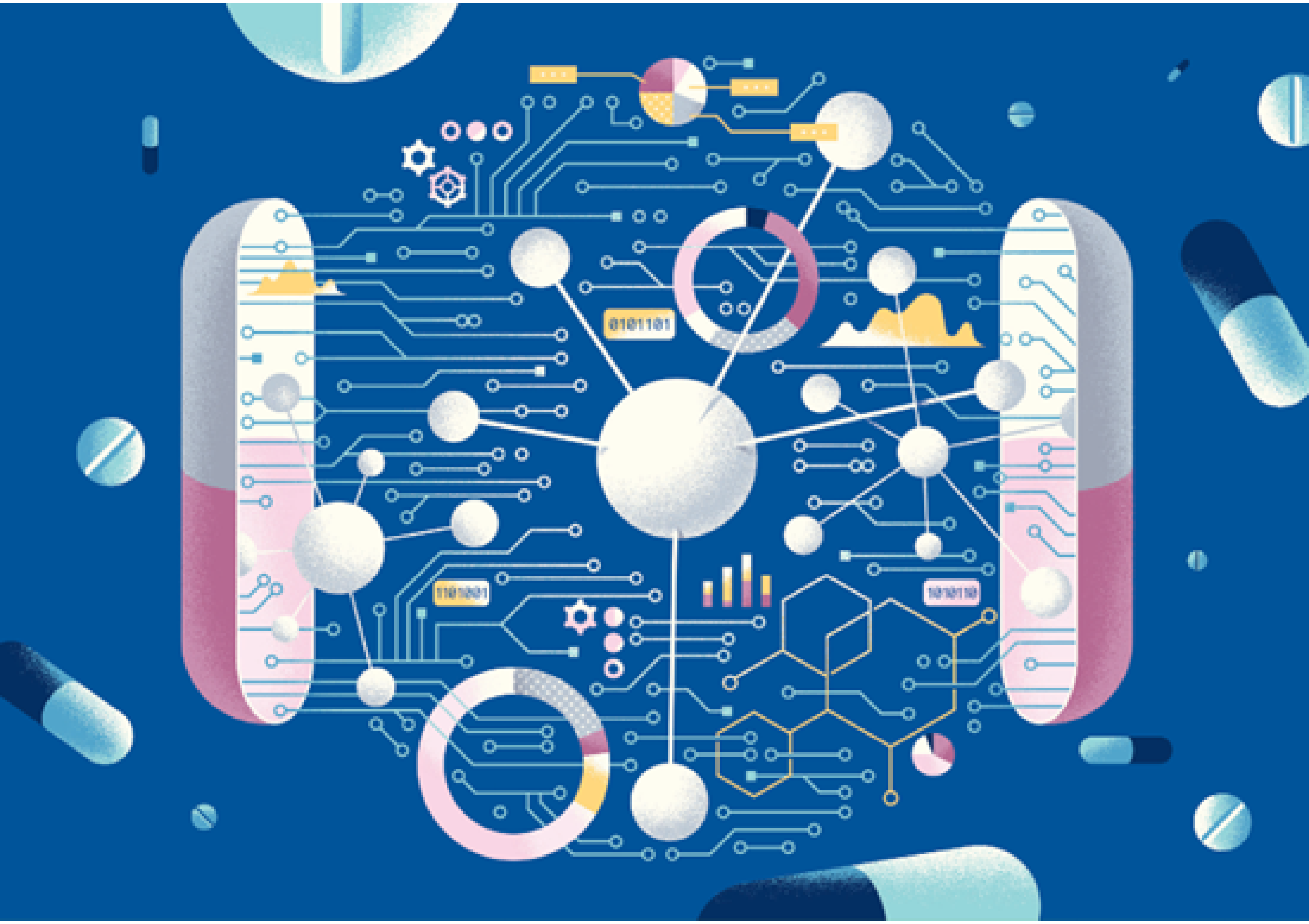
Advancing Targeted Therapies: Bridging the Chemical Space Gap through Deep Learning for Drug Discovery



Universiteit Leiden

TU Delft Delft University of Technology

Erasmus Universiteit Rotterdam

Advancing Targeted Therapies: Bridging the Chemical Space Gap through Deep Learning for Drug Discovery

Kiefer Comassi

Student number: 4402359

26 September 2024

Thesis in partial fulfilment of the requirements for the joint degree of
Master of Science in Technical Medicine
Leiden University ; Delft University of Technology ; Erasmus University Rotterdam

Master thesis project (TM30004 ; 35 ECTS)
TU Delft
20-11-2023 – 10-10-2024

Supervisor(s):
Associate Professor dr. Yann Seimbille
Assistant Professor dr. Gennady Roshchupkin
...

Thesis committee members:
Associate Professor dr. Yann Seimbille Erasmus MC (chair)
Assistant Professor dr. Gennady Roshchupkin Erasmus MC
Associate Professor dr. Rienk Eelkema TU Delft ...

Contents

Abstract

Introduction: This study explores the application of the ChemSpaceAL pipeline, an AI-driven tool for molecular generation, in discovering drug candidates for various molecular targets, such as the HNH domain of Cas9, Fibroblast Activation Protein-alpha (FAP-alpha) and Trophoblast cell surface antigen 2 (TROP2). The goal was to evaluate the pipeline's capacity to generate promising molecules and compare them with known inhibitors, including FDA-approved drugs for c-Abl kinase.

Methods: The ChemSpaceAL pipeline employs deep learning, specifically a Generative Pretrained Transformer (GPT)-based model, to generate novel molecules across multiple iterations. Active learning was used to refine the generated molecules by docking them to specific target proteins and scoring them based on predicted binding affinities. In the case of FAP-alpha, known patented inhibitors were scored to create a benchmark for the AI-generated molecules. The process was iteratively improved by adjusting learning parameters, such as the number of epochs and selection thresholds for active learning.

Results: The pipeline demonstrated the ability to generate molecules with a maximum score of 77 for c-Abl kinase, surpassing the highest score among FDA-approved inhibitors (67.5 for bafetinib), while the average score for the generated molecules was 48.5, compared to 53.1 for the FDA-approved inhibitors. In the case of FAP-alpha, known patented inhibitors scored between 10.5 and 21. AI-generated molecules produced comparable results, with an initial average score of 19.19 and a maximum score of 38.5 in the first iteration. Subsequent iterations saw fluctuations in performance, with iterative improvements stabilizing at an average score of 18.62 and a maximum score of 39 by the third iteration. Adjusting the active learning threshold from the top 10% to 20% of scored complexes yielded more substantial improvements in the molecular generation process.

Conclusion: The results suggest that ChemSpaceAL can explore chemical spaces beyond known inhibitors, occasionally identifying novel molecules with superior predicted binding affinity. However, the study highlights the limitations of relying solely on computational scoring methods, as aspects such as bio-availability and off-target effects are not captured. Future work will focus on reducing the number of active learning epochs to balance model performance and exploring more diverse chemical spaces. The development of a graphical user interface (GUI) and interdisciplinary collaboration for experimental validation will further enhance the pipeline's accessibility and effectiveness, accelerating the drug discovery process.

1 Introduction

In understanding the urgency that motivates this research, it is essential to explore the current landscape of drug development for targeted therapies, especially considering the challenges inherent in the traditional approach. Cancer, a pervasive global health issue and one of the leading causes of mortality worldwide, underscores the need for a reevaluation of conventional treatments, such as chemotherapy and radiation therapy. These treatments often induce severe side effects, impacting both healthy and cancerous cells, and they may not always achieve optimal results. In response to these limitations, targeted therapies have emerged as a promising method in cancer treatment. These innovative approaches aim to specifically target cancer cells or the molecular pathways driving their proliferation, with the goal of minimizing harm to healthy tissues.

The effectiveness of targeted therapies depends heavily on the identification and design of precise components, such as small molecules, that can selectively bind to biomarkers, which include enzymes or receptors involved in cancer cell proliferation or resistance to treatment. Successful therapeutic design requires accurate identification of these biomarkers and pathways to ensure selectivity and efficacy.

In the context of targeted therapy development, selecting appropriate biomarkers is crucial. Poor biomarker selection and insufficient data on binding potential often lead to early failures in drug research programs. Targeted therapies, which are compounds used for disease treatment and diagnosis, contain a specific binding component. They work by interacting with specific biomarkers, such as enzymes or receptors that are overexpressed in diseased tissues. The selection of biomarkers depends on the biological features of the disease being studied, such as cancer or other pathologies, as well as the attributes of the biomarker itself, such as its location, level of expression, and specificity. Targeted therapies can achieve excellent efficacy and diagnostic accuracy by specifically binding to biomarkers that are highly expressed in diseased tissues [13].

In drug development, biomarkers play a crucial role in characterizing patient populations and measuring the degree to which novel medications bind to their intended targets, modify pathological pathways, and produce desired clinical outcomes. In fields such as oncology and neurodegeneration, biomarkers can be used to stratify patients or assess the efficacy of drugs in primary prevention or disease-modification trials. The integration of clinically meaningful biomarkers into regulatory and therapeutic decision-making processes allows for faster identification of promising drug candidates, enabling their rapid progression through clinical development [4].

Traditionally, most cancer drugs aim to block one or more steps in the survival or multiplication of cancer cells. However, this approach poses the risk of harming healthy cells that also require division and preventing natural processes like apoptosis. To mitigate these toxicities, strategies have been developed that conjugate therapeutic agents to small molecule ligands that specifically target tumor cells. This reduces the likelihood of the drug being transported to healthy cells and decreases collateral damage [11].

One of the key challenges in drug discovery is identifying molecules that can bind to a specific biomarker of interest. The vast chemical space of potential drug-like molecules is estimated to range between 10^23 and 10^60 molecules [3]. Despite the existence of compound libraries, only a small fraction of the synthesizable, drug-like chemical space has been explored. Discovering drugs that effectively bind to specific biomarkers in this vast space is a time-consuming and costly process. Recent advances in deep learning (DL) techniques have revolutionized drug discovery by enabling the generation of innovative molecular structures. These approaches can significantly reduce both the time and cost associated with drug development.

One such approach is *ChemSpaceAL* [6], a deep learning-based methodology for molecular generation. ChemSpaceAL leverages generative artificial intelligence models to produce targeted molecules. It has been demonstrated to optimize a GPT-based molecular generator for c-Abl kinase inhibitors and prove its applicability to proteins (such as enzymes and receptors) that currently lack commercially available small-molecule binders. This framework ensures broad relevance in the emerging field of in silico molecular generation and offers open-source accessibility via the ChemSpaceAL Python package. In addition to c-Abl kinase, the HNH domain of Cas9 was also examined in this study. The HNH domain of Cas9 lacks known small-molecule binders, making it a challenging target. By applying the ChemSpaceAL model to such targets, this study explores uncharted chemical spaces and aims to discover novel binders that could serve as starting points for future drug development efforts.

This research seeks to contribute to this burgeoning field by implementing ChemSpaceAL to advance targeted therapies through the generation of new molecules that could potentially be synthesized into viable drugs. Although significant advancements have been made in this field, methods like ChemSpaceAL have not yet been adopted by the Department of Radiology and Nuclear Medicine at Erasmus Medical Centre in Rotterdam. Therefore, this study will focus on the validation and potential implementation of ChemSpaceAL. Specifically, ChemSpaceAL will be used to generate new small molecules that reside outside the existing chemical space. Our efforts will include extending the transferability, robustness, and applicability of ChemSpaceAL. The generated molecules will be validated in collaboration with the Department of Radiology and Nuclear Medicine, utilizing their in-house generated data.

Two key biomarkers will be the focus of this study: *fibroblast activation protein alpha (FAP-alpha)* and *Trophoblast cell surface antigen 2 (TROP2)*.

FAP-alpha was chosen because it is a biomarker of interest for the Department of Radiology and Nuclear Medicine, and they have developed several FAP-alpha binders that will serve as reference points for evaluating the performance of ChemSpaceAL-generated molecules. FAP-alpha is a significant target in cancer research, particularly for its role in tumor microenvironments. Evaluating potential binders for FAP-alpha could yield valuable insights into both diagnostic and therapeutic applications.

In addition, *TROP2* will also be tested. TROP2 is an important biomarker in cancer research, known for its involvement in tumor growth and metastasis. Unlike FAP-alpha, there are currently no small-molecule binders available for TROP2, making it a particularly challenging and novel target. Since there are no existing references for TROP2-binding small molecules, this study presents a unique opportunity to explore uncharted chemical space and potentially discover the first small-molecule binders for this target.

By harnessing the power of deep learning, this study aims to make a meaningful contribution to improving drug development and advancing personalized medicine, with a particular emphasis on cancer treatment. The selection of generated molecules will be tested on fibroblast activation protein alpha (FAP-alpha) and Trophoblast cell surface antigen 2 (TROP2) to assess their potential efficacy.

2 Objectives

2.1 Research Questions

1. How does the choice of biomarker (c-Abl kinase, HNH Domain of Cas9, FAP-alpha, and TROP2) affect the performance of ChemSpaceAL in generating potential drug candidates?

2. Does the ChemSpaceAL model perform consistently across different biomarkers, and how does its effectiveness vary between well-studied targets like c-Abl kinase and more challenging targets like TROP2?

3. What are the key factors that influence the success of ChemSpaceAL in generating effective drug candidates for different cancer-related biomarkers?

3 Expected Results

By validating the recently published tool ChemSpaceAL, weak points will be identified, as well as problems, solving them as we encounter them and extending the existing method to increase robustness and applicability.

4 Literature Review Summary

Many methods using Deep Learning in drug discovery rely on prior knowledge, drawing upon the structure and properties of known molecules to create similar candidates or extracting information on protein pocket binding sites to identify molecules capable of binding [Figure 1]. Deep learning (DL)-based techniques support variable molecule formats such as molecule fingerprints, molecule graphs, simplified molecular input line entry system (SMILES) [10], and three-dimensional (3D) structures. They also have a wide range of applications in the field of de-novo design. SMILES encodes molecular graphs in a concise, human-readable format. It uses a formal grammar to represent molecules using a specific set of characters such as c and C for aromatic and aliphatic carbon atoms, and O for oxygen, as well as symbols like -, =, and # for single, double, and triple bonds. Moreover, the produced molecules could be optimized when the generative model and reinforcement learning (RL) are coupled. In conclusion, deep generative networks have allowed for significant advancements in drug discovery.

The available methods for generating molecules can be categorized into two types: ligand-based molecule generation (LBMG) and pocket-based molecule generation (PBMG). LBMG methods create new molecules by considering the structure and properties of existing ligands. For instance, 3DMolGNN [5] refines a molecule produced by a pre-trained network to closely resemble known inhibitors of the same protein family. This approach aims to generate small molecules that have a strong binding affinity to the target protein and possess desirable drug properties. Their work developed a drug design pipeline by utilizing the capabilities of deep learning and molecular modeling methodologies. This methodology can be helpful in situations when target-specific ligand datasets are few or nonexistent. To generate an initial target-specific dataset, inhibitors of the homologues of the target protein were screened at the target protein's active site. The target-specific dataset's features were learned via transfer learning. The docking scores of the created compounds were predicted using a deep predictive model. Reinforcement learning was used to integrate both of these models to create new chemical entities with optimal docking scores. By creating inhibitors against the human JAK2 protein—none of which had been used for training—the pipeline was validated.

Another example is the work by Merk et al. [9], where they input SMILES strings and fine-tune the models on a smaller dataset of active drugs to generate molecule libraries with specific activity towards the target protein. A new perspective on molecular design is provided by generative artificial intelligence. They describe a deep learning model's first prospective use for creating novel drug-like molecules with desired actions. In order to achieve this, they trained a recurrent neural network to recognize the composition of a sizable collection of well-known bioactive substances, which are represented as SMILES strings. This general model was improved to recognize retinoid X and peroxisome proliferator-activated receptor agonists by transfer learning. Five of the best-ranked compounds created by the generative model were synthesized. In experiments utilizing cells, four of the compounds demonstrated modulatory action of receptors ranging from nanomolar to low-micromolar concentrations.

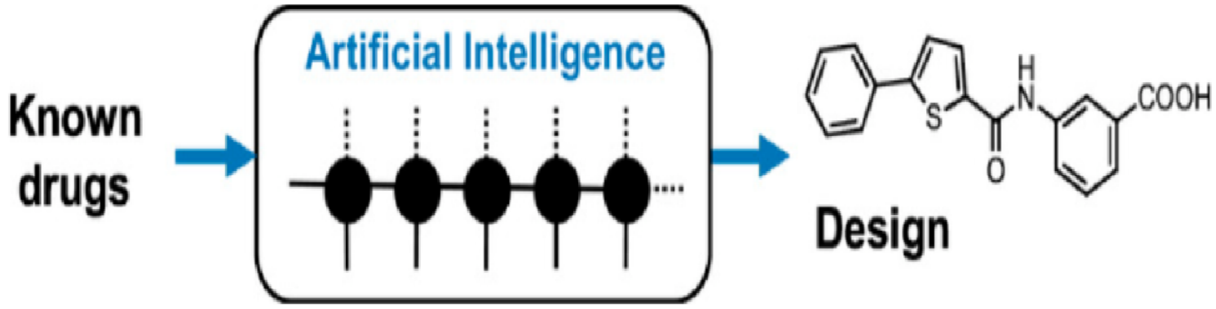


Figure 1: Simplified method of drug design

GENTRL [15], on the other hand, combines generative models with reinforcement learning, variational inference, and tensor decompositions to design inhibitors for a specific protein. It is pretrained using the ZINC database and then further trained using data related to the target protein and common kinase inhibitors. Generative Tensorial Reinforcement Learning (GENTRL) for de novo small molecule design is a deep generative model. GENTRL is optimized for synthetic feasibility, novelty, and biological activity. Using GENTRL, they were able to discover strong inhibitors of DDR1, a kinase target involved in fibrosis and related diseases, in just 21 days. Four compounds were active in biochemistry assays, and two were confirmed in cell-based assays.

PBMG methods, on the other hand, rely on known information about protein binding sites. MolAIcal [1], for example, generates 3D structures of molecules based on the 3D pockets of protein targets. It utilizes the PDBbind database (a comprehensive collection of experimentally measured binding affinity data for protein-ligand complexes deposited in the Protein Data Bank) and Autodock Vina [12] for docking screening.

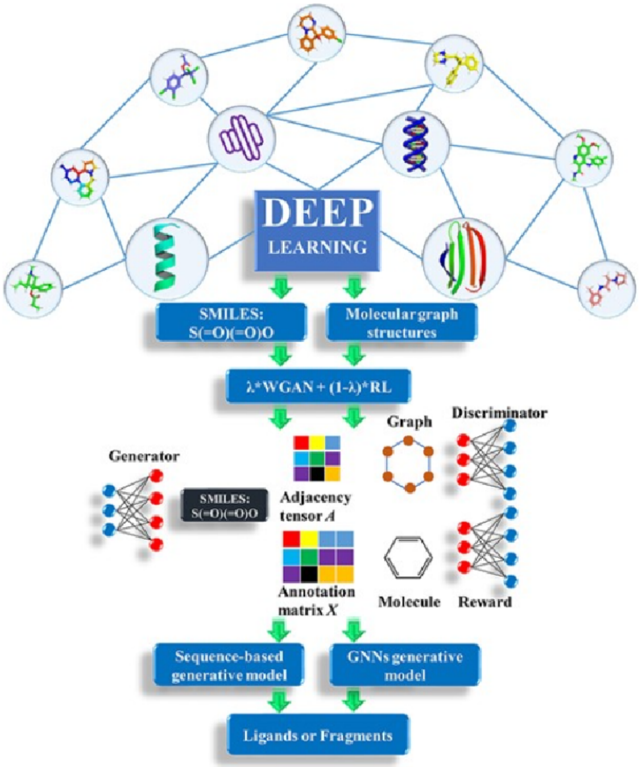


Figure 2: Drug design method using SMILES and Molecular graph structures in a Deep Learning network

MoleAICal is a 3D drug design software that has two main modules. The first module uses a genetic algorithm, a deep learning model trained from FDA-approved drug fragments, and the Vinardo score fitting from the PDBbind database to design drugs. The second module uses a deep learning generic model trained from drug-like molecules in the ZINC database, which is a web-based database with 3D molecules in several formats compatible with most docking programs, and molecular docking invoked from Autodock Vino automatically. To filter out unwanted ligands, they have introduced Lipinski's rule of five [7], PAINS (Pan-assay Interference compounds), and synthetic accessibility (SA). To demonstrate MolAICal's functionality, they selected the glucagon receptor and the SARS-CoV-2 main protease as the focal drug targets.

Miha et al. [14] trained a generative adversarial model to generate compound structures that complement proteins based on the obtained structures of protein-ligand complexes.

Other methods, such as LiGAN and GVAE_SVAE [8], encode protein pockets and use conditional variational autoencoders to generate new molecules that can bind to these pockets. They propose a deep generative model capable of producing 3D molecular structures based on a specific 3D binding pocket. Convolutional neural networks are used to encode atomic density grids into receptor and ligand latent spaces.

The ligand latent space is designed to be variational, allowing the generation of new molecules via sampling. A decoder network then generates atomic densities for novel ligands while taking the receptor into account. These continuous densities are then transformed into discrete atoms, resulting in the creation of unique and valid molecular structures. As they explore further from the seed structure in the latent space, the novelty of generated structures increases, but predicted binding affinity decreases.

Similarly, EGCM_cRNN [2] integrates 3D structural information of protein binding pockets into conditional recurrent neural network models to control the generation of drug-like molecules. This model characterizes protein binding pocket composition using a coarse-grain strategy and represents 3D pocket information through the sorted eigenvalues of the Coulomb matrix (EGCM) of coarse-grained atoms forming the pocket.

Despite substantial progress, challenges remain. LBMG methods struggle to break free from the existing chemical space, limiting their ability to generate truly novel structures. They often produce molecules with only a few new atoms added to the reference molecule.

This is why researchers are now focusing on a third type of method, which steps outside the existing chemical space by generating molecules based on the target protein sequence [2].

In summary, existing methods differ in several key aspects: whether they generate entirely new molecules or simply modify existing ones; their reliance on prior knowledge, such as the 3D structure of the target; whether the generated molecules are outside known chemical space; whether they produce too many molecules that need to be docked individually; and whether they are publicly available. An overview of these differences is presented in Table 1.

Model	**Generates True New Molecule**	**Relies Not On Prior Knowledge**	**Steps Out Of Existing Chemical Space**	**Generates <50 Molecules**	**Accessible Open Source**
3DMolGNN [7]	Yes	No	No	No	Yes
Merk et al. [8]	Yes	No	No	No	Yes
GENTRL [9]	No	No	No	No	Yes
MolAIcal [10]	No	No	No	No	Yes
Miha et al. [13]	Yes	No	No	No	Yes
GVAE_SVAE [14]	No	No	No	No	Yes
EGCM_cRNN [15]	No	No	No	No	Yes
ChemSpaceAL [8]	Yes	No	No	Yes	Yes
Yangyang et al. [7]	Yes	Yes	Yes	No	No

Table 1: Comparison of models based on different criteria

5 Methods

After comparing the different methods, the most promising method appeared to be ChemSpaceAL [6] [Table 1]. Another promising method was that of Yangyang et al. [13], but since this was not publicly available, the method of choice is still ChemSpaceAL. Therefore, this will be the one that will be validated on new target molecules.

5.1 Graphical representation of the Active Learning Methodology from ChemSpaceAL

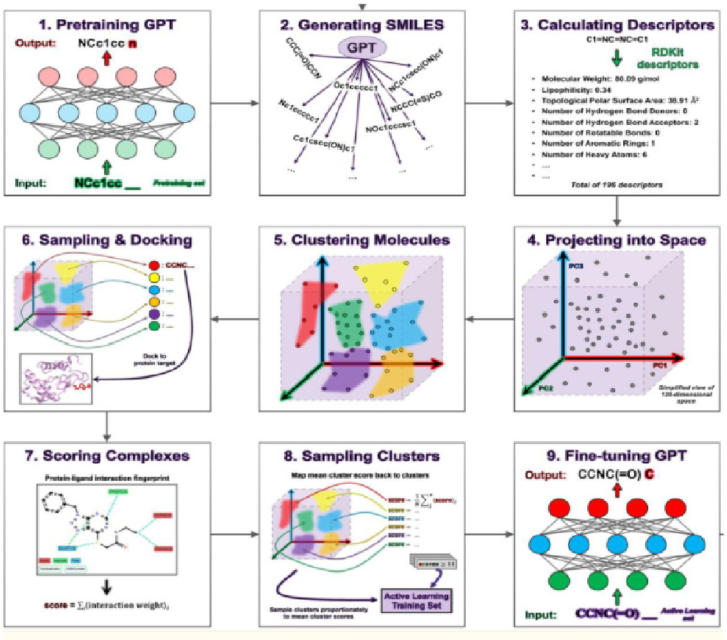


Figure 3: Drug design method using SMILES and Molecular graph structures in a Deep Learning network

The following nine steps form the base of the active learning model 'ChemSpaceAL':

1. Pretrain the GPT-based model on millions of SMILES strings.
2. Use the trained model to generate 100,000 unique molecules.
3. Calculate molecular descriptors for each generated molecule.
4. Project the descriptor vectors into a PCA-reduced space constructed from the pretraining set descriptors.
5. Use k-means clustering on the generated molecules within the PCA space to group those with similar properties.
6. Sample about 1% of molecules from each cluster and dock each of them to a protein target.
7. Evaluate the top-ranked pose of each protein-ligand complex with an attractive interaction-based scoring function.
8. Construct an active learning training set by sampling from the clusters proportionally to the mean scores of the evaluated molecules within each cluster, and combining with replicas of high-scoring evaluated molecules.
9. Fine-tune the model with the active learning training set.

Steps 2–9 are repeated for multiple iterations to progressively improve the model's ability to generate molecules optimized for the target protein.

5.2 Data Collection

The first step that is needed is the data collection and the preprocessing of data. In this case, this has already been done [6]. This was accomplished by merging every SMILES (Simplified Molecular Input Line Entry System) string from ChEMBL 33, GuacaMol v1, MOSES, and BindingDB, eliminating any duplicate strings, and filtering out the strings that the RDKit molecular parser identifies as invalid. The aggregated dataset that is produced has 5,622,772 distinct and legitimate SMILES strings.

While this number may seem large, it is relatively small in comparison to the estimated chemical space, which is believed to range between 10^{23} and 10^{60} molecules. If necessary, this SMILES library could be further extended, as the current dataset may not contain compounds suitable for specific biomarkers like FAP-alpha or TROP2. Extending the library would increase the likelihood of discovering molecules that better fit these challenging targets.

There are originally 196 distinct tokens in this merged dataset. It was found that 148 tokens are represented in the dataset fewer than 1000 times. All SMILES strings containing at least one token that appears fewer than 1000 times were eliminated in order to reduce the size of the vocabulary (from 196 to 48). Rare transition metals and isotopes are included in the majority of the SMILES strings that were rejected.

5.2.1 Preprocessing the SMILES

99.99% of the strings in the dataset had 133 or fewer tokens, with the longest SMILES string in the combined dataset having 1,503 tokens (details in Figures S11.1 – S11.2 in the Supporting Information). A SMILES string length cutoff of 133 is applied, and any string longer than this cutoff is eliminated from the dataset. The longest SMILES string in the dataset (133 tokens) is then expanded, if needed, to match the length of all other SMILES strings.

5.2.2 Preprocessing of the PDB Files

The preprocessing of PDB (Protein Data Bank) files for use in computational pipelines like ChemSpaceAL is a crucial step that streamlines the structural data, focusing on the essential atomic coordinate information. This process involves stripping away various metadata and crystallographic details, leaving only the core ATOM records that describe the three-dimensional structure of the protein.

When a PDB file is initially obtained, it contains a wealth of information beyond just the atomic coordinates. The header section, for instance, provides valuable context about the structure, including details about the experiment that determined the structure, publication information, and various annotations. While this information is undoubtedly valuable for understanding the broader context of the protein structure, it is not directly utilized in most computational analyses or docking simulations.

Similarly, the crystallographic information, such as the ORIGX and SCALE lines, provides critical data for crystallographers about how the crystal structure relates to the coordinate system. However, for the purposes of molecular docking and many other computational chemistry tasks, this information is superfluous.

By retaining only the ATOM records, the preprocessing focuses on the essential information needed for computational analysis. Each ATOM record contains critical details: the atom's serial number, name, the residue it belongs to, its chain identifier, sequence number, three-dimensional coordinates, occupancy, temperature factor (B-factor), and element symbol. This distilled information forms the backbone of most structural analyses and simulations.

The reasons for this preprocessing are multifaceted. Primarily, it simplifies the input, creating a more standardized format that reduces the likelihood of parsing errors in downstream analyses. This standardization is particularly crucial when dealing with large numbers of structures or when computational resources are at a premium. The reduced file size that results from this streamlining can significantly improve efficiency, especially when processing multiple structures.

Moreover, many computational chemistry tasks, including molecular docking and scoring, rely solely on the atomic coordinates and basic atom information. By focusing on these essential data points, the preprocessing ensures that all structures are treated consistently, regardless of their original source or the amount of additional information they might have contained. This consistency is key to ensuring reproducible and comparable results across different structures and experiments.

However, it's important to note that this preprocessing is not without its trade-offs. In stripping away the additional information, some potentially useful data is lost. Information about ligands, non-standard residues, or experimental conditions, which might be important for interpreting results or for more specialized analyses, is removed. This loss of information is generally acceptable for broad computational studies but may limit certain types of in-depth analyses that rely on these additional details.

In the context of the ChemSpaceAL pipeline, this preprocessing serves a vital role. It ensures that the focus remains squarely on the protein structure's atomic coordinates, which are the critical elements for tasks like molecular docking and scoring. By removing extraneous information, the pipeline can operate more efficiently and with reduced risk of errors that might arise from variations in PDB file formats or unnecessary data.

This streamlined approach aligns well with the goals of computational drug discovery pipelines, where the emphasis is on rapid screening and evaluation of potential drug candidates. The simplified PDB format allows for faster processing, enabling researchers to explore larger chemical spaces and iterate through more potential molecules in their search for promising drug candidates.

In essence, the preprocessing of PDB files represents a careful balance between retaining essential structural information and optimizing for computational efficiency. While some granularity is lost in the process, the benefits in terms of standardization, efficiency, and focus on the most critical data make it an invaluable step in modern computational chemistry workflows, particularly in the realm of drug discovery and design.

5.3 Setup and Dependencies

To ensure smooth execution of the ChemSpaceAL pipeline, several key dependencies must be properly installed. The following packages with specific versions are required: `prolif` (2.0.1), `pandas` (1.5.3), `numpy` (1.23.1), `rdkit` (2023.03.3), and `torch` (2.1.0). Additionally, a CUDA version of 9 or higher is necessary for optimal performance, particularly for the deep learning components of the pipeline.

To facilitate this setup process, a dedicated initialization cell will be added to the pipeline notebook. This cell includes clear instructions for installing these dependencies, along with commands to check for their correct installation and version compatibility. For packages that may require specific installation procedures, such as `rdkit` or `torch` with CUDA support, detailed guidance is provided.

Users are strongly advised to execute this initialization cell and verify all dependencies before proceeding with the main pipeline. This step is crucial for avoiding runtime errors and ensuring consistent performance across different environments. Furthermore, where possible, local installations of these packages are recommended to maintain a stable working environment.

The initialization cell also includes a script to check for these dependencies and their versions, helping users quickly identify any missing or incompatible packages. This proactive approach to environment setup significantly reduces the likelihood of errors during the pipeline's execution and ensures a smoother experience for first-time users.

By clearly stating these requirements and providing an automated way to check and install them, the ChemSpaceAL pipeline becomes more accessible to researchers who may not have extensive experience with Python environment management or deep learning frameworks. This attention to the initial setup process is a critical step in making the tool more user-friendly and reducing potential barriers to its adoption in diverse research settings.

5.4 Workflow Overview

The code works as follows. The first step is that the GPT (Generative Pretrained Transformer)-based model is trained on millions of SMILES (Simplified Molecular Input Line Entry System) strings. The GPT-based transformer is fine-tuned toward, for example, a protein with FDA-approved small molecule inhibitors like c-Abl kinase. A dataset is created using various datasets to build a rich internal representation of these SMILES strings, enabling it to generate a diverse array of molecules (in total about 5.6 million SMILES strings).

The second step is to let the trained model generate 100,000 molecules. Then the molecular descriptors are calculated for each generated molecule. These molecular descriptors are projected into a PCA (Principal Component Analysis)-reduced space, constructed from the descriptors of these generated molecules. K-means clustering is used to group the generated molecules with similar properties within the space. About 1 percent of each of these clusters is docked onto the target protein. For evaluation, an interaction-based scoring method is used, along with fine-tuning if needed. This whole process can be repeated for multiple iterations. For this research, the ChemSpaceAL method is used to generate and evaluate molecules outside of the existing chemical space using this active learning pipeline.

5.5 Pipeline Execution and Environment Setup

The pipeline was executed via a SLURM script, submitted to the GPU cluster using the command:

```
sbatch full_iteration.sh
```

The code was implemented in a Jupyter Notebook environment using Python, with the source code available on GitHub at `batistagroup/ChemSpaceAL`.

5.6 Adapting for Jupyter Notebook Execution

To ensure broader compatibility, the pipeline was modified to use the `subprocess` module instead of shell commands. For example:

```
!python /content/DiffDock/datasets/esm_embedding_preparation.py
--protein_ligand_csv /content/input_protein_ligand.csv
--out_file /content/DiffDock/data/prepared_for_esm.fasta
```

was replaced with `subprocess` calls, such as:

```
import subprocess

subprocess.run([
    "python", "/content/DiffDock/datasets/esm_embedding_preparation.py",
    "--protein_ligand_csv", "/content/input_protein_ligand_csv",
    "--out_file", "/content/DiffDock/data/prepared_for_esm.fasta"
], check=True)
```

5.7 Parameter Tuning Process

Before script submission, the current protein ID and iteration number were verified and adjusted in the 'full_iteration.sh' script. Further parameter tuning, including the threshold and number of epochs, was performed in the '50_refitModel_activeLearning.py' script.

5.8 Experimental Validation

After molecules have been generated and selected, it is crucial that they undergo experimental assessment. This step should be conducted in collaboration with the Department of Radiology and Nuclear Medicine. The generated molecules should be evaluated for viability using the specific target protein for which they were designed, such as fibroblast activation protein-alpha (FAP-alpha) or TROP2, depending on the focus of the study. This experimental validation is essential to confirm the computational predictions and assess the actual effectiveness of the generated molecules. Based on the outcomes of these experiments, further adjustments to the generation process or the molecules themselves may be necessary to improve their efficacy and suitability as potential therapeutic agents.

5.9 Establishing Reference Scores for Target Molecules

A critical step in this methodology is the establishment of reference scores for each target molecule by evaluating existing inhibitors. This process involves collecting SMILES representations of known inhibitors for the specific protein target and subjecting them to the same scoring function used in the generative pipeline. By doing so, a benchmark can be created against which the generated molecules can be compared. This reference score serves multiple purposes:

1. **Realistic Target Setting:** It provides a realistic target for the optimization process. Knowing the scores of effective, existing inhibitors gives a clear goal to aim for in the molecular generation.

2. **Scoring Function Validation:** It helps in validating the effectiveness of the scoring function. If the function correctly assigns high scores to known effective inhibitors, it increases the confidence in its ability to identify potentially effective new molecules.

3. **Performance Assessment:** It offers a means to assess the relative performance of the generated molecules. The scores of the new molecules can be directly compared to those of known inhibitors, giving us a tangible measure of how promising the generated compounds might be.

4. **Threshold Determination:** The scores of known inhibitors can help in setting thresholds for selecting promising candidates from the generated set. Molecules scoring above or close to these thresholds would be prioritized for further investigation.

For instance, in the case of c-Abl kinase, the SMILES of FDA-approved inhibitors such as imatinib, nilotinib, and dasatinib will be evaluated. Each of these drugs has demonstrated effectiveness in clinical settings, making their scores valuable benchmarks. Then, these SMILES will be processed through the scoring pipeline, which includes docking simulations and interaction analysis, to obtain their reference scores.

The scores of these known effective inhibitors guide the generative process and help contextualize the results. For example, if the generated molecules consistently score higher than known inhibitors, it could indicate that a chemical space with potentially higher binding affinities is being explored. Conversely, if the molecules consistently score lower, it might suggest that the model needs further refinement or that a different area of chemical space is being explored.

Moreover, comparing the scores of the generated molecules to these references allows for gauging whether compounds with potential binding affinities similar to or better than existing drugs are being generated. This comparison is crucial in the early stages of drug discovery, as it helps prioritize which generated molecules are worth pursuing for further experimental validation.

While the initial validation of ChemSpaceAL focused on c-Abl kinase, a target with well-known small molecule inhibitors, the true potential of this pipeline lies in its application to novel or challenging targets. Two of such target are TROP2 (Trophoblast cell-surface antigen 2) and FAP-alpha, both proteins of significant interest in cancer research. Unlike small molecule inhibitors, which are represented by SMILES, therapeutic approaches for TROP2 are currently antibody-based and cannot be easily addressed using traditional SMILES input. The lack of small molecule inhibitors for TROP2 presents both a challenge and an opportunity for ChemSpaceAL to explore new chemical space and identify novel modulators.

The pipeline's ability to generate diverse molecules and score them based on predicted interactions with the target protein becomes especially valuable in this case. By applying ChemSpaceAL to TROP2, researchers aim to identify small molecule leads that could complement or potentially offer advantages over existing antibody therapies, such as improved tissue penetration or oral bioavailability. This application underscores the versatility of the ChemSpaceAL approach in addressing targets that lack established small molecule inhibitors, potentially opening new avenues in drug discovery for challenging targets.

A comprehensive guide detailing the setup and usage of the ChemSpaceAL pipeline, including the adaptations for Jupyter Notebook environments and the considerations for epoch adjustment, is provided in Appendix A. This guide offers step-by-step instructions for researchers looking to implement or adapt the pipeline for their specific needs, ensuring broader accessibility and reproducibility of this methodology.

6 Results

6.1 Implementation of the ChemSpaceAL Pipeline

The ChemSpaceAL pipeline was successfully implemented in a Jupyter Notebook environment, enhancing its accessibility for users at the Erasmus Medical Centre. Key outcomes of the implementation include:

1. Automation of multiple iterations of SMILES string generation, sampling, docking to target molecules, and scoring of docked complexes.

2. Creation of a system that requires only the target molecule's PDB file as input for generating candidates.

3. Development of a parallelized docking process, theoretically reducing docking time for 1,000 SMILES from 18 to 9 hours, and total iteration time from 22 to 13 hours.

4. Implementation of automatic directory creation for storing target-ligand complexes, preventing overwriting of previous files.

5. Addition of features such as email notifications upon completion or error occurrence, and visualization of the top ten scoring candidates using RDKit's `draw` module.

6.2 Handling of Molecular Sanitization Errors

During the execution phase, particularly in scoring, molecular sanitization errors were encountered despite initial filtering for valid SMILES strings. The `scoring.py` script was successfully modified to handle these errors by skipping problematic molecules and continuing with the next, enhancing the pipeline's robustness.

6.3 Parameter Tuning Outcomes

Initial parameter settings, such as basing the threshold on the average of the top 30% of scores, required adjustments between iterations for optimal results. A threshold that approximately 30% of the complexes could reach generally yielded satisfactory results, though this often needed adjustment between iterations.

6.4 Molecular Evolution comparison with the ChemSpaceAL paper

Inhibitor	**Score**
Imatinib	62.5
Bosutinib	44.5
Asciminib	41.0
Nilotinib	53.5
Ponatinib	61.0
Bafetinib	67.5
Dasatinib	41.5
Range	41.0 - 67.5
Average	53.1

Table 2: Scoring results for FDA-approved c-Abl kinase inhibitors from the ChemSpaceAL paper

Table 2 presents the scoring results for seven FDA-approved c-Abl kinase inhibitors using the ChemSpaceAL pipeline's scoring function. The scores range from 41.0 to 67.5, with bafetinib achieving the highest score of 67.5 and asciminib the lowest at 41.0. Imatinib and ponatinib also scored relatively high at 62.5 and 61.0, respectively. The average score across all seven inhibitors is 53.1. These scores provide a benchmark for evaluating the performance of the AI-generated molecules against known, effective inhibitors of c-Abl kinase.

HNH	Average Score	Percentage of Scores Above 11 (%)	Maximum Score
al0	7.9	21.30	32.50
al1	9.1	31.90	26.50
al2	9.8	39.10	25.00
al3	10.4	43.90	23.00
al4	11.1	50.10	33.50
al5	11.5	52.10	34.00

Table 3: Evolution of protein-ligand attractive interaction scores between molecules from the ChemSpaceAL paper for HNH

HNH	Average Score	Percentage of Scores Above 11 (%)	Maximum Score
al0	8.21	21.10	29.50
al1	7.86	17.80	29.00
al2	7.93	19.10	31.00
al3	8.51	23.30	33.50
al4	9.02	26.50	31.00
al5	9.85	32.70	36.00
al6	11.57	44.40	46.00

Table 4: Evolution of protein-ligand attractive interaction scores between molecules for HNH

Table 3 presents the evolution of the SMILES scores for the HNH domain of Cas9 that the authors from ChemspaceAL found Table 4 presents the evolution of protein-ligand attractive interaction scores between generated molecules and the HNH domain of Cas9 across the active learning iterations that were generated in this thesis. For each iteration, the table displays three key metrics: the percentage of generated molecules with scores above the threshold (%>Threshold), the mean score of all generated molecules, and the maximum score achieved in that iteration. Iteration 0 represents the initial state of the model after pretraining, while subsequent iterations reflect the model's output following each round of active learning. This table allows for a quantitative assessment of how the distribution of scores changes throughout the iterative process. The complete set of top-scoring molecules for each iteration can be found in Appendix B, providing a more detailed view of the generated compounds.

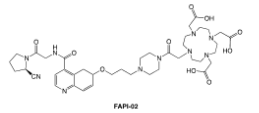
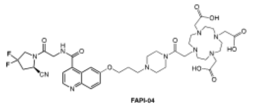
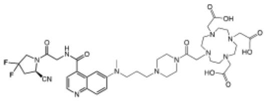
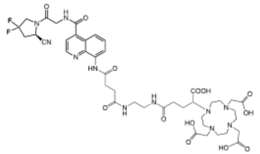
FAPIs	SMILES	SCORE
 FAPI-02	O=C(NCC(N1CCC[C@@H]1C#N)=O)C2=C3C(C=CC(OCCCN4CCN(C(CN(CCN(CC(O)=O)CCN(CC5)CC(O)=O)CCN5CC(O)=O)=O)CC4)C3)=NC=C2	12.5
 FAPI-04	O=C(NCC(N1CC(F)(F)C[C@@H]1C#N)=O)C2=C3C(C=CC(OCCCN4CCN(C(CN(CCN(CC(O)=O)CCN(CC5)CC(O)=O)CCN5CC(O)=O)=O)CC4)=C3)=NC=C2	17
 FAPI-46	O=C(NCC(N1CC(F)(F)C[C@@H]1C#N)=O)C2=C3C(C=CC(N(C)CCCN4CCN(C(CN(CCN(CC(O)=O)CCN(CC5)CC(O)=O)CCN5CC(O)=O)=O)CC4)=C3)=NC=C2	17
 ONCOFAP-DOTAGA	O=C(NCC(N1CC(F)(F)C[C@@H]1C#N)=O)C2=C3C(C(NC(CCC(NCCNC(CCC(N(CCN(CC(O)=O)CCN(CC4)CC(O)=O)CCN4CC(O)=O)C(O)=O)=O)=O)=O)=CC=C3)=NC=C2	10.5

Figure 4: Molecular structure and score of FAP-alpha inhibitors.

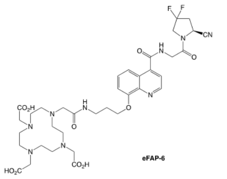
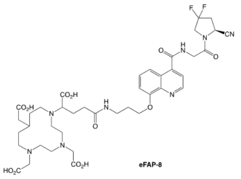
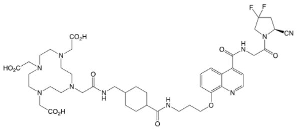
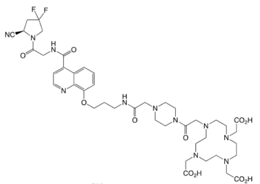
 eFAP-6	O=C(NCC(N1[C@H](C#N)CC(F)(F)C1)=O)C2=CC=NC3=C(OCCCNC(CN4CCN(CC(O)=O)CCN(CCN(CC4)CC(O)=O)CC(O)=O)=O)C=CC=C32	19.5
 eFAP-8	O=C(NCC(N1[C@H](C#N)CC(F)(F)C1)=O)C2=CC=NC3=C(OCCCNC(CCC(C(O)=O)N4CCC(CC(O)=O)CCN(CC(O)=O)CCN(CC4)CC(O)=O)=O)C=CC=C32	19
 eFAP-9	O=C(NCC(N1[C@H](C#N)CC(F)(F)C1)=O)C2=CC=NC3=C(OCCCNC(C4CCC(CNC(CN5CCN(CC(O)=O)CCN(CC(O)=O)CCN(CC(O)=O)CC5)=O)CC4)=O)C=CC=C32	21
 eFAP-12	O=C(NCC(N1[C@@H](C#N)CC(F)(F)C1)=O)C2=CC=NC3=C(OCCCNC(CN4CCN(C(CN5CCN(CC(O)=O)CCN(CC(O)=O)CCN(CC(O)=O)CC5)=O)CC4)=O)C=CC=C32	15

Figure 5: Molecular structure and score of FAP-alpha inhibitors.

Figures 4 and 5 present the molecular structures and corresponding scores of several patented in-house FAP-alpha inhibitors. These inhibitors, developed by the Erasmus Medical Centre, serve as important reference points for evaluating the performance of the ChemSpaceAL pipeline. The scores for these known inhibitors range from 10.5 to 21, providing a benchmark for assessing the quality of the AI-generated molecules. This range of scores for established inhibitors offers valuable context for interpreting the effectiveness of the molecules generated by ChemSpaceAL, allowing for a direct comparison between known effective compounds and the novel structures produced by the AI model.

6.5 FAP-alpha Results

Iteration	Average Score	Percentage of Scores Above 22	Maximum Score
al0	19.19	29.50%	38.50
al1	18.40	23.40%	39.00
al2	18.43	25.60%	36.00
al3	18.62	25.90%	38.50
al4	18.56	23.00%	36.50
al5	19.45	31.10%	44.00
al6	21.57	25.60%	46.00

Table 5: Evolution of FAP-alpha protein-ligand interaction scores for generated molecules

Table 5 presents the evolution of protein-ligand interaction scores for the FAP-alpha target across seven iterations of the ChemSpaceAL pipeline. The table displays three key metrics for each iteration: the average score, the percentage of scores above the threshold of 22, and the maximum score achieved. The data spans from the initial iteration (al0) to the final iteration (al6). The average scores range from 18.40 to 21.57, while the percentage of scores above the threshold varies between 23.00% and 31.10%. The maximum scores recorded across all iterations fall between 36.00 and 46.00.

To provide a comprehensive view of the molecular evolution throughout the active learning process, the top three scoring molecules for both FAP-alpha and TROP2 targets are presented for each iteration in Appendix B. This detailed record allows for a thorough examination of the pipeline's progression. In the main Results section, for clarity and conciseness, the focus lies on comparing only the top three scoring molecules from the first and last iterations. This comparison effectively demonstrates the overall improvement in molecular design achieved by the ChemSpaceAL pipeline, while the full iterative progression can be examined in appendix B for those interested in the step-by-step evolution.

Figure 6 and figure 7 provide a visual comparison of the top three scoring molecules generated for FAP-alpha in the initial (iteration 0) and final (iteration 6) rounds of the active learning process. The figure presents the 2D structures of these molecules, allowing for a direct visual assessment of their chemical features. Each molecule is accompanied by its respective score, enabling a quantitative comparison between the initial and final iterations. This visualization offers a qualitative glimpse into how the structural characteristics of the highest-scoring molecules evolved over the course of the active learning process. For a comprehensive view of the top-scoring molecules from all iterations, readers are directed to Appendix B, which contains the full progression of generated structures throughout the study.

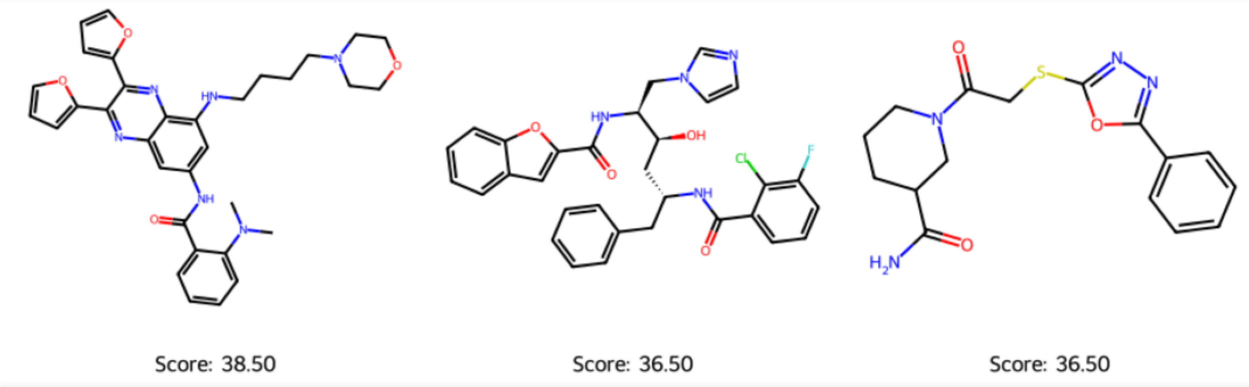


Figure 6: Top three scoring molecules for iteration zero for FAP-alpha

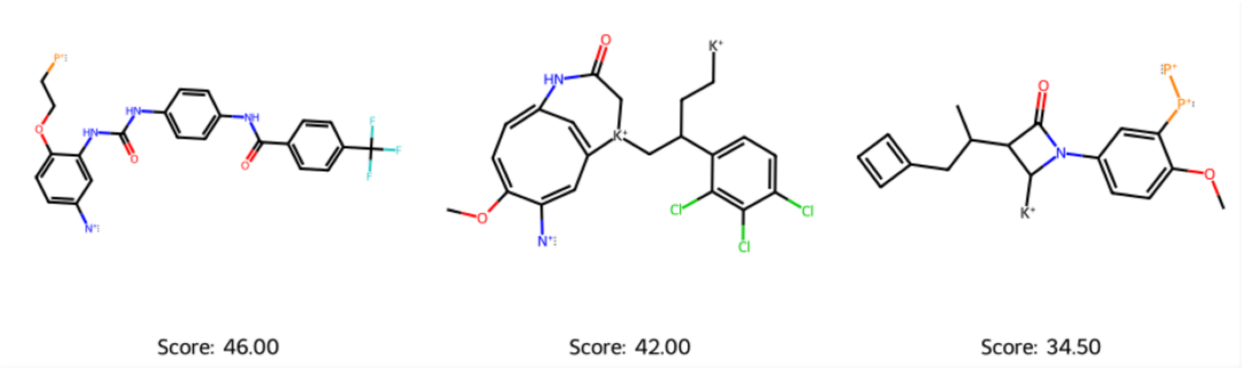


Figure 7: Top three scoring molecules for iteration six for FAP-alpha

Figure 6 displays the top three scoring molecules generated for FAP-alpha during iteration zero (al0) of the ChemSpaceAL pipeline. The figure presents the 2D structures of these molecules along with their respective scores. Figure 7 shows the top three scoring molecules for FAP-alpha from iteration six (al6), the final iteration of the pipeline. Similarly, this figure includes the 2D structures of the molecules and their corresponding scores. These figures provide a visual representation of the highest-scoring molecules at the beginning and end of the active learning process for the FAP-alpha target.

6.6 TROP2 Results

Iteration	Average Score	Percentage of Scores Above 22	Maximum Score
al0	17.06	18.1%	36.00
al1	16.81	13.5%	33.50
al2	16.99	14.8%	36.50
al3	18.56	23.0%	36.50
al4	19.81	31.3%	42.5

Table 6: Evolution of TROP2 protein-ligand interaction scores for generated molecules

Table 6 presents the evolution of protein-ligand interaction scores for the TROP2 target across five iterations of the ChemSpaceAL pipeline. The table displays three key metrics for each iteration: the average score, the percentage of scores above the threshold of 22, and the maximum score achieved. The data spans from the initial iteration (al0) to the fourth iteration (al4). The average scores range from 16.81 to 19.81, while the percentage of scores above the threshold varies between 13.5% and 31.3%. The maximum scores recorded across all iterations fall between 33.50 and 42.5.

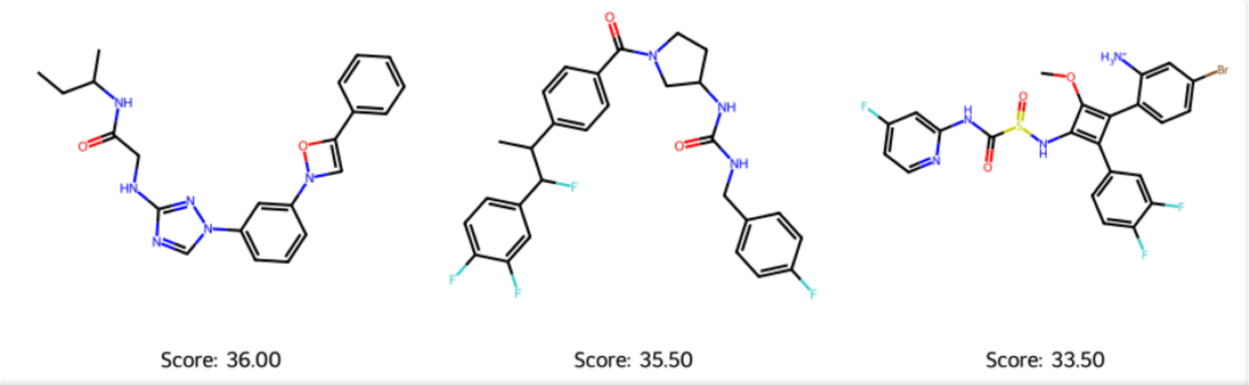


Figure 8: Top three scoring molecules for iteration one for TROP2

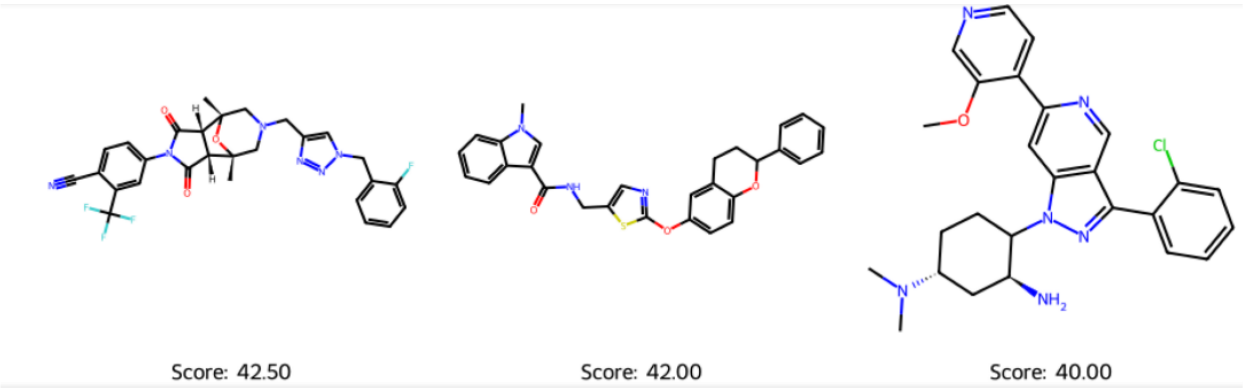


Figure 9: Top three scoring molecules for iteration four for TROP2

Figure 8 displays the top three scoring molecules generated for TROP2 during iteration one (al1) of the ChemSpaceAL pipeline. The figure presents the 2D structures of these molecules along with their respective scores. Figure 9 shows the top three scoring molecules for TROP2 from iteration four (al4), the final iteration of the pipeline for this target. Similarly, this figure includes the 2D structures of the molecules and their corresponding scores. These figures provide a visual representation of the highest-scoring molecules at an early stage and at the end of the active learning process for the TROP2 target.

7 Discussion

7.1 Implications of Pipeline Implementation Challenges

The challenges encountered during the implementation of the ChemSpaceAL pipeline in a Jupyter Notebook environment reveal important insights into the development of computational tools for drug discovery. The initial difficulty in familiarizing users with the pipeline underscores a common tension in scientific software development: balancing sophistication with user-friendliness. This experience highlights the need for more intuitive interfaces and comprehensive documentation in computational chemistry tools, particularly as the field moves towards interdisciplinary collaboration.

The issues with Google Colab, including the non-sequential execution of steps and unreliable resource allocation, point to broader challenges in cloud-based scientific computing. While platforms like Colab offer accessibility, their limitations in complex, resource-intensive tasks suggest that more robust, dedicated computing environments may be necessary for advanced drug discovery pipelines. This observation has implications for how research institutions approach computational infrastructure for drug discovery projects.

7.2 Significance of Adaptations and Improvements

The successful adaptation of the pipeline for Jupyter Notebook execution, particularly the transition to using the `subprocess` module, represents a significant step towards creating more portable and robust scientific software. This improvement not only enhances the pipeline's usability across different computing environments but also serves as a case study in software adaptation for the scientific community. The experience gained here could inform best practices in developing flexible, environment-agnostic computational tools in chemistry and related fields.

The attempt to parallelize the docking process, despite facing practical limitations, provides valuable insights into the complexities of optimizing computational workflows in drug discovery. The discrepancy between theoretical and actual performance gains highlights a critical aspect of high-performance computing in chemistry: the need to balance algorithm design with the realities of available hardware and shared resources. This experience suggests that future developments in computational chemistry might benefit from more adaptive algorithms that can dynamically adjust to available computational resources.

7.3 Interpretation of Parameter Tuning Challenges

The necessity for frequent manual interventions and adjustments of parameters like the scoring threshold reveals the complex, non-linear nature of the chemical space being explored. This observation has profound implications for the field of AI-driven drug discovery. It suggests that fully automated, one-size-fits-all approaches may be insufficient for navigating the intricate landscape of potential drug candidates. The variability in optimal parameters across iterations and possibly across different molecular targets indicates that successful AI in drug discovery may require more sophisticated, adaptive learning algorithms that can autonomously adjust their parameters based on the specific characteristics of the chemical space being explored.

7.4 Broader Impact

The challenges and insights gained from this implementation point to several critical areas for future research and development:

1. Adaptive Parameter Setting: Developing machine learning algorithms that can autonomously adjust parameters like scoring thresholds based on the evolving characteristics of the generated molecules could significantly enhance the efficiency and effectiveness of the drug discovery process.

2. Intelligent Automation: Creating more sophisticated automation processes that can make informed decisions about when human intervention is necessary could strike a balance between the benefits of automation and the need for expert oversight.

3. Resource-Aware Computing: Designing algorithms and workflows that can dynamically adapt to available computational resources could make advanced drug discovery techniques more accessible to a broader range of research institutions.

4. Interdisciplinary User Interfaces: Developing more intuitive interfaces that cater to chemists, biologists, and computational scientists alike could accelerate interdisciplinary collaboration in drug discovery.

These advancements could have far-reaching implications beyond just improving the ChemSpaceAL pipeline. They could contribute to making AI-driven drug discovery more efficient, accessible, and adaptable to diverse research questions and computational environments. Ultimately, this could accelerate the pace of drug discovery, potentially leading to faster development of new treatments for a wide range of diseases.

In conclusion, while the implementation of the ChemSpaceAL pipeline faced several challenges, these very challenges have illuminated critical areas for improvement in the field of computational drug discovery. By addressing these areas, future iterations of this and similar pipelines could play a transformative role in how we approach the complex task of identifying and developing new drug candidates.

7.5 The Evaluation of Generated Molecules Against FDA-Approved Inhibitors

In the original study of the ChemSpaceAL pipeline, the effectiveness of generated molecules for c-Abl kinase inhibition was primarily assessed through structural similarity comparisons with known FDA-approved small molecule inhibitors. However, this approach lacked a direct functional comparison. To address this limitation, a new analysis was conducted, where the same scoring function used in the ChemSpaceAL pipeline was applied to seven FDA-approved c-Abl kinase inhibitors.

The scoring results for these approved inhibitors were as follows: imatinib (62.5), bosutinib (44.5), asciminib (41), nilotinib (53.5), ponatinib (61), bafetinib (67.5), and dasatinib (41.5). These scores range from 41 to 67.5, with an average of 53.1, see table 2. In comparison, the original paper reported that after five iterations of the ChemSpaceAL pipeline, the generated molecules achieved an average score of 48.5 and a maximum score of 77.

This direct comparison provides several insights into the performance of the ChemSpaceAL pipeline. The maximum score of 77 achieved by the pipeline exceeds the highest score among the FDA-approved inhibitors (67.5 for bafetinib), suggesting that the method can potentially explore chemical space beyond known effective inhibitors and identify novel structures with enhanced binding characteristics. However, the lower average score of the generated molecules (48.5) compared to the average of the known inhibitors (53.1) indicates that while the pipeline can produce high-scoring outliers, it may not consistently generate molecules that match or exceed the quality of existing drugs.

This analysis underscores the importance of considering both peak performance and overall score distribution when evaluating generative models in drug discovery. While the ability to generate molecules with scores exceeding those of known drugs is promising, the goal should be to shift the entire distribution of generated molecules towards higher scores. Future iterations of the pipeline might focus on narrowing this gap, aiming to increase the average score of generated molecules while maintaining the ability to produce high-scoring outliers.

Furthermore, this comparison highlights the value of incorporating known drug scores as a benchmark in the evaluation process. Such comparisons provide a more nuanced understanding of the model's performance and its potential to contribute to drug discovery efforts. They also offer a clear target for future improvements: consistently generating molecules that score comparably to or better than existing drugs while potentially exploring novel chemical structures.

7.5.1 Analysis of HNH Domain of Cas9 Results

The application of the ChemSpaceAL pipeline to the HNH domain of Cas9 yielded intriguing results that both align with and diverge from those reported in the original ChemSpaceAL paper, see table 3. This implementation showed a gradual improvement in average scores from 8.21 in the initial iteration (al0) to 11.57 by the sixth iteration (al6), demonstrating the pipeline's ability to learn and generate increasingly promising molecules over time, see table 4. This trend is consistent with the original paper, which reported an increase from 7.9 to 11.5 over five iterations. However, the rate of improvement in this study appears to be slower, with the al5 results (average score 9.85) lagging behind the reported al5 results (average score 11.5).

Interestingly, while the average scores were generally lower, the maximum scores showed a different pattern. This implementation achieved higher maximum scores in later iterations, reaching 46.00 by al6, compared to the highest reported score of 34.0 in the original paper. This suggests that this version of the pipeline may be exploring a wider range of chemical space, potentially identifying more diverse and novel candidates, albeit at the cost of overall consistency in high scores.

The percentage of scores above the threshold of 11 also showed a notable difference. In this implementation, this percentage increased from 21.10% to 44.40% over six iterations, while the original paper reported an increase from 21.3% to 52.1% over five iterations. This discrepancy could indicate that this version of the pipeline is generating a more diverse set of molecules, including both high-scoring outliers and a larger proportion of lower-scoring compounds.

These differences in results could stem from various factors, including variations in the implementation details, differences in the scoring function, or the stochastic nature of the generative process. They underscore the importance of reproducibility studies in computational drug discovery and highlight the potential impact of subtle changes in methodology. Further investigation into these discrepancies could provide valuable insights into the robustness and sensitivity of the ChemSpaceAL approach, potentially leading to improvements in its application to challenging targets like the HNH domain of Cas9.

7.6 Analysis of FAP-alpha Results

The evaluation of potential FAP-alpha (Fibroblast Activation Protein-alpha) inhibitors using the ChemSpaceAL pipeline represents a significant step in the drug discovery process. To establish a baseline for comparison, a set of patented small molecule inhibitors from Erasmus MC were scored using the same methodology that would be applied to the generated molecules. This approach provides valuable context for interpreting the results of the AI-generated compounds.

The scoring of these known inhibitors yielded a range of values: 12.5, 17, 17, 10.5, 19.5, 19, 21, and 15, see figures 4 and 5. These scores reflect the predicted binding affinity or interaction strength between each molecule and the FAP-alpha target. The variation in scores among these known inhibitors is noteworthy, ranging from a low of 10.5 to a high of 21. This spread indicates that even among patented compounds, there is considerable variation in predicted effectiveness.

The highest score of 21, achieved by the seventh compound in the list, sets a benchmark for what might be considered a promising candidate in this scoring system. Conversely, the lowest score of 10.5 for the fourth compound suggests that even molecules with relatively low scores in this system might still possess inhibitory activity against FAP-alpha.

This range of scores for known inhibitors is crucial for several reasons:

- **Benchmark Setting:** It establishes a realistic range of scores that can be considered promising for potential FAP-alpha inhibitors. Molecules generated by ChemSpaceAL that score within or above this range could be considered particularly interesting for further investigation.

- **Validation of Scoring Method:** The fact that known inhibitors receive varied but generally positive scores lends credibility to the scoring method used in the ChemSpaceAL pipeline. It suggests that the scoring function is capable of identifying molecules with potential inhibitory activity.

- **Contextualizing Generated Results:** When evaluating the scores of AI-generated molecules, these reference scores provide a context for interpretation. For instance, a generated molecule scoring above 21 might be considered exceptionally promising, while those scoring below 10.5 might be viewed with more skepticism.

- **Understanding Scoring Sensitivity:** The range of scores among known inhibitors (from 10.5 to 21) indicates that the scoring function has a good dynamic range, capable of distinguishing between molecules with potentially different levels of activity.

- **Guiding Optimization Efforts:** For any AI-generated molecules, these reference scores can guide further optimization efforts. Researchers might aim to modify generated structures to push their scores towards or beyond the higher end of this reference range.

It's important to note that while these scores are predictive and useful for prioritizing compounds, they are not a direct measure of actual inhibitory activity. Factors such as solubility, cell permeability, and in vivo behavior are not captured by this scoring method. Therefore, while these scores are valuable for initial screening and prioritization, experimental validation remains crucial for confirming the actual effectiveness of any potential inhibitor.

In the context of the ChemSpaceAL pipeline, these reference scores serve as a valuable calibration tool. They allow researchers to gauge the potential of generated molecules relative to known, patented inhibitors. This comparison can help in quickly identifying the most promising candidates from among the vast number of molecules that an AI system like ChemSpaceAL can generate, streamlining the drug discovery process and potentially leading to the identification of novel, effective FAP-alpha inhibitors.

The results from the active learning iterations for FAP-alpha present an interesting pattern of performance. Initially, at iteration 0, the model achieved an average score of 19.19, with 29.50% of molecules scoring above the threshold of 22, and a maximum score of 38.5, see table 5. These baseline metrics are impressive, especially when compared to the in-house generated inhibitors, which scored between 10.5 and 21.

Subsequent iterations show a nuanced progression. Iteration 1 saw a slight decrease in performance, with the average score dropping to 18.4 and the percentage above threshold falling to 23.4%, though the maximum score increased slightly to 39. Iterations 2 and 3 showed minor improvements, with average scores of 18.43 and 18.62 respectively, and the percentage above threshold rising to 25.6% and 25.9%. However, iteration 4 maintained a similar average score of 18.56 but experienced a decrease in molecules above the threshold to 23%, with a maximum score of 36.5.

This pattern aligns with the previously discussed exploration-exploitation trade-off in machine learning. The initial high performance followed by a dip and then gradual improvement is typical of models adapting to specific targets. However, the relatively small fluctuations in later iterations suggest that the model might be reaching a plateau in its learning process.

The consistent generation of molecules with scores comparable to or exceeding the in-house inhibitors is noteworthy. However, the lack of substantial improvement over iterations, particularly in the average score and percentage above threshold, indicates potential overtraining. This could be due to using too many epochs during the fine-tuning process in each active learning cycle. Excessive epochs might cause the model to overfit to the high-scoring molecules from previous iterations, limiting its ability to explore new, potentially beneficial areas of the chemical space.

To address this, future experiments, particularly with the next target protein TROP2, will be conducted with a reduced number of epochs. This adjustment aims to strike a better balance between learning from high-scoring molecules and maintaining the model's ability to explore diverse chemical spaces, potentially leading to more significant improvements across iterations.

7.7 Analysis of the TROP2 Results

7.7.1 Adjusting Epoch Count for TROP2

In the initial iteration for TROP2, a reduced number of epochs was implemented during the active learning phase to mitigate potential overfitting, as previously discussed. However, this adjustment led to unexpected results. After 100 epochs, the model converged to a loss value exceeding 1, which was higher than anticipated. This suboptimal convergence had a detrimental effect on the subsequent iteration, resulting in the generation of SMILES strings of lower quality compared to the previous iteration. This observation highlights the delicate balance required in determining the optimal number of training epochs, emphasizing the need for careful tuning of hyperparameters in the active learning process.

The results for TROP2 highlight the challenges of applying this methodology to targets without known inhibitors. The initial iteration (al0) showed promising results with a mean score of 19.81 and 31.3% of molecules above the threshold. However, the subsequent iteration (al1) saw a decrease in performance, with the mean score dropping to 17.06 and only 18.1% of molecules above the threshold. This regression suggests that the initial threshold estimation and epoch count for TROP2 may have been suboptimal. The lower threshold likely led to the inclusion of less optimal molecules in the active learning set, while the reduced epoch count may have resulted in underfitting. These observations underscore the importance of careful parameter tuning, especially for novel targets. Future work should focus on developing more robust methods for threshold estimation in the absence of known inhibitors and on adaptive epoch adjustment strategies to balance between overfitting and underfitting risks.

7.7.2 TROP2 Results

Based on the results provided for the evolution of TROP2 small molecules, here's an interpretation for the discussion section of your thesis:

The evolution of small molecules targeting TROP2 over five iterations of the ChemSpaceAL pipeline reveals an interesting pattern of improvement, albeit with some fluctuations. Initially, in iteration 0 (al0), the model generated molecules with an average score of 17.06, with 18.1% of molecules scoring above the threshold of 22, and a maximum score of 36, see table 6. This baseline performance suggests that the pretrained model had some capacity to generate potentially binding molecules for TROP2, despite the lack of known small-molecule binders for this target.

Interestingly, iterations 1 and 2 (al1 and al2) showed a slight decline in performance, with average scores dipping to 16.81 and 16.99 respectively, and the percentage of molecules above the threshold decreasing to 13.5% and 14.8%. This initial decline is not uncommon in machine learning processes and may represent the model's exploration of the chemical space as it adapts to the specific requirements of TROP2 binding.

A turning point is observed in iteration 3 (al3), where we see a notable improvement across all metrics. The average score increased to 18.56, the percentage above threshold rose to 23.0%, and the maximum score reached 36.5. This improvement suggests that the model began to effectively learn from previous iterations, refining its understanding of the chemical features that contribute to TROP2 binding.

The final iteration (al4) demonstrates the most significant improvement, with the highest average score of 19.81, 31.3% of molecules above the threshold, and a maximum score of 42.5. This marked increase in performance indicates that the model has successfully learned to generate molecules with improved binding potential for TROP2.

The overall trend, particularly the substantial improvement in the later iterations, supports the effectiveness of the active learning approach employed by ChemSpaceAL. It suggests that the model is capable of progressively refining its understanding of the chemical space relevant to TROP2 binding, even in the absence of known small-molecule binders to guide the process.

However, it's important to note that while the improvement is promising, the scores are still relatively modest compared to those typically seen for targets with known binders. This underscores the challenging nature of generating small molecules for novel targets like TROP2. The fluctuations in performance across iterations also highlight the complexity of the task and the potential sensitivity of the model to the specific molecules selected for each active learning cycle.

These results provide valuable insights for future applications of ChemSpaceAL to novel targets. They suggest that multiple iterations are crucial for allowing the model to explore and then exploit the relevant chemical space. Furthermore, they indicate that patience is necessary when applying such methods to challenging targets, as initial iterations may not show immediate improvements.

Future work could focus on extending the number of iterations to see if further improvements can be achieved, as well as investigating the structural features of the highest-scoring molecules to gain insights into potential binding modes for TROP2. Additionally, experimental validation of the top-scoring molecules would be crucial to confirm the model's predictions and potentially identify the first small-molecule binders for TROP2.

7.8 Cross-Target Analysis of Learning Patterns

An intriguing pattern emerged across the experiments with different target molecules, including the HNH domain of Cas9 and FAP-alpha. It was observed that the initial iteration (al0) typically yielded higher average scores and a greater percentage of high-scoring molecules compared to the first active learning iteration (al1). This phenomenon, consistent across multiple targets, provides valuable insights into the learning dynamics of the ChemSpaceAL pipeline. The pretrained model, represented by al0, appears to generate a diverse set of molecules that broadly sample the chemical space, occasionally hitting upon high-scoring compounds by chance. As the active learning process begins in al1, a temporary dip in performance can be seen. This initial decline likely reflects the model's first attempts to adapt to the specific target, moving away from its general pretraining biases and exploring new, potentially fruitful areas of chemical space.

The subsequent iterations, from al2 onwards, show a consistent upward trend in scores, indicating that the model progressively refines its understanding of the target-specific chemical space. This pattern of initial decline followed by steady improvement demonstrates the model's capacity to balance exploration of new chemical territories with exploitation of promising leads. It showcases the robust nature of the active learning approach, which allows the model to overcome initial setbacks and ultimately generate molecules more tailored to each specific target. This consistent learning curve across different molecular targets not only validates the effectiveness of this methodology but also highlights the importance of running multiple iterations to fully harness the potential of the ChemSpaceAL pipeline in drug discovery efforts.

7.9 Methodological Adaptations and Their Implications

A significant deviation from the original ChemSpaceAL methodology emerged in the implementation regarding the number of epochs used for active learning. While the original paper reported using only 10 epochs for each active learning iteration, these experiments revealed that this was insufficient to achieve comparable loss reduction. It was found that 100-150 epochs were necessary to reach loss levels similar to those reported in the original study. This substantial increase in training time raises important considerations about the trade-offs between model performance and potential overfitting. The need for extended training could indicate differences in the dataset, model architecture, or implementation details. It might also suggest that the original 10-epoch approach, while efficient, may not fully capture the complexity of the chemical space for the specific targets. However, the increased number of epochs also introduces risks of overfitting to the active learning set, potentially limiting the model's ability to generalize and explore diverse chemical spaces in subsequent iterations.

This observation underscores the delicate balance required in fine-tuning deep learning models for molecular generation. To further investigate this discrepancy and assess the variability in model performance, multiple iteration 0's (the initial random sampling step) can be conducted. This approach would provide insight into the inherent variability of the process and help determine whether the need for additional epochs is consistent across different initializations. Such analysis could guide decisions on the optimal training regimen and potentially reveal insights into the robustness and reproducibility of the ChemSpaceAL methodology across diverse molecular targets.

The modification of the ChemSpaceAL methodology involved exploring the impact of varying the threshold for selecting target molecules in the active learning phase. The original paper used a fixed score threshold based on known inhibitors or database statistics, rather than a percentage-based cutoff. This approach, while grounded in empirical data, may not always adapt well to different protein targets or scoring functions. Experiments with using the top 10% of scored protein-ligand complexes to construct the active learning dataset resulted in only minor improvements in the average scores of the 1000 scored complexes in each iteration.

This limited progress suggested that the stringent 10% cutoff might be too restrictive, potentially excluding valuable near-miss candidates from the learning process. However, expanding the selection to the top 20% of scored complexes led to more substantial improvements. This enhanced performance can be attributed to the larger and more diverse active learning dataset, which provides the model with a richer set of examples to learn from. The broader selection likely captures a wider range of favorable interactions and structural features, allowing the model to better generalize and generate improved molecules in subsequent iterations. It may also help in capturing subtle patterns that contribute to high scores but are not immediately apparent in only the top-scoring complexes.

This finding highlights the importance of carefully tuning the selection criteria in active learning approaches for molecular generation, balancing the quality of selected examples with the size and diversity of the training set. The optimal threshold may vary depending on factors such as the complexity of the protein target, the nature of the scoring function, and the initial diversity of the generated molecules, emphasizing the need for adaptive strategies in molecular design workflows.

7.10 The Importance of Synthetic Feasibility in Molecular Design

Upon visual inspection of the molecular structures presented in Figures 6-9 for both FAP-alpha and TROP2 targets, it is challenging to discern clear structural differences between high-scoring and low-scoring molecules at first glance. The subtle variations in chemical structure that contribute to improved scores are not immediately apparent to the untrained eye. This observation underscores the complexity of structure-activity relationships in drug design and highlights the need for expert analysis. Specialized chemists, with their in-depth knowledge of medicinal chemistry and structure-function relationships, would be better equipped to identify and interpret the key structural features that contribute to improved binding scores. Their expertise is crucial for a more nuanced understanding of how molecular structures relate to predicted binding affinities and for guiding further optimization of these AI-generated molecules.

While the scoring function employed in the ChemSpaceAL pipeline provides valuable insights into potential binding affinity and drug-likeness, it does not account for the synthetic feasibility of the generated molecules. This is a critical limitation, as highly scored molecules may be challenging or even impossible to synthesize in practice, rendering them impractical for further development. Addressing this gap between computational design and real-world applicability is crucial for the effective translation of *in silico* results to viable drug candidates.

One proposed approach to tackle this issue is the incorporation of expert knowledge from medicinal chemists. A system could be implemented where a panel of experienced chemists evaluates the generated molecules through a majority voting process, categorizing each molecule as synthetically feasible or unfeasible. While this method introduces a degree of subjectivity and may not capture the nuanced complexities of organic synthesis, it provides a practical filter to prioritize molecules that are more likely to be synthesizable.

Alternatively, machine learning models trained on large datasets of known synthesizable compounds could be integrated into the pipeline to predict synthetic accessibility. However, such models would need to be regularly updated and validated to keep pace with advancing synthetic methodologies. Ultimately, balancing computational efficiency with synthetic realism remains a key challenge in AI-driven drug discovery, emphasizing the need for ongoing collaboration between computational scientists and synthetic chemists in refining and validating molecular design pipelines.

7.11 Future Improvements in User Interface and Accessibility

While the current implementation of ChemSpaceAL has demonstrated promising results in generating targeted molecules, it was recognized that its usability could be enhanced to make it more accessible to a broader range of researchers. Presently, the pipeline requires a certain level of programming proficiency, which may limit its adoption by researchers without extensive computational backgrounds.

A significant area for future improvement is the development of a graphical user interface (GUI). Such an interface would greatly enhance the user-friendliness of the tool, allowing researchers to interact with the pipeline more intuitively. A GUI could provide easy-to-use controls for adjusting parameters, initiating runs, and visualizing results. For instance, users could adjust clustering parameters, modify scoring thresholds, or fine-tune the active learning process through simple interface elements rather than modifying code directly.

Furthermore, the GUI could incorporate real-time visualization tools, allowing users to observe the evolution of generated molecules across iterations, view the distribution of molecular properties, or examine docking results visually. This would not only make the tool more accessible but also provide immediate feedback on the impact of parameter adjustments, facilitating a more intuitive understanding of the molecule generation process.

Another potential improvement could be the integration of pre-set configurations for common use cases or target proteins. This would allow less experienced users to quickly get started with the tool while still providing the flexibility for more advanced users to customize all aspects of the pipeline.

By focusing on these user-centric improvements, the aim was to broaden the accessibility of ChemSpaceAL, potentially accelerating its adoption in diverse research settings and enhancing its impact on drug discovery efforts. The development of such a user-friendly interface would represent a significant step towards democratizing AI-driven molecular design tools in the scientific community.

7.12 Proposed Approaches for GUI Implementation

In considering the development of a graphical user interface for ChemSpaceAL, two primary approaches emerge as promising solutions: a desktop application and a web-based interface. Each of these approaches offers distinct advantages and challenges, particularly in light of ChemSpaceAL's current structure as a Python package requiring local installation.

The desktop application approach, utilizing PyQt, presents a viable solution that aligns well with ChemSpaceAL's existing architecture. This method would involve creating a standalone executable that bundles Python, ChemSpaceAL, and all necessary dependencies. Tools such as PyInstaller or cx_Freeze could be employed to package these components into a single, easily installable application. This approach offers several advantages: it provides native performance for resource-intensive tasks, allows for offline usage, and eliminates the need for users to manually manage Python environments or packages.

In practice, the PyQt-based application would serve as an intuitive wrapper around ChemSpaceAL's core functionality. Users would interact with familiar GUI elements to set parameters, initiate computational runs, and analyze results. The application would handle the interaction with the ChemSpaceAL package behind the scenes, translating user inputs into appropriate function calls and presenting outputs in a visually comprehensible format.

Alternatively, a web-based application could be developed to provide access to ChemSpaceAL's capabilities through a browser interface. This approach would involve implementing ChemSpaceAL on a server, with a Flask or Django backend exposing its functionality through a RESTful API. A React-based frontend could then provide an interactive interface for users to input parameters, submit jobs, and visualize results. This solution offers the advantage of accessibility from any device with a web browser, without requiring local software installation.

However, the web-based approach presents challenges, particularly in terms of server-side resource management. ChemSpaceAL's computationally intensive processes would need to be handled on the server, potentially introducing latency for users and requiring careful resource allocation. Additionally, maintaining and scaling such a system would necessitate ongoing attention and resources.

The choice between these implementations will depend on a careful evaluation of the target user base, available resources, and long-term maintenance considerations. Regardless of the specific implementation chosen, the development of a GUI represents a critical step in enhancing the accessibility of ChemSpaceAL.

By providing intuitive interfaces for complex computational processes, a GUI would lower the barrier to entry for researchers without extensive programming backgrounds. This could potentially accelerate the adoption of ChemSpaceAL in diverse research settings, broadening its impact on drug discovery efforts. The integration of real-time visualization tools and pre-set configurations, as mentioned earlier, would further enhance the user experience and facilitate a more intuitive understanding of the molecule generation process.

In conclusion, the development of a graphical user interface for ChemSpaceAL, whether as a desktop or web-based application, represents a significant opportunity to democratize access to advanced molecular generation techniques. This enhancement has the potential to accelerate the pace of drug discovery across a broader spectrum of the scientific community, furthering the impact of AI-driven molecular design tools in pharmaceutical research.

8 Conclusion

8.1 Summary of Key Findings and Limitations

Based on the thesis content, here are concise answers to the research questions for your conclusion:

1. The choice of biomarker significantly affects ChemSpaceAL's performance in generating potential drug candidates:

- For c-Abl kinase, a well-studied target, the pipeline generated molecules with scores surpassing known FDA-approved inhibitors (maximum score of 77 vs. 67.5 for bafetinib). - For the HNH domain of Cas9, the pipeline showed gradual improvement over iterations, demonstrating adaptability to less-studied targets. - For FAP-alpha, AI-generated molecules produced comparable or better scores than known patented inhibitors. - For TROP2, a novel target without known small-molecule binders, the pipeline faced initial challenges but showed significant improvement over iterations, highlighting its potential for exploring uncharted chemical spaces.

2. The ChemSpaceAL model's performance varies across different biomarkers:

- For well-studied targets like c-Abl kinase, the model showed consistent improvement and generated high-scoring molecules. - For challenging targets like TROP2, the model's performance was less consistent initially but demonstrated significant improvement in later iterations. - The model's effectiveness appears to correlate with the availability of prior knowledge about the target, with well-studied targets yielding more immediate results.

3. Key factors influencing ChemSpaceAL's success in generating effective drug candidates include:

- Availability of known inhibitors for reference scoring and parameter tuning. - Complexity of the target protein's binding site. - Optimization of hyperparameters, particularly the number of epochs and scoring thresholds. - Balance between exploration of new chemical spaces and exploitation of promising leads. - Appropriate selection criteria for the active learning dataset. - Consideration of synthetic feasibility in the scoring function.

These findings underscore the versatility of ChemSpaceAL in addressing various biomarkers while highlighting the need for target-specific optimizations and the importance of iterative learning, especially for novel targets. The study also emphasizes the potential of this approach in exploring new chemical spaces for challenging targets in cancer research.

References

[1] Qiang Bai et al. Molaical: a soft tool for 3d drug design of protein targets by artificial intelligence and classical algorithm. *Briefings in Bioinformatics*, 22(3):161, 2021.

[2] Yichen Chen et al. Deep generative model for drug design from protein target sequence. *Journal of Cheminformatics*, 15(1), 2023.

[3] Joseph A DiMasi, Henry G Grabowski, and Ronald W Hansen. Innovation in the pharmaceutical industry: new estimates of r and d costs. *Journal of Health Economics*, 47:20–33, 2016.

[4] Roland A Frank and Richard Hargreaves. Clinical biomarkers in drug discovery and development. *Nature Reviews Drug Discovery*, 2(7):566–580, 2003.

[5] SR Krishnan et al. Accelerating de novo drug design against novel proteins using deep learning. *Journal of Chemical Information and Modeling*, 61(2):621–630, 2021.

[6] G W Kyro, Andrey Morgunov, R I Brent, and Victor S Batista. Chemspaceal: an efficient active learning methodology applied to protein-specific molecular generation. *PubMed*, 2023.

[7] Christopher A Lipinski. Lead- and drug-like compounds: the rule-of-five revolution. *Drug Discovery Today: Technologies*, 1(4):337–341, 2004.

[8] Tomohiro Masuda, Matthew Ragoza, and David R Koes. Generating 3d molecular structures conditional on a receptor binding site with deep generative models. *arXiv preprint arXiv:2010.14442*, 2020.

[9] David Merk et al. De novo design of bioactive small molecules by artificial intelligence. *Molecular Informatics*, 37(1–2):1700153, 2018.

[10] Marwin H Segler et al. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1):120–131, 2018.

[11] Mohan Srinivasarao, Cynthia V Galliford, and Philip S Low. Principles in the design of ligand-targeted cancer therapeutics and imaging agents. *Nature Reviews Drug Discovery*, 14(3):203–219, 2015.

[12] Oleg Trott and Arthur J Olson. Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31(2):455–461, 2010.

[13] Xinyi Wang, Cheng Chen, Ji Yan, Ying Xu, Daqing Pan, Li Wang, and Mengsu Yang. Druggability of targets for diagnostic radiopharmaceuticals. *ACS Pharmacology & Translational Science*, 6(8):1107–1119, 2023.

[14] Min Xu, Tao Ran, and Hao Chen. De novo molecule design through the molecular generative model conditioned by 3d information of protein binding sites. *Journal of Chemical Information and Modeling*, 61(7):3240–3254, 2021.

[15] Alex Zhavoronkov et al. Deep learning enables rapid identification of potent ddr1 kinase inhibitors. *Nature Biotechnology*, 37(9):1038–1040, 2019.

Appendix

A

For the latest version of this modified version of ChemSpaceAL please see the following github link and go to the 'scripts' folder: https://github.com/sonjakatz/ChemSpaceAL-slurm

B

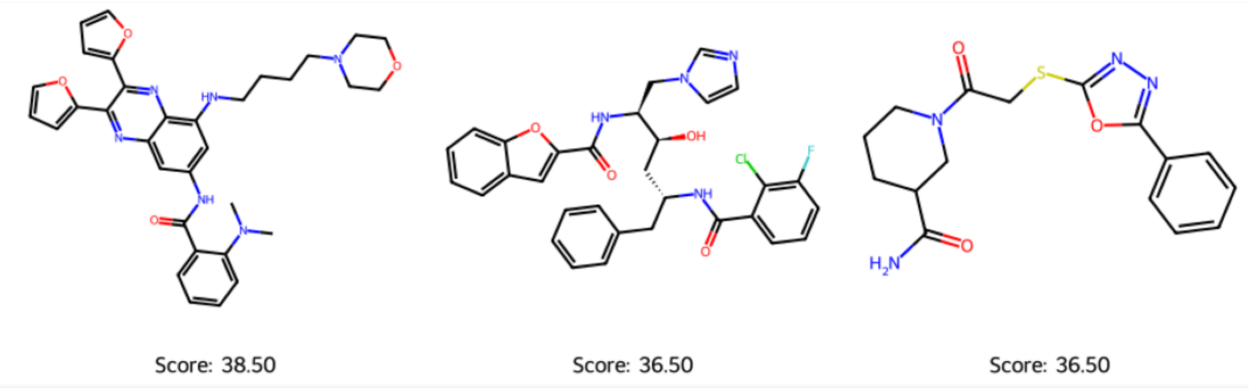


Figure 10: top 3 scoring molecules for FAP-alpha for iteration 0

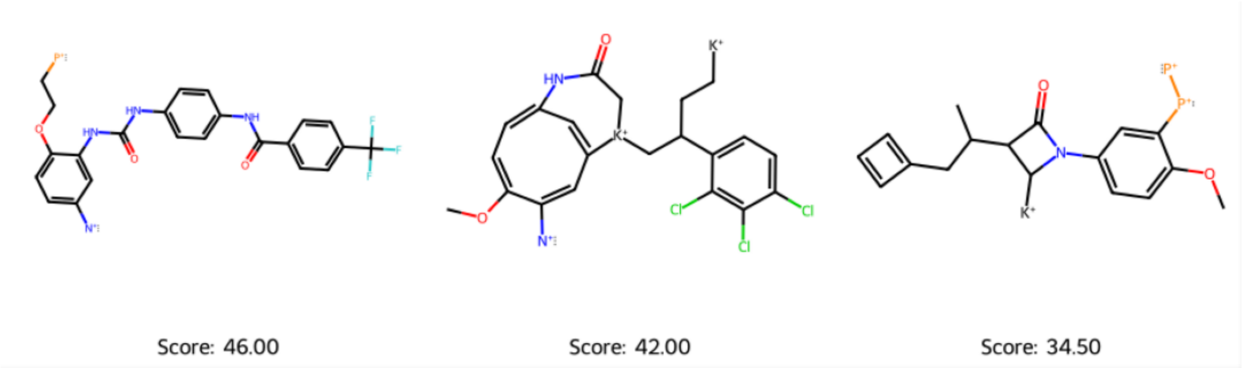


Figure 11: top 3 scoring molecules for FAP-alpha for iteration 1

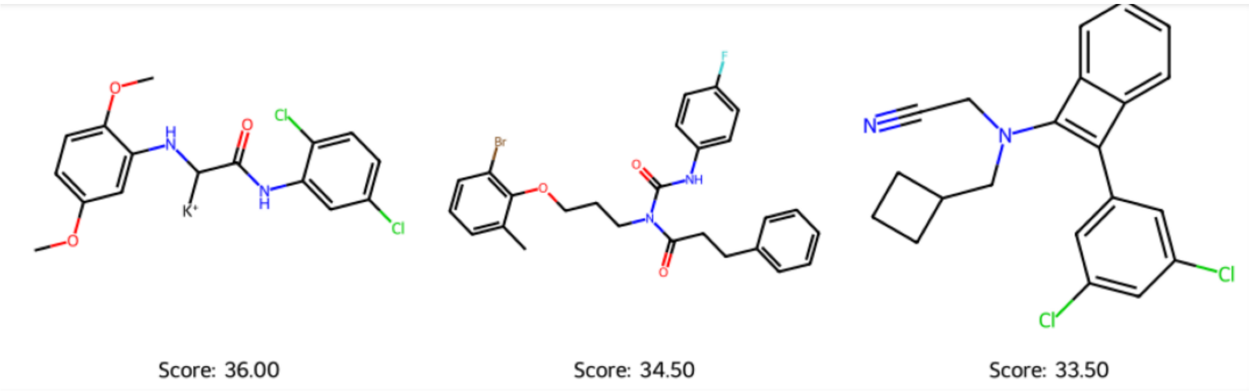


Figure 12: top 3 scoring molecules for FAP-alpha for iteration 2

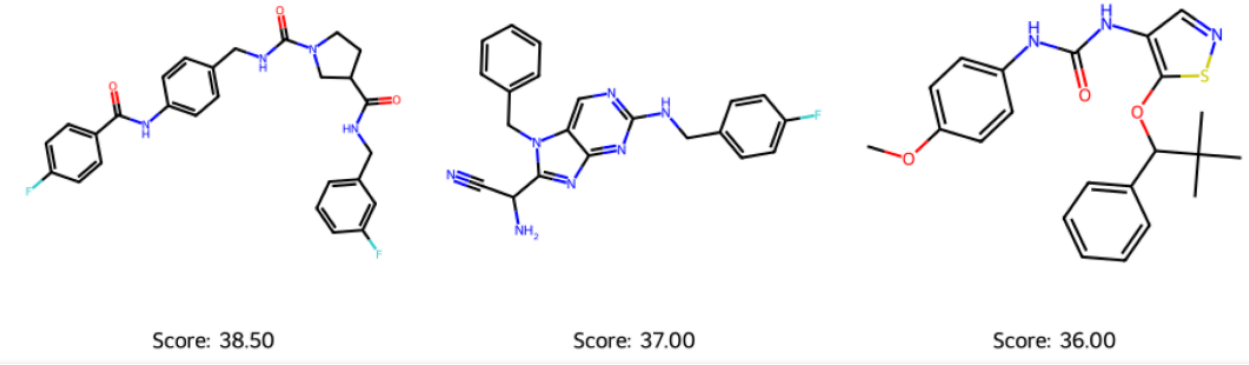


Figure 13: top 3 scoring molecules for FAP-alpha for iteration 3

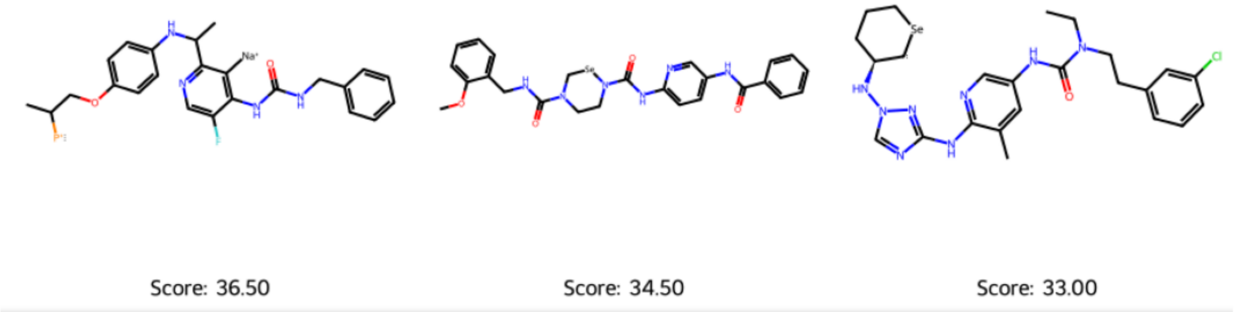


Figure 14: top 3 scoring molecules for FAP-alpha for iteration 4

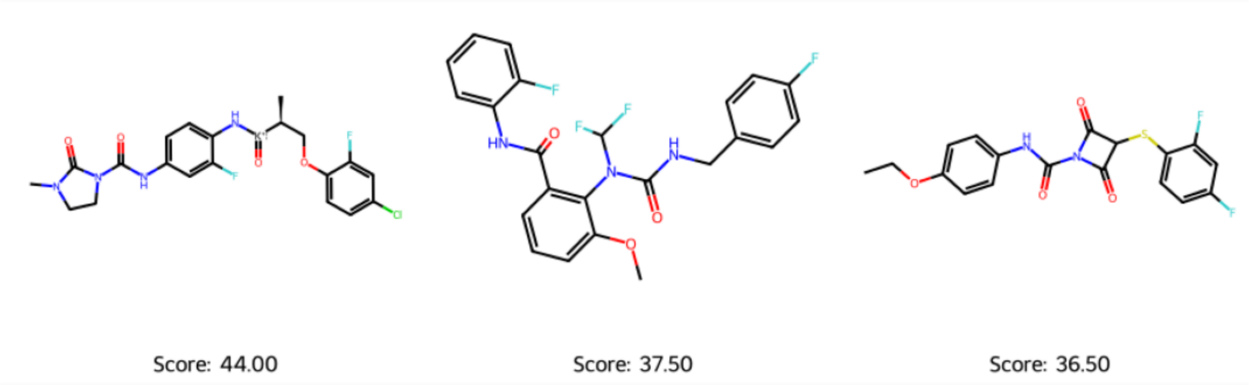


Figure 15: top 3 scoring molecules for FAP-alpha for iteration 5

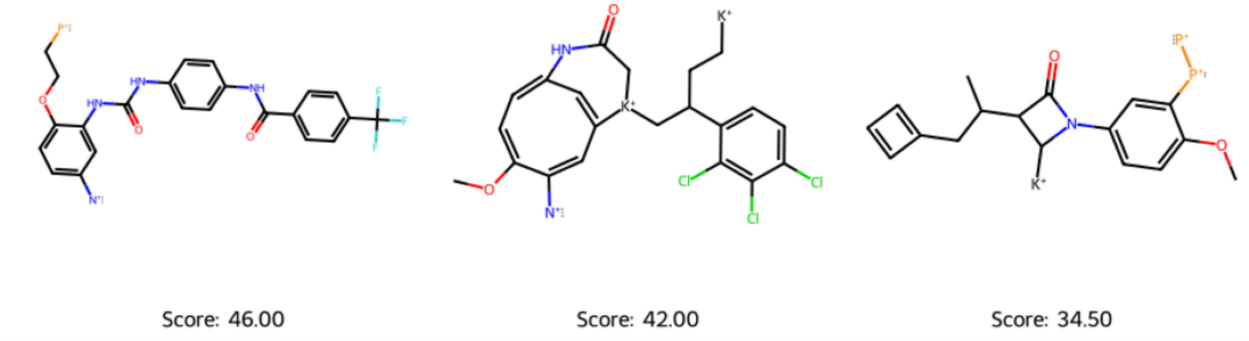


Figure 16: top 3 scoring molecules for FAP-alpha for iteration 6

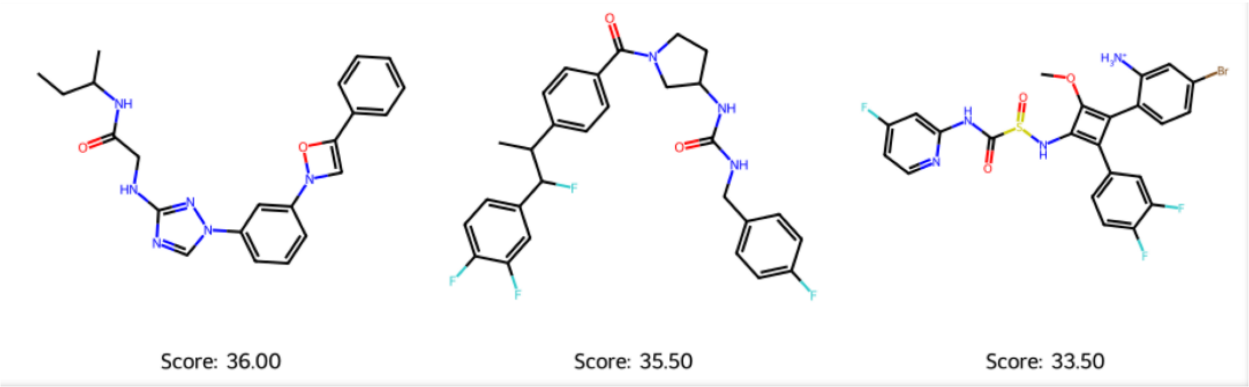


Figure 17: top 3 scoring molecules for TROP2 for iteration 0

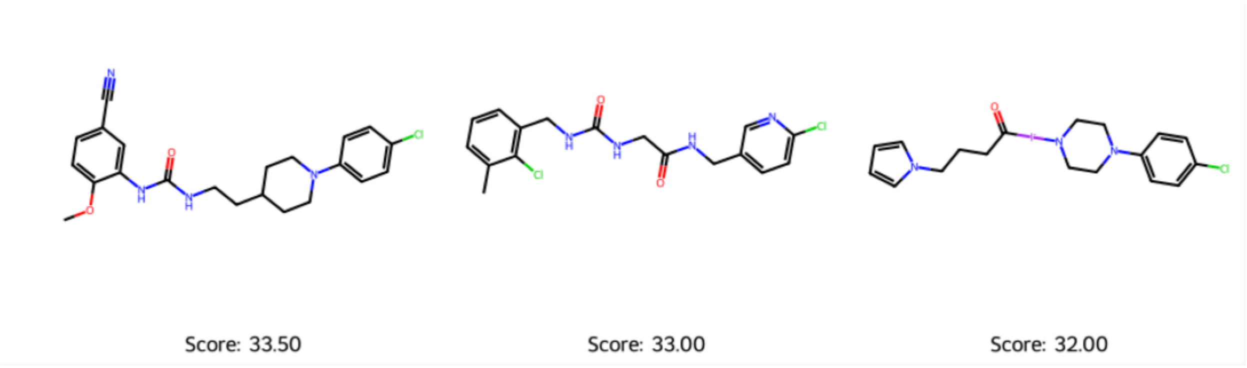


Figure 18: top 3 scoring molecules for TROP2 for iteration 1

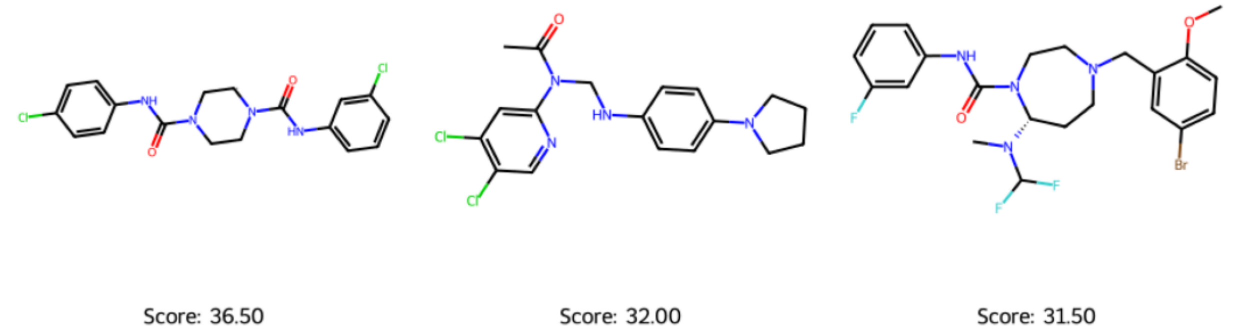


Figure 19: top 3 scoring molecules for TROP2 for iteration 2

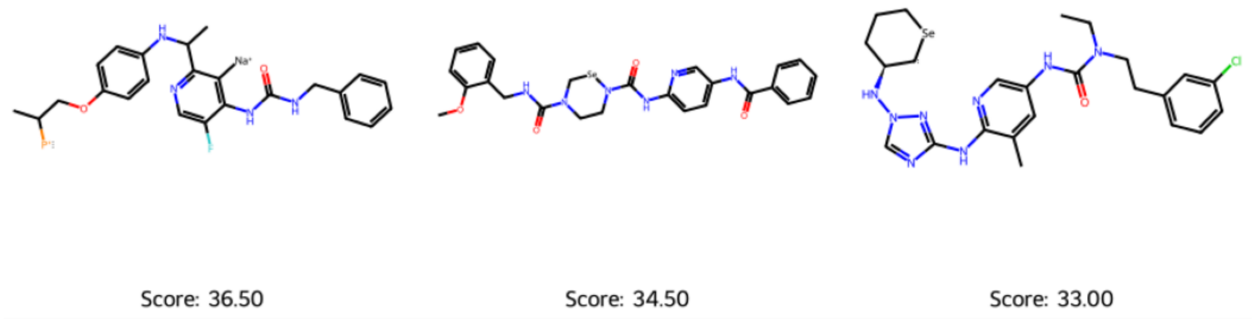


Figure 20: top 3 scoring molecules for FAP-alpha for iteration 3

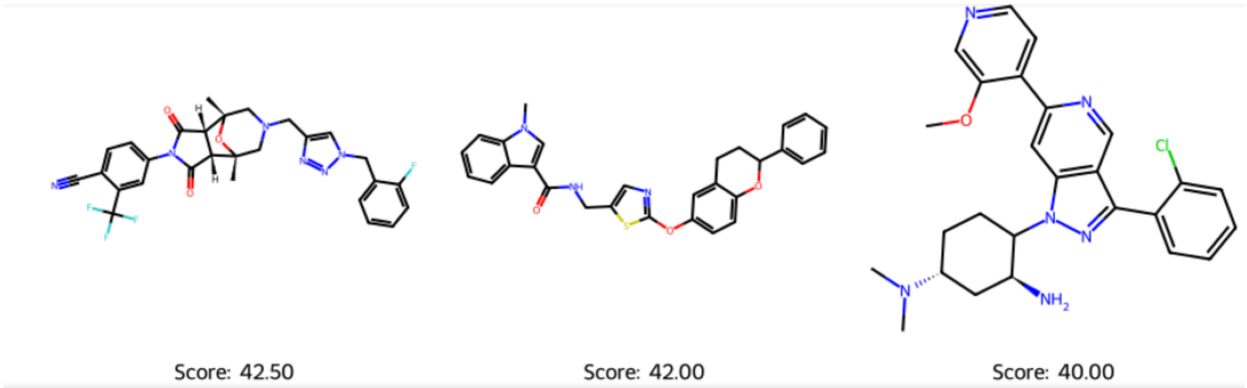
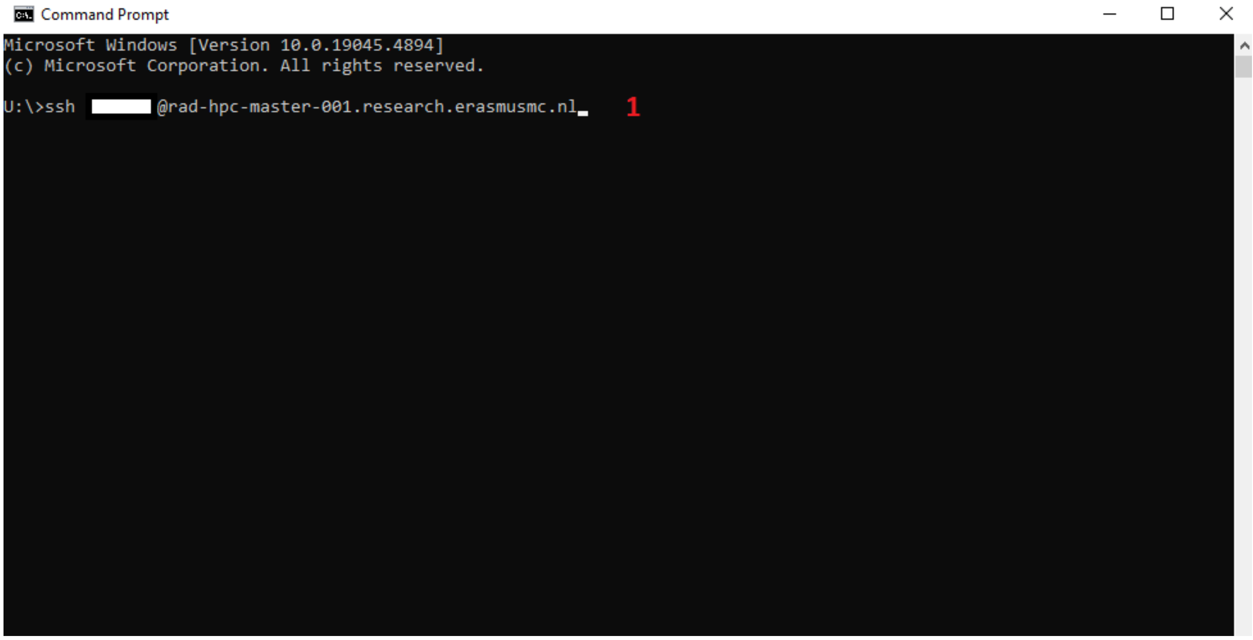


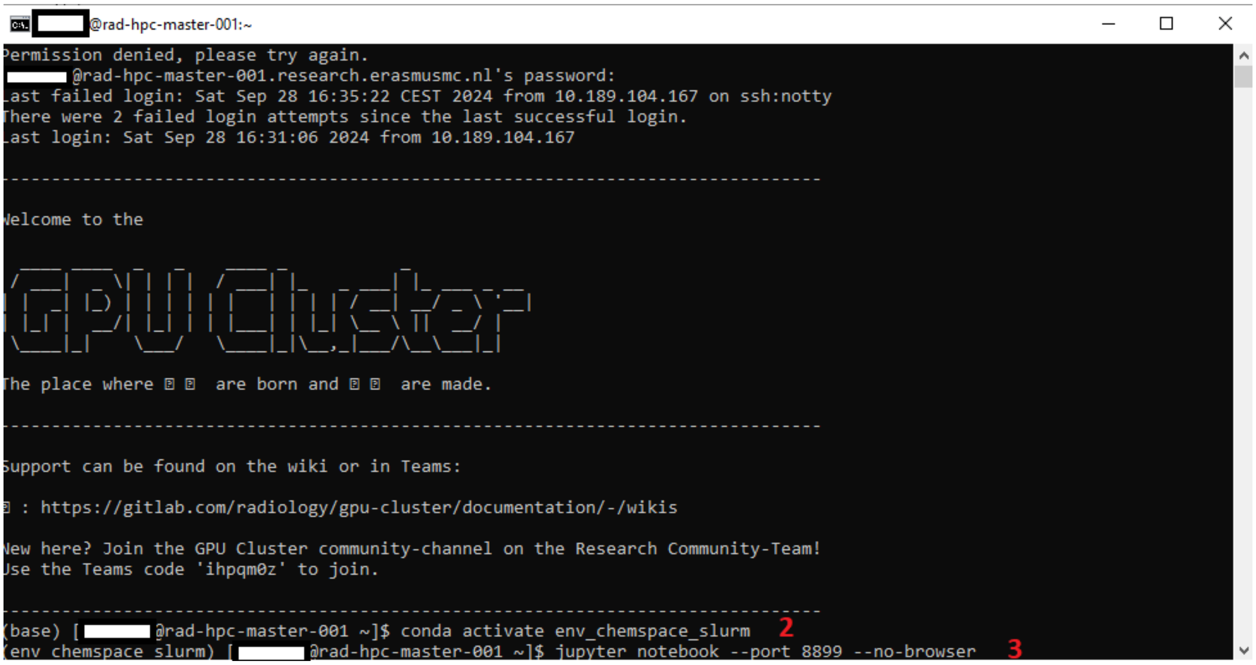
Figure 21: top 3 scoring molecules for FAP-alpha for iteration 4

C



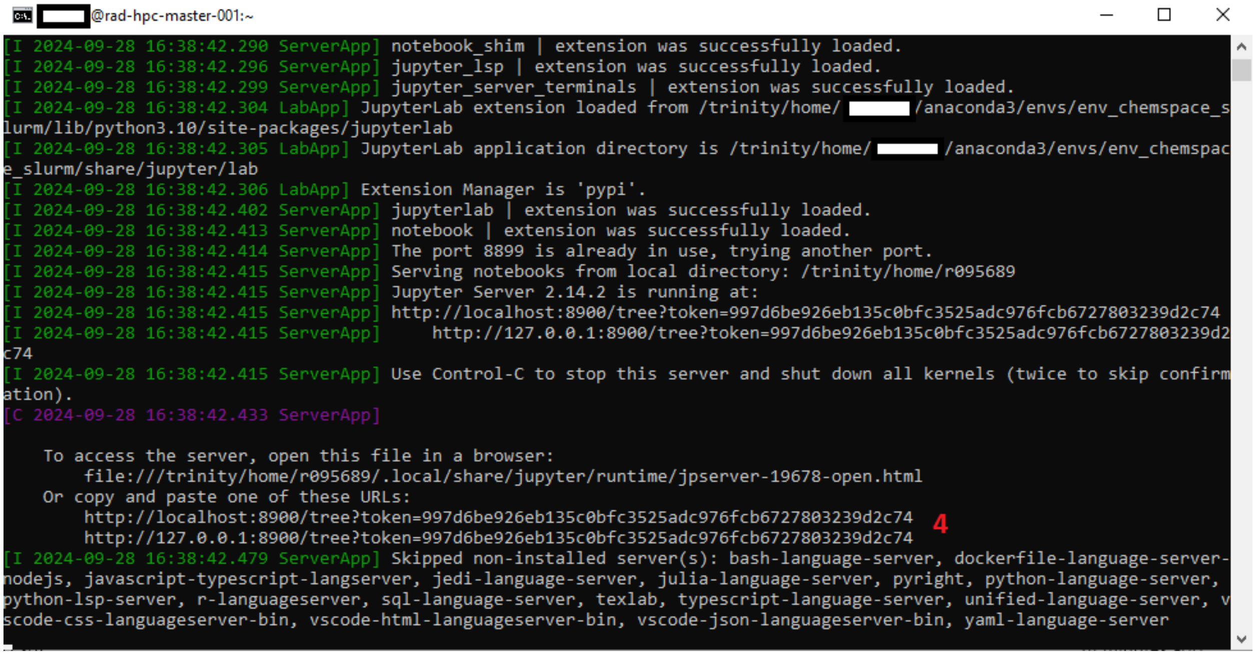
```
Microsoft Windows [Version 10.0.19045.4894]
(c) Microsoft Corporation. All rights reserved.

U:\>ssh [redacted]@rad-hpc-master-001.research.erasmusmc.nl
```

Figure 22: step 1: log on to the GPU cluster



```
Permission denied, please try again.
[redacted]@rad-hpc-master-001.research.erasmusmc.nl's password:
Last failed login: Sat Sep 28 16:35:22 CEST 2024 from 10.189.104.167 on ssh:notty
There were 2 failed login attempts since the last successful login.
Last login: Sat Sep 28 16:31:06 2024 from 10.189.104.167

--------------------------------------------------------------------------------

Welcome to the

  ____ ____  _   _    ____ _           _
 / ___|  _ \| | | |  / ___| |_   _ ___| |_ ___ _ __
| |  _| |_) | | | | | |   | | | | / __| __/ _ \ '__|
| |_| |  __/| |_| | | |___| | |_| \__ \ ||  __/ |
 \____|_|    \___/   \____|_|\__,_|___/\__\___|_|

The place where ⍰ ⍰  are born and ⍰ ⍰  are made.

--------------------------------------------------------------------------------

Support can be found on the wiki or in Teams:

⍰ : https://gitlab.com/radiology/gpu-cluster/documentation/-/wikis

New here? Join the GPU Cluster community-channel on the Research Community-Team!
Use the Teams code 'ihpqm0z' to join.

--------------------------------------------------------------------------------
(base) [[redacted]@rad-hpc-master-001 ~]$ conda activate env_chemspace_slurm
(env_chemspace_slurm) [[redacted]@rad-hpc-master-001 ~]$ jupyter notebook --port 8899 --no-browser
```

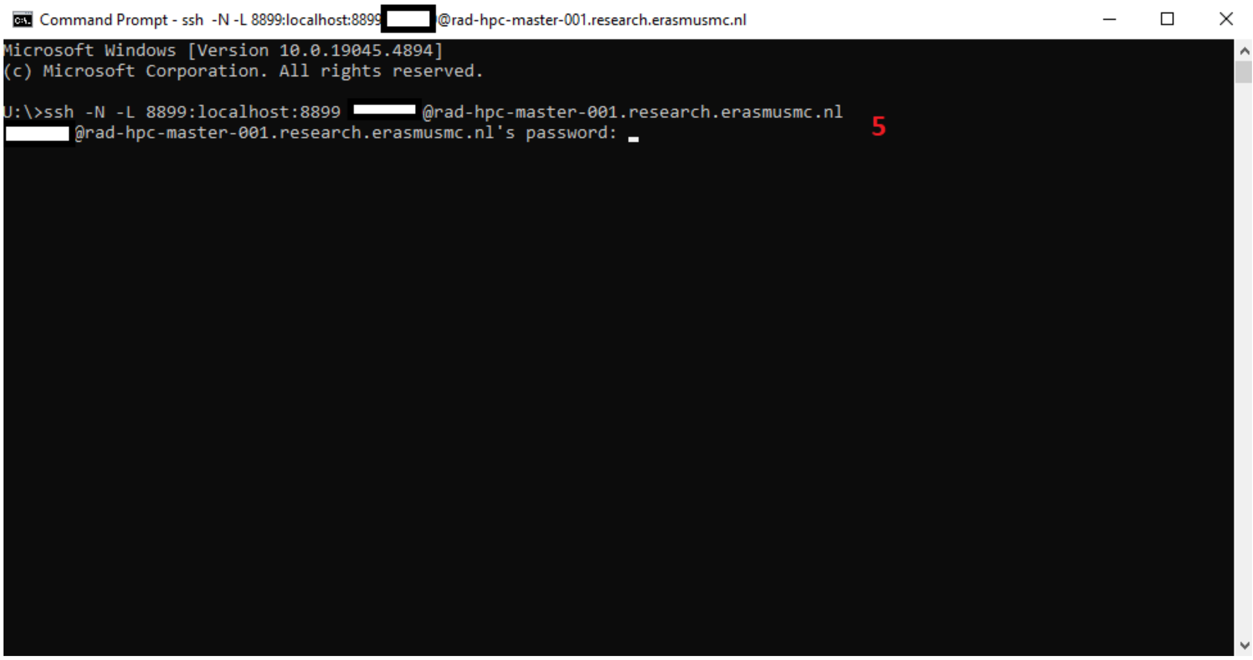
Figure 23: step 2: activate the environment and open jupyter notebook



[redacted]@rad-hpc-master-001:~

```
[I 2024-09-28 16:38:42.290 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-09-28 16:38:42.296 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-09-28 16:38:42.299 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-09-28 16:38:42.304 LabApp] JupyterLab extension loaded from /trinity/home/[redacted]/anaconda3/envs/env_chemspace_s
lurm/lib/python3.10/site-packages/jupyterlab
[I 2024-09-28 16:38:42.305 LabApp] JupyterLab application directory is /trinity/home/[redacted]/anaconda3/envs/env_chemspac
e_slurm/share/jupyter/lab
[I 2024-09-28 16:38:42.306 LabApp] Extension Manager is 'pypi'.
[I 2024-09-28 16:38:42.402 ServerApp] jupyterlab | extension was successfully loaded.
[I 2024-09-28 16:38:42.413 ServerApp] notebook | extension was successfully loaded.
[I 2024-09-28 16:38:42.414 ServerApp] The port 8899 is already in use, trying another port.
[I 2024-09-28 16:38:42.415 ServerApp] Serving notebooks from local directory: /trinity/home/r095689
[I 2024-09-28 16:38:42.415 ServerApp] Jupyter Server 2.14.2 is running at:
[I 2024-09-28 16:38:42.415 ServerApp] http://localhost:8900/tree?token=997d6be926eb135c0bfc3525adc976fcb6727803239d2c74
[I 2024-09-28 16:38:42.415 ServerApp]     http://127.0.0.1:8900/tree?token=997d6be926eb135c0bfc3525adc976fcb6727803239d2
c74
[I 2024-09-28 16:38:42.415 ServerApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirm
ation).
[C 2024-09-28 16:38:42.433 ServerApp]

    To access the server, open this file in a browser:
        file:///trinity/home/r095689/.local/share/jupyter/runtime/jpserver-19678-open.html
    Or copy and paste one of these URLs:
        http://localhost:8900/tree?token=997d6be926eb135c0bfc3525adc976fcb6727803239d2c74
        http://127.0.0.1:8900/tree?token=997d6be926eb135c0bfc3525adc976fcb6727803239d2c74
[I 2024-09-28 16:38:42.479 ServerApp] Skipped non-installed server(s): bash-language-server, dockerfile-language-server-
nodejs, javascript-typescript-langserver, jedi-language-server, julia-language-server, pyright, python-language-server,
python-lsp-server, r-languageserver, sql-language-server, texlab, typescript-language-server, unified-language-server, v
scode-css-languageserver-bin, vscode-html-languageserver-bin, vscode-json-languageserver-bin, yaml-language-server
```

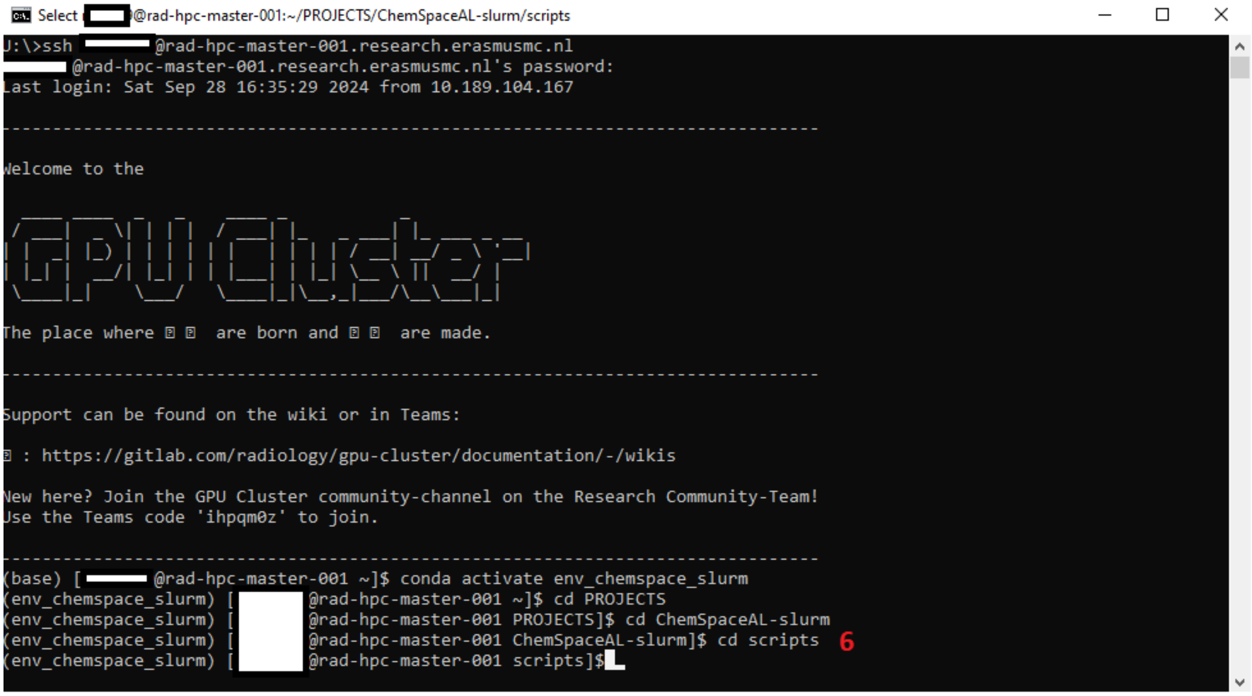
Figure 24: step 3: copy and paste the URLs



Command Prompt - ssh -N -L 8899:localhost:8899[redacted]@rad-hpc-master-001.research.erasmusmc.nl

```
Microsoft Windows [Version 10.0.19045.4894]
(c) Microsoft Corporation. All rights reserved.

U:\>ssh -N -L 8899:localhost:8899 [redacted]@rad-hpc-master-001.research.erasmusmc.nl
[redacted]@rad-hpc-master-001.research.erasmusmc.nl's password:
```

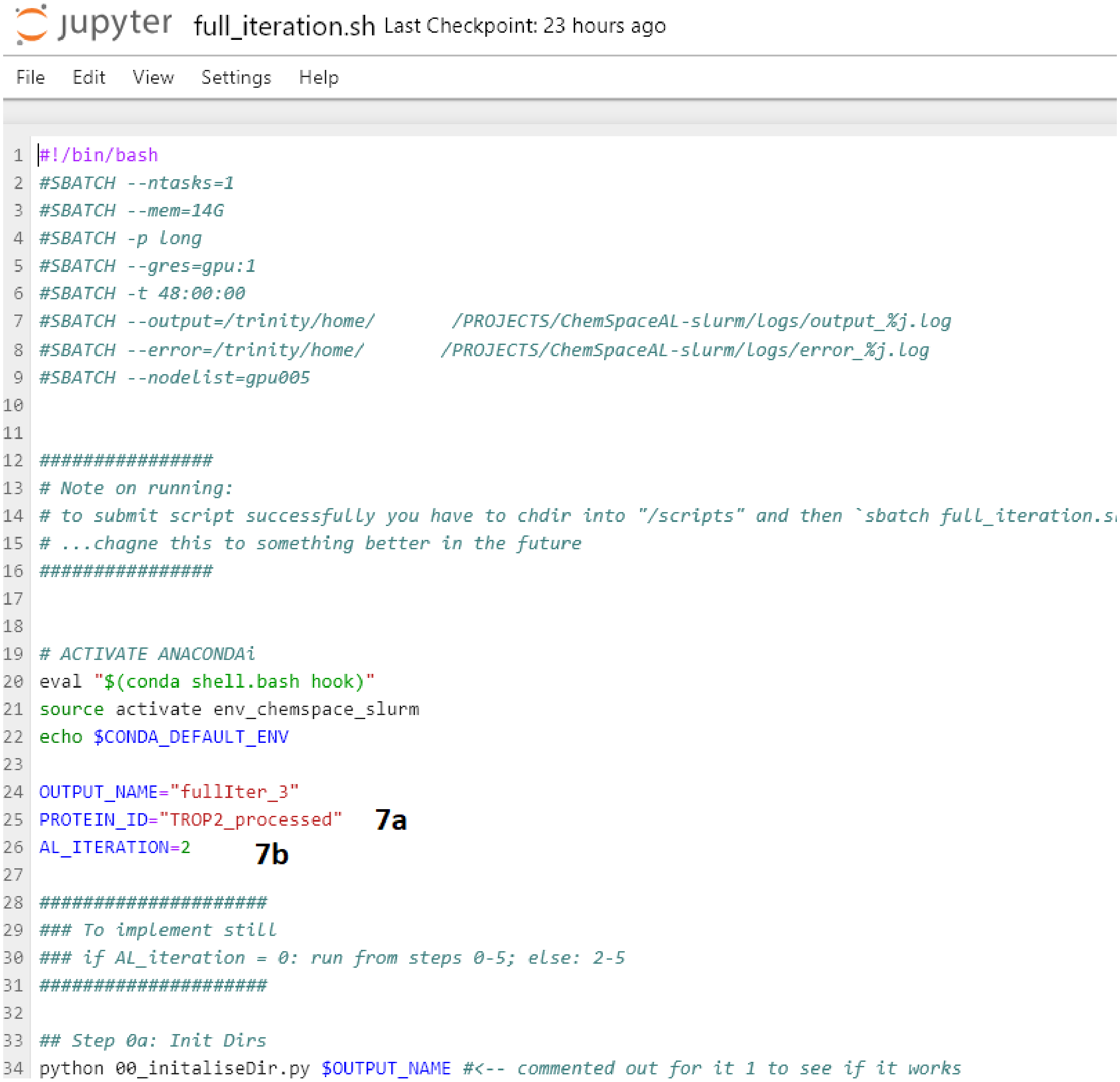
Figure 25: step 4: log on to the GPU cluster in a separate command window



Select [redacted]@rad-hpc-master-001:~/PROJECTS/ChemSpaceAL-slurm/scripts

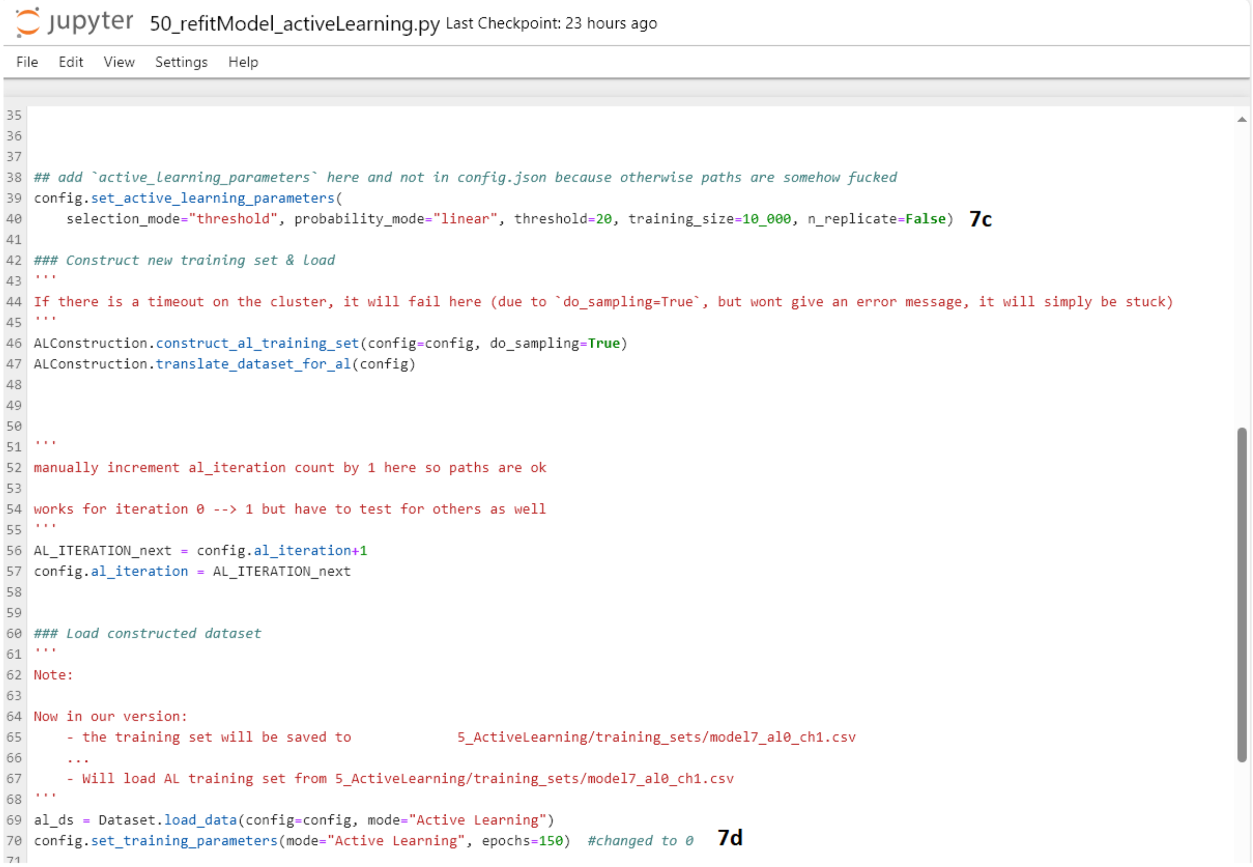
```
U:\>ssh [redacted]@rad-hpc-master-001.research.erasmusmc.nl
[redacted]@rad-hpc-master-001.research.erasmusmc.nl's password:
Last login: Sat Sep 28 16:35:29 2024 from 10.189.104.167

--------------------------------------------------------------------------------

Welcome to the

  ____ ____  _   _    ____ _           _
 / ___|  _ \| | | |  / ___| |_   _ ___| |_ ___ _ __
| |  _| |_) | | | | | |   | | | | / __| __/ _ \ '__|
| |_| |  __/| |_| | | |___| | |_| \__ \ ||  __/ |
 \____|_|    \___/   \____|_|\__,_|___/\__\___|_|

The place where ⍰ ⍰  are born and ⍰ ⍰  are made.

--------------------------------------------------------------------------------

Support can be found on the wiki or in Teams:

⍰ : https://gitlab.com/radiology/gpu-cluster/documentation/-/wikis

New here? Join the GPU Cluster community-channel on the Research Community-Team!
Use the Teams code 'ihpqm0z' to join.

--------------------------------------------------------------------------------
(base) [[redacted]@rad-hpc-master-001 ~]$ conda activate env_chemspace_slurm
(env_chemspace_slurm) [[redacted]@rad-hpc-master-001 ~]$ cd PROJECTS
(env_chemspace_slurm) [[redacted]@rad-hpc-master-001 PROJECTS]$ cd ChemSpaceAL-slurm
(env_chemspace_slurm) [[redacted]@rad-hpc-master-001 ChemSpaceAL-slurm]$ cd scripts
(env_chemspace_slurm) [[redacted]@rad-hpc-master-001 scripts]$
```

Figure 26: step 5: move to the correct directory



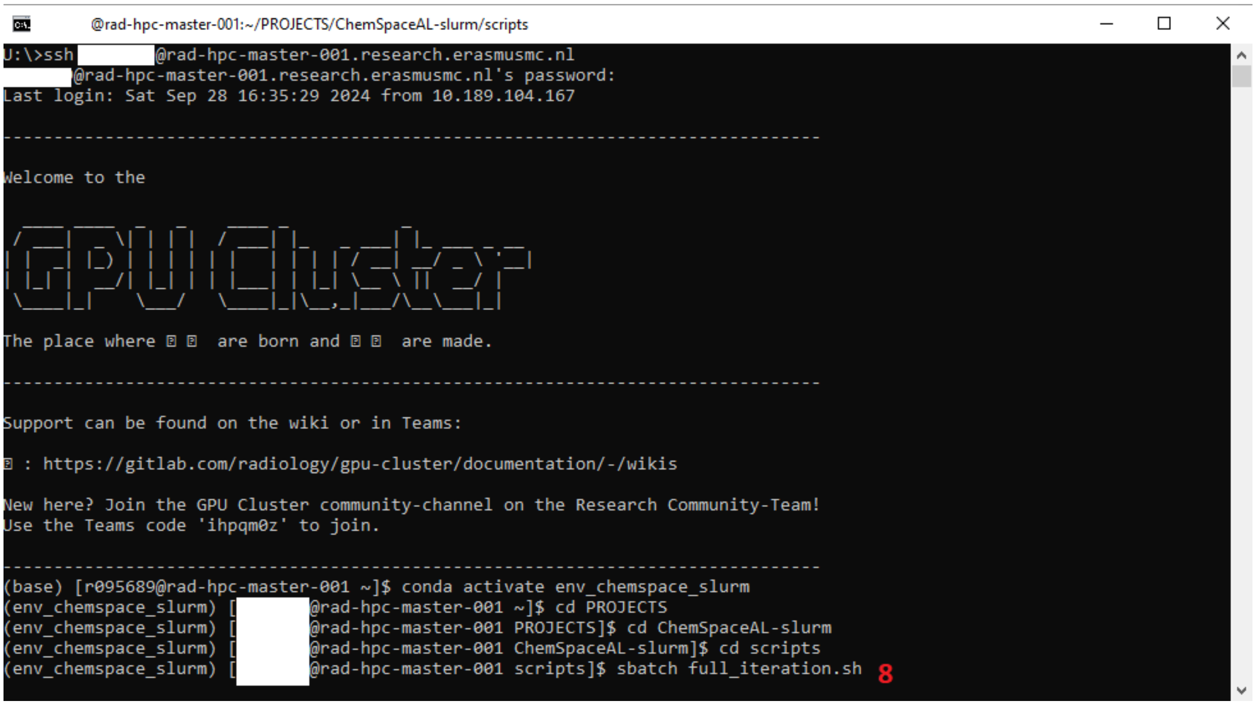
```
#!/bin/bash
#SBATCH --ntasks=1
#SBATCH --mem=14G
#SBATCH -p long
#SBATCH --gres=gpu:1
#SBATCH -t 48:00:00
#SBATCH --output=/trinity/home/       /PROJECTS/ChemSpaceAL-slurm/logs/output_%j.log
#SBATCH --error=/trinity/home/       /PROJECTS/ChemSpaceAL-slurm/logs/error_%j.log
#SBATCH --nodelist=gpu005


################
# Note on running:
# to submit script successfully you have to chdir into "/scripts" and then `sbatch full_iteration.s
# ...chagne this to something better in the future
################


# ACTIVATE ANACONDAi
eval "$(conda shell.bash hook)"
source activate env_chemspace_slurm
echo $CONDA_DEFAULT_ENV

OUTPUT_NAME="fullIter_3"
PROTEIN_ID="TROP2_processed"
AL_ITERATION=2

#####################
### To implement still
### if AL_iteration = 0: run from steps 0-5; else: 2-5
#####################

## Step 0a: Init Dirs
python 00_initaliseDir.py $OUTPUT_NAME #<-- commented out for it 1 to see if it works
```

Figure 27: step 6a: modify the iteration and target in 'AL_ITERATION' and 'PROTEIN_ID'


jupyter 50_refitModel_activeLearning.py Last Checkpoint: 23 hours ago
File
Edit
View
Settings
Help
7c
7d

```python



## add `active_learning_parameters` here and not in config.json because otherwise paths are somehow fucked
config.set_active_learning_parameters(
    selection_mode="threshold", probability_mode="linear", threshold=20, training_size=10_000, n_replicate=False)

### Construct new training set & Load
'''
If there is a timeout on the cluster, it will fail here (due to `do_sampling=True`, but wont give an error message, it will simply be stuck)
'''
ALConstruction.construct_al_training_set(config=config, do_sampling=True)
ALConstruction.translate_dataset_for_al(config)



'''
manually increment al_iteration count by 1 here so paths are ok

works for iteration 0 --> 1 but have to test for others as well
'''
AL_ITERATION_next = config.al_iteration+1
config.al_iteration = AL_ITERATION_next


### Load constructed dataset
'''
Note:

Now in our version:
    - the training set will be saved to             5_ActiveLearning/training_sets/model7_al0_ch1.csv
    ...
    - Will load AL training set from 5_ActiveLearning/training_sets/model7_al0_ch1.csv
'''
al_ds = Dataset.load_data(config=config, mode="Active Learning")
config.set_training_parameters(mode="Active Learning", epochs=150)  #changed to 0
```

Figure 28: step 6b: adjust parameters



```
U:\>ssh [redacted]@rad-hpc-master-001.research.erasmusmc.nl
[redacted]@rad-hpc-master-001.research.erasmusmc.nl's password:
Last login: Sat Sep 28 16:35:29 2024 from 10.189.104.167

--------------------------------------------------------------------------------

Welcome to the

  ____ ____  _   _    ____ _           _
 / ___|  _ \| | | |  / ___| |_   _ ___| |_ ___ _ __
| |  _| |_) | | | | | |   | | | | / __| __/ _ \ '__|
| |_| |  __/| |_| | | |___| | |_| \__ \ ||  __/ |
 \____|_|    \___/   \____|_|\__,_|___/\__\___|_|

The place where ⍰ ⍰  are born and ⍰ ⍰  are made.

--------------------------------------------------------------------------------

Support can be found on the wiki or in Teams:

⍰ : https://gitlab.com/radiology/gpu-cluster/documentation/-/wikis

New here? Join the GPU Cluster community-channel on the Research Community-Team!
Use the Teams code 'ihpqm0z' to join.

--------------------------------------------------------------------------------
(base) [r095689@rad-hpc-master-001 ~]$ conda activate env_chemspace_slurm
(env_chemspace_slurm) [[redacted]@rad-hpc-master-001 ~]$ cd PROJECTS
(env_chemspace_slurm) [[redacted]@rad-hpc-master-001 PROJECTS]$ cd ChemSpaceAL-slurm
(env_chemspace_slurm) [[redacted]@rad-hpc-master-001 ChemSpaceAL-slurm]$ cd scripts
(env_chemspace_slurm) [[redacted]@rad-hpc-master-001 scripts]$ sbatch full_iteration.sh
```

Figure 29: step 7: submit the script to the GPU cluster