **e** Comparative spider charts visualizing molecular descriptors relative to the mean values of all generated molecules

and levels of automation. These tools vary in their ease of use, receptor curation capabilities, and the ability to generate hybrid molecules or analyze pharmacokinetic properties. Additionally, some tools provide advanced functionalities such as scaffold hopping and analog generation, which are beneficial for optimizing lead compounds and exploring chemical diversity in drug design.

A comprehensive comparison of these tools, detailing their key features, ease of use, output types, and capabilities in receptor curation, automation, hybrid generation, pharmacokinetics, and scaffold hopping/analog generation is provided in Table 3.

**Table 3** Comparison of Molecular Docking and Hybrid Generation Tools

| Feature | Key features | Ease of use | Outputs | Receptor curation | Automation level | Hybrid generation | Pharmacokinetics | Scaffold hopping/ analog generation |
|---|---|---|---|---|---|---|---|---|
| M01 Tool | Fully automated hybrid generation, docking pipeline, and receptor curation with RDKit, EasyDock, PDBFixer, MGLTools | User-friendly GUI, minimal expertise required | Lists of hybrids, docking results, drug-likeness properties, and pharmacokinetics | Automated via PDBFixer | High: Minimal user interaction | Comprehensive: All possible hybrids generated | Integrated drug-likeness and molecular descriptor calculations | Yes (analog generation) |
| Schrödinger Suite [39, 40] | Comprehensive molecular modeling and peptide design | Advanced GUI, requires expertise | Binding affinities, property predictions | Manual or semi-automated | Moderate | Not designed for hybrids | Separate modules required | Yes (scaffold hopping supported) |
| AutoDock Vina [22, 41] | Molecular docking and screening | Command-line, moderate learning | Docking scores, binding poses | Pre-prepared PDB files needed | Low | Not supported | Not integrated | No |
| MOE [42] | Peptide modeling, docking, and pharmacophore design | GUI, requires training | Docking results, binding interactions, property analysis | Manual or semi-automated | Moderate | Limited to ligand design | Drug-likeness analysis possible with additional tools | Yes (scaffold hopping supported) |
| FlexAID [43] | Flexible ligand and receptor design | Moderate, simple interface | Docking poses, flexibility analysis | Supports flexible receptors | Moderate | Not supported | Not integrated | No |
| MolHyb [44] | structure-based drug design through molecular hybridization | Simple, user-friendly | Novel Ligands | Not integrated | Low | Yes | Not integrated | Yes (scaffold hopping supported) |

### Limitations and outlook

The model's performance may be limited by the available computational power, especially during large-scale docking simulations of complex molecular structures. A key issue is the reliance on scoring functions, which often oversimplify molecular interactions, neglecting factors like solvation effects, ligand flexibility, and protein dynamics. Additionally, docking studies can be resource-intensive; although AutoDock Vina or Smina is efficient, larger systems or high-throughput screenings still require significant computational power and time. This limitation restricts the number of compounds that can be docked effectively within a feasible timeframe. The stochastic nature of docking algorithms also introduces variability, necessitating multiple runs for reliable results, which further increases computational demands. Additionally, conjugated molecules with amino acids present docking challenges due to their size and complexity. While AutoDock Vina or Smina has advanced molecular docking, researchers must be aware of these limitations and complement computational studies with experimental validation and more sophisticated modeling approaches. The vast number of potential structures and the size of the grid box significantly increase the computational demands. However, even the largest grid boxes often fail to encompass all the molecules of large protein structures, highlighting the need for a more precise, active-site-based grid center.

Additionally, using UniProt IDs as input can present challenges for users. Each UniProt entry typically contains only one protein subunit, whereas many proteins consist of multiple peptides or nucleic acid chains. For certain ligands, such as fluoroquinolone antimicrobials, the binding occurs at the cleavage site between two protein chains of their target receptor, DNA gyrase, and may also interact with DNA. In these scenarios, relying solely on the UniProt ID may not yield acceptable results.

With the introduction of the M01 tool, we aim to automate the docking process extensively, making it user-friendly for the entire scientific community for implementing complex de novo drug design models. Future developments will include options for docking with nucleic acids, a full-option web server platform, homology modeling of proteins, and customizable grid boxes. Moreover, user feedback and real-world applications will offer valuable insights for continuous improvements.

## Conclusion

This study presents an automated hybrid generating and docking pipeline, named M01 tool, in which the user can measure ligand-receptor affinities with minimal previous expertise in the field. This tool is capable of generating various peptides to be hybridized with a small molecule, allowing the identification of the optimal structures. This study mainly consisted of previously generated modules that do not necessitate further validation. With the advent of artificial intelligence and especially machine learning methods, there appears to be an increasing need for simple platforms to provide models with a robust, easy-to-implement reward system for new molecules, which can be handled with this automated, customizable Python-based docking package. The ligand generation and drug-likeness assessment section of this tool is also available at https://m01tool.com/. More features and tools specially docking environment can be accessed by GitHub.

### Availability and requirements

Project name: M01 tool.

Project home page: https://github.com/mahsasheikh/M01-tool.

Operating system(s): Platform independent.

Programming language: Python 3.

Other requirements: RDKit, vina, Smina, PDBFixer, mgltools.

License: MIT-license.

Any restrictions to use by non-academics: no limitation.

#### Abbreviations

| | |
|---|---|
| 3D | Three-dimensional |
| GUI | Graphic user interface |
| API | Application programming interface |
| TPSA | Topological polar surface area |
| QED | Quantitative estimate of drug-likeness |

## Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-025-06120-5.

Supplementary material 1: Molecules generated by M01 tool. Description of data: All of the molecules, as described in the Performance section, are included in this file

Supplementary material 2: Hybrids Regenerated by M01 tool. Description of data: The peptide-conjugated molecules, described in the work of Embo-Ibouanga et al., that were regenerated by this tool are included in this file

#### Acknowledgements

Not applicable.

#### Author contributions

M.S. developed the program, performed tests and wrote the manuscript. M.N. developed the program, developed the online tool. A.F. wrote the manuscript. All authors reviewed the manuscript.

#### Funding

No special funding was applied for this research.

#### Availability of data and materials

All data generated or analyzed during this study are included in this published article and its supplementary information files. The curated protein structures were obtained from the Protein Data Bank (PDB, https://www.rcsb.org/). The M01 tool code and related files are available in supplementary files and on GitHub at https://github.com/mahsasheikh/M01-tool. Additionally, the following datasets and tools were used: Molecular structures (SMILES) obtained from PubChem (https://pubchem.ncbi.nlm.nih.gov/compound/). CondaColab (https://github.com/conda-incubator/condacolab). MGLTools (https://ccsb.scripps.edu/mgltools/). AutoDock Vina (https://vina.scripps.edu/). Biopython (https://biopython.org/). EasyDock (https://github.com/ci-lab-cz/easydock). PDBFixer (https://github.com/openmm/pdbfixer). RDKit (https://www.rdkit.org/). 3D protein structures tracked from UniProt and downloaded from either the RCSB PDB (https://www.rcsb.org/) or AlphaFold (https://alphafold.ebi.ac.uk/files/).

## Declarations

#### Ethics approval and consent to participate

This study does not report on or involve the use of any human participants, human data, human tissue, or animals. Therefore, this section is not applicable.

#### Consent for publication

This manuscript does not contain any individual person's data in any form. Therefore, this section is not applicable.

#### Competing interests

The authors declare no competing interests.

Received: 19 November 2024 Accepted: 27 March 2025
Published online: 14 April 2025

**References**


1. Sun D, Gao W, Hu H, Zhou S. Why 90% of clinical drug development fails and how to improve it? Acta Pharmaceutica Sinica B. 2022;12(7):3049–62.
2. Kiriiri GK, Njogu PM, Mwangi AN. Exploring different approaches to improve the success of drug discovery and development projects: a review. Future J Pharmaceutical Sci. 2020;6(1):27.
3. Wang C, Yang C, Chen Y-c, Ma L, Huang K. Rational design of hybrid peptides: a novel drug design approach. Current Med Sci. 2019;39(3):349–55.
4. Rizvi SFA, Zhang L, Zhang H, Fang Q. Peptide-drug conjugates: design, chemistry, and drug delivery system as a novel cancer theranostic. ACS Pharmacol Trans Sci. 2024;7(2):309–34.
5. Dean TT, Jelú-Reyes J, Allen ALC, Moore TW. Peptide-drug conjugates: an emerging direction for the next generation of peptide therapeutics. J Med Chem. 2024;67(3):1641–61.
6. Nemoto W, Yamagata R, Nakagawasai O, Tan-No K. Angiotensin-related peptides and their role in pain regulation. Biology. 2023;12(5):755.
7. Shevchouk OT, Tufvesson-Alm M, Jerlhag E. An overview of appetite-regulatory peptides in addiction processes; from bench to bed side. Front Neurosci. 2021;15: 774050.
8. Bhat US, Shahi N, Surendran S, Babu K. Neuropeptides and behaviors: how small peptides regulate nervous system function and behavioral outputs. Front Mol Neurosci. 2021;14: 786471.
9. Massari S, Nannetti G, Desantis J, Muratore G, Sabatini S, Manfroni G, Mercorelli B, Cecchetti V, Palù G, Cruciani G. A broad anti-influenza hybrid small molecule that potently disrupts the interaction of polymerase acidic protein–basic protein 1 (PA-PB1) subunits. J Med Chem. 2015;58(9):3830–42.
10. Ouji M, Nguyen M, Mustière R, Jimenez T, Augereau J-M, Benoit-Vical F, Deraeve C. Novel molecule combinations and corresponding hybrids targeting artemisinin-resistant Plasmodium falciparum parasites. Bioorg Med Chem Lett. 2021;39: 127884.
11. Liu X, Zhao L, Wu B, Chen F. Improving solubility of poorly water-soluble drugs by protein-based strategy: a review. Int J Pharm. 2023;634: 122704.
12. Dumitrascuta M, Bermudez M, Trovato O, De Neve J, Ballet S, Wolber G, Spetea M. Antinociceptive efficacy of the μ-opioid/nociceptin peptide-based hybrid KGNOP1 in inflammatory pain without rewarding effects in mice: An experimental assessment and molecular docking. Molecules. 2021;26(11):3267.
13. Song HQ, Fan Y, Hu Y, Cheng G, Xu FJ. Polysaccharide–peptide conjugates: a versatile material platform for biomedical applications. Adv Func Mater. 2021;31(6):2005978.
14. Mäde V, Els-Heindl S, Beck-Sickinger AG. Automated solid-phase peptide synthesis to obtain therapeutic peptides. Beilstein J Org Chem. 2014;10(1):1197–212.
15. Sun X, Li Y, Liu T, Li Z, Zhang X, Chen X. Peptide-based imaging agents for cancer detection. Adv Drug Deliv Rev. 2017;110:38–51.
16. Lee YS. Peptidomimetics and their applications for opioid peptide drug discovery. Biomolecules. 2022;12(9):1241.
17. Bhagat RT, Butle SR, Khobragade DS, Wankhede SB, Prasad CC, Mahure DS, Armarkar AV. Molecular docking in drug discovery. J Pharmaceutical Res Int. 2021;33(30B):46–58.
18. Minibaeva G, Ivanova A, Polishchuk P. EasyDock: customizable and scalable docking tool. J Cheminform. 2023;15(1):102.
19. RDKit [https://www.rdkit.org/]
20. Meeko: preparation of small molecules for AutoDock [https://github.com/forlilab/Meeko]
21. MGLTools [https://ccsb.scripps.edu/mgltools]
22. Trott O, Olson AJ. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. J Comput Chem. 2010;31(2):455–61.
23. Eberhardt J, Santos-Martins D, Tillack AF, Forli S. AutoDock Vina 1.2. 0: new docking methods, expanded force field, and python bindings. J Chem Inform Model. 2021;61(8):3891–8.
24. Koes DR, Baumgartner MP, Camacho CJ. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. J Chem Inf Model. 2013;53(8):1893–904.
25. ten Brink T, Exner TE. Influence of protonation, tautomeric, and stereoisomeric states on protein– ligand docking results. J Chem Inf Model. 2009;49(6):1535–46.
26. Bender BJ, Gahbauer S, Luttens A, Lyu J, Webb CM, Stein RM, Fink EA, Balius TE, Carlsson J, Irwin JJ. A practical guide to large-scale docking. Nat Protoc. 2021;16(10):4799–832.
27. UniProt: the universal protein knowledgebase in 2023. Nucleic acids research 2023, 51(D1): D523-D531
28. Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN, Bourne PE. The protein data bank. Nucleic Acids Res. 2000;28(1):235–42.
29. Cock PJ, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009;25(11):1422.
30. Tiwari R, Mahasenan K, Pavlovicz R, Li C, Tjarks W. Carborane clusters in computational drug design: a comparative docking evaluation using AutoDock, FlexX, Glide, and Surflex. J Chem Inf Model. 2009;49(6):1581–9.
31. Wildman SA, Crippen GM. Prediction of physicochemical parameters by atomic contributions. J Chem Inf Comput Sci. 1999;39(5):868–73.
32. Lipinski CA. Drug-like properties and the causes of poor solubility and poor permeability. J Pharmacol Toxicol Methods. 2000;44(1):235–49.
33. Bickerton GR, Paolini GV, Besnard J, Muresan S, Hopkins AL. Quantifying the chemical beauty of drugs. Nat Chem. 2012;4(2):90–8.
34. Embo-Ibouanga AW, Nguyen M, Paloque L, Coustets M, Joly J-P, Augereau J-M, Vanthuyne N, Bikanga R, Coquin N, Robert A. Hybrid peptide-alkoxyamine drugs: a strategy for the development of a new family of antiplasmodial drugs. Molecules. 2024;29(6):1397.
35. Hadipour H, Liu C, Davis R, Cardona ST, Hu P. Deep clustering of small molecules at large-scale via variational autoencoder embedding and K-means. BMC Bioinform. 2022;23(4):132.

36. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math. 1987;20:53–65.
37. Caliński T, Harabasz J. A dendrite method for cluster analysis. Commun Stat. 1974;3(1):1–27.
38. Davies DL, Bouldin DW. A cluster separation measure. IEEE Trans Pattern Analysis a Mach Intell. 1979;PAMI-1(2):224–7.
39. Sankar K, Trainor K, Blazer LL, Adams JJ, Sidhu SS, Day T, Meiering E, Maier JKX. A descriptor set for quantitative structure-property relationship prediction in biologics. Mol Inf. 2022;41(9):2100240.
40. Yang Y, Yao K, Repasky MP, Leswing K, Abel R, Shoichet BK, Jerome SV. Efficient exploration of chemical space with docking and deep learning. J Chem Theory Comput. 2021;17(11):7106–19.
41. Eberhardt J, Santos-Martins D, Tillack AF, Forli S. AutoDock vina 1.2.0: new docking methods, expanded force field, and python bindings. J Chem Inform Model. 2021;61(8):3891–8.
42. ULC CCG: Molecular Operating Environment (MOE)), 2024.0601. In.: 910–1010 Sherbrooke St. W., Montreal, QC H3A 2R7; 2025
43. Gaudreault F, Najmanovich RJ. FlexAID: revisiting docking on non-native-complex structures. J Chem Inf Model. 2015;55(7):1323–36.
44. Wang H, Pan X, Zhang Y, Wang X, Xiao X, Ji C. MolHyb: a web server for structure-based drug design by molecular hybridization. J Chem Inf Model. 2022;62(12):2916–22.

## Publisher's Note