UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

AGRONÓMICA, ALIMENTARIA Y DE BIOSISTEMAS

MÁSTER EN BIOLOGÍA COMPUTACIONAL

DEPARTAMENTO DE BIOTECNOLOGÍA - BIOLOGÍA VEGETAL (N)

Implementation of Ligand-based Virtual Screening workflow in Scipion and its validation.

TRABAJO FIN DE MÁSTER

Autor/a: Alba Lomas Redondo

Tutor/a: María Garrido Arandia
Cotutor/a: Carlos Óscar Sánchez Sorzano
Institución: Centro Nacional de Biotecnología (CNB)

Junio de 2022

AKNOWLEDGEMENT

I should mention in particular Daniel del Hoyo Gómez, PhD student in the Department of Macromolecule Structure at the CNB, who has been in charge of most of the supervision of this master's thesis.

I would also like to thank my UPM tutor María Garrido Arandia for her guidance and willingness to help me.

ABSTRACT

The process of developing new drugs has an impact on improving the quality of life for human beings, as it is focused on improving prognosis, survival and quality in life of patients. Drug development begins with the discovery of a new potential drug substance. The early stages of drug development have been digitalized through Computer-Aided Drug Design techniques (CADDS). The screening technique based on the structure of a known active ligand (LBVS) has been relegated to second place in favor of the approach based on the target molecule structure (SBVS), which a priori seems more attractive due to the use of 3D structures and the greater difficulty that the use of the former apparently presents. This study intends to bring together different LBVS process analyses within a flow management platform denominated Scipion in order to obtain a workflow applicable to the discovery of new drug targets. This integration into Scipion is expected to provide advantages to the user in terms of simplicity and manageability of data and the necessary bioinformatics tools. The work results indicate that the use of LBVS structure pre-processing is highly helpful for the further experimental procedure. Additionally, advantages of integrating these tools within Scipion are presented, as well as the improvement in some methods to obtain a fully automated process. For integration purposes, the Scipion platform is employed, with all its resources; and different open-source tools and packages, that have been adapted to the nature of the platform and to the needs presented by the variety of data typologies that can be included in this type of analysis. Once the protocols validation is performed, it can be concluded that the integration has been successful in most of them. However, some tools are still only indicative and require improvement in terms of code development. Also, there are clear advantages in this implementation both in terms of user manageability as well as in process automation.

TABLE OF CONTENTS

1. INTRODUCTION

Virtual drug screening has become very important in the discovery of new targets due to the reduction of the number of candidate molecules in huge databases [1] . The discovery of new drugs stalled a few decades ago because of the cost in time and money involved in the experimental approach, as it is estimated that it costs approximately 1 to 2 billion euros to develop a drug, and the process may require 10 to 15 years [2]. In this context, CADDs appeared in 1981 and its techniques have boosted Drug Design Virtual Screening [3].

Virtual Screening (VS) is one of the most widely used CADDs [4] and its objective is to find those molecules (known or to be synthesized) that have a high probability of showing biological activity on a target molecule, from an extensive database. The basic aim of these methods is to predict the nature and strength of binding of a given molecule to a target.

There are mainly two types of VS, Structure-Based Virtual Screening (SBVS) or direct drug design and Ligand-Based Virtual Screening (LBVS) or indirect drug design. The structure-based method is the approach of choice in cases where the 3D structure of the target protein has been experimentally characterized. However, the ligand-based approach is necessary when the 3D structure of the target does not exist or cannot be characterized [5].

This work focuses only on the second of the approaches, the LBVS. The main techniques used in this approach are different 2D chemical similarity analysis methods [6], searching molecules with a similar shape to known actives and screening using pharmacophores obtained by consensus pharmacophoric characteristics of several active ligands. A ligand-based screen ranges from a single analysis, such as filtering by pharmacokinetic characteristics, to a set of analyses that together constitute a workflow.

Here, a workflow divided into two stages is introduced. The first one focuses on the analysis of the chemical and structural characteristics of the screened molecules and includes the following steps: filtering by bioactivity, filtering by 2D descriptors (fingerprints), filtering by shape, analysis of ADME properties and filtering by PAINS. The second step is related to obtaining a consensus pharmacophore and screening the structures that pass the first part of the workflow with this pharmacophore.

In the last few years, different bioinformatics tools that perform the steps involved in ligand-based virtual screening have emerged [7-9]. For example, there are several software packages or libraries dedicated to the calculation molecular descriptors, such as ODDT [10] or RDKit [11], which are used to filter ADME properties or to calculate fingerprints. There are also tools and catalogues to detect structures potentially detrimental to the screening result, and to compare structural similarities between two molecules such as SwissSimilarity [12-13]. Other tools capable of obtaining pharmacophores from a set of ligands, such as PharmaGist [14], and tools for screening using the obtained pharmacophore, for example Pharmit [15], have also emerged. Finally, it is possible to automatically download datasets from well-known databases such as ChEMBL and PDB with preliminary filtering according to the preferences or needs of the project.

The proliferation of a wide variety of facilities for each of the above-mentioned analyses has both advantages and disadvantages: on the positive side, there is a wide variety of options for screening. On the negative side, there is a great difficulty for the researcher to study and learn how to handle each of the tools, each of these resources with their respective requirements and software installations. Together with the additional challenge of compatibilizing output files from one step to the following one, implies a considerable work and time cost.

One of the possible solutions is the integration of a set of tools, in a workflow shape approach, into a software management platform (Scipion) that facilitates the use of these programs through the implementation of graphical user interfaces (GUI). This solution is focused on achieving almost complete automation of a defined LBVS process within a tool capable of minimizing user requirements in the installation of the framework, in basic knowledge of chemoinformatics and in molecular archive formats.

The development functions used in most of the analyses come from the Python package RDKit and the source code for a significant number of workflow steps is adapted from the TeachOpenCadd project released by Volkamer Lab [16]. In addition, this work focuses on improving some protocol steps to increase filtering possibilities of the user and to substitute certain phases that usually have to be performed manually, which reduce the dynamism of the process.

The project concludes with the integration of the necessary protocols to execute a LBVS in Scipion and its subsequent validation by ensuring the same performance of all the tools both inside and outside the framework.

2. OBJECTIVES

The main objective of this work is to integrate a workflow based on LBVS for virtual drug screening into a platform/flow manager that allows the concatenation of different software in a single environment (Scipion) and its use worldwide. This workflow aims to integrate the most relevant LBVS filters/protocols and processes, as well as programmatic access to databases such as PDB or CHEMBL to automate the screening as much as possible. In addition, it implements tools to analyze the results of the different steps and improves some scripts of the standard workflow.

The report also discusses the validation of the different Scipion integrated protocols.

3. MATERIALS AND METHODS

3.1. FRAMEWORK

Scipion is an open-source project, developed at the National Centre of Biotechnology (CNB, CSIC) that can be downloaded from http://scipion.cnb.csic.es. Originally, it was a software framework for integrating several 3DEM (Electron Microscopy) software packages through a workflow-based approach, allowing the execution of reusable, standardized, traceable and reproducible image-processing protocols. These protocols incorporate tools from different programs while providing full interoperability among them and the implemented workflows can be modified to suit the user's needs [17].

These software packages are integrated into Scipion as plug-ins. Those that are open source are automatically downloaded with the platform (GROMACS [18]), while for those that require a license, for example Schrödinger, Scipion notifies the user how the program must be acquired and installed.

It is also possible to integrate tools in the form of scripts that are executed inside Scipion through conda environments. This is the context of the workflow presented in the paper.

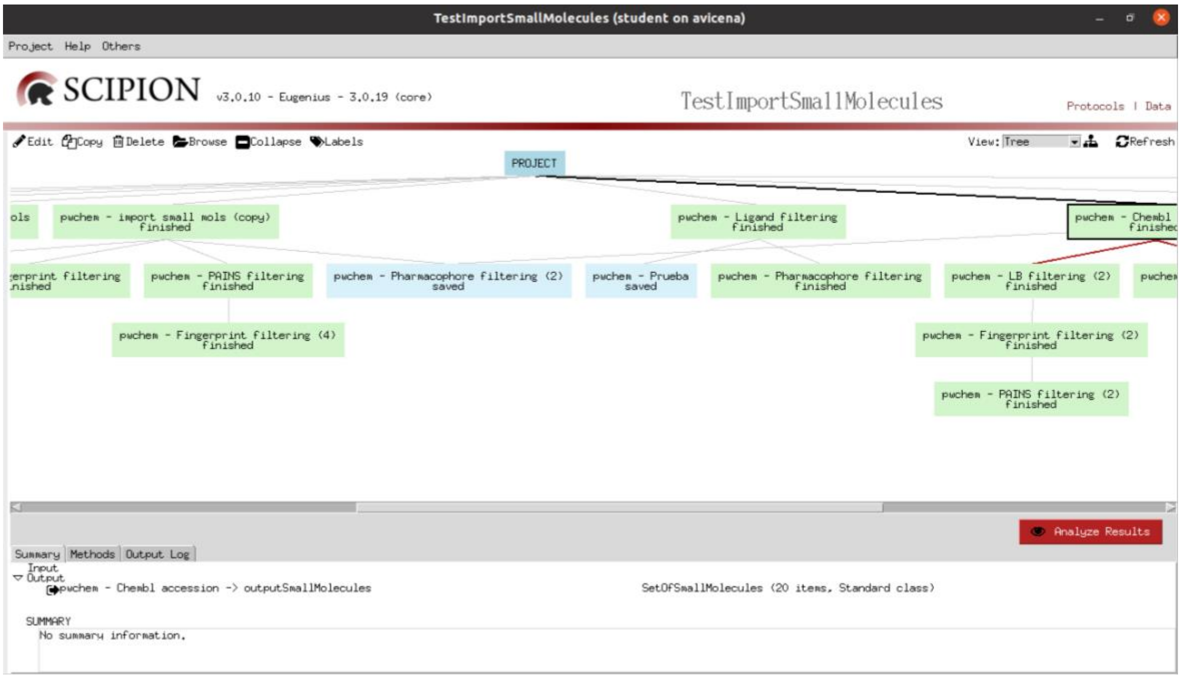
Scipion has grown in different branches: Scipion-em, for Electron Microscopy tools (https://github.com/scipion-em); and Scipion-chem, for CHEMoinformatics and virtual drug screening tools (https://github.com/scipion-chem). This last one houses the project of LBVS implementation.

Scipion-chem has the advantage of combining different structural biology softwares on a single platform, as well as presenting them in the form of protocols linked to a graphical interface. The arrangement of the tools in protocols makes them interchangeable and possibilities their reutilization. In addition, the framework records the steps and options chosen by the user for further study and reproduction. Finally, Scipion has implemented tools, in different protocols, with the same objective, giving the user a wide range of options to process their data and compare results.

3.2. PROTOCOLS IN SCIPION

A protocol is defined as a data processing task that involves the execution of several steps, which can range from calls to external programs (plug-in) to Python scripts that perform a series of tasks [19].

Each protocol requires a set of input Scipion objects and the setting of input parameters; and produces some other Scipion objects to store the results of their execution. Protocols can be linked to each other, forming a workflow (***Figure 1***), through Scipion objects, which contains different files and attributes for different instances, such as small molecules (SmallMolecule object) or proteins (AtomStruct object).


TestImportSmallMolecules (student on avicena)
Project Help Others
SCIPION v3.0.10 - Eugenius - 3.0.19 (core)
TestImportSmallMolecules
Protocols | Data
Edit Copy Delete Browse Collapse Labels
View: Tree
Refresh
PROJECT
ols
pwchem - import small mols (copy) finished
pwchem - Ligand filtering finished
pwchem - Chembl finished
erprint filtering nished
pwchem - PAINS filtering finished
pwchem - Pharmacophore filtering (2) saved
pwchem - Prueba saved
pwchem - Pharmacophore filtering finished
pwchem - LB filtering (2) finished
pwchem
pwchem - Fingerprint filtering (4) finished
pwchem - Fingerprint filtering (2) finished
pwchem - PAINS filtering (2) finished
Analyze Results
Summary Methods Output Log
Input
Output

```
pwchem - Chembl accession -> outputSmallMolecules        SetOfSmallMolecules (20 items, Standard class)

SUMMARY
  No summary information.
```

Figure 1. Example of a Scipion workflow.
Each box corresponds to a protocol, they are linked by lines forming different workflows in a hierarchical tree structure.

3.3. PROTOCOL CLASS STRUCTURE

Figure 2 shows the structure of a Protocol Class, which is divided into five main sections:

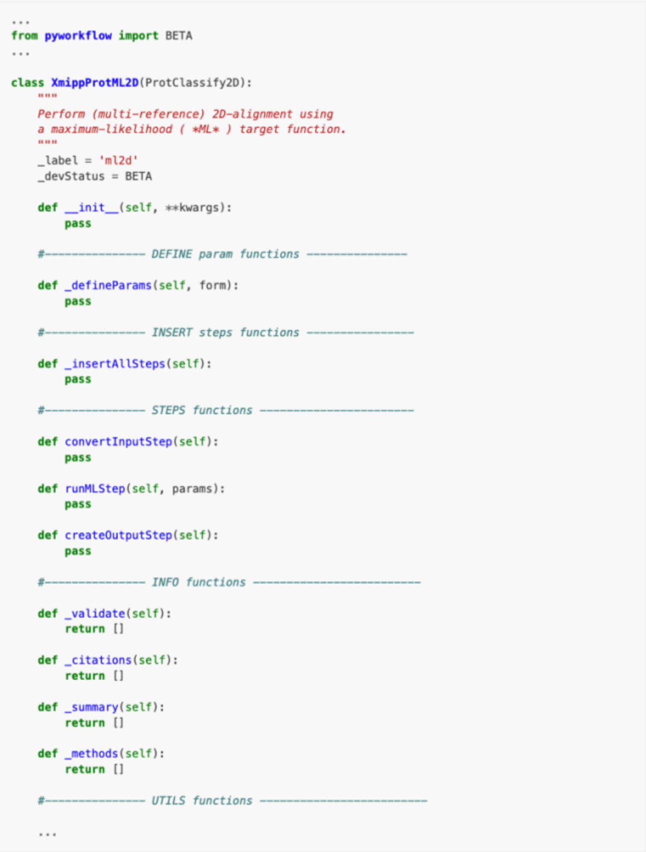
- **Parameter definition**: declaration of the GUI parameters which would be the attributes of the protocol.

- **Steps list:** choice of steps or functions that will conform objective tasks of the protocol.

- **Steps functions:** step functions are the most important part of the protocol as they contain the Python code which will be executed or a call to external programs and scripts.

- **Validation and info functions:** these functions are not necessary for the development of the protocol, but they provide very useful information for the user.
    a. Validate and Warning function: both are executed just before the protocol. They provide information about the possible errors that can appear in the protocol run. Validate function is a critical step since the protocol will not run if the stated errors there are not corrected.
    b. Citations function: the main objective of this function is to record the references (listed in bibtex format) of the protocol methods.
    c. Summary function: this function provides information about the current step of the protocol. In addition, when the protocol ends, this function shows information about the results, the quality, or any other relevant data.
    d. Methods function: it provides more descriptive information about the protocol execution than the Summary function, as it can be used for a *Materials and methods* section of a paper.

- **Other utils functions:** these are helper functions used throughout the code in the main functions and are protocol specific.



```
...
from pyworkflow import BETA
...

class XmippProtML2D(ProtClassify2D):
    """
    Perform (multi-reference) 2D-alignment using
    a maximum-likelihood ( *ML* ) target function.
    """
    _label = 'ml2d'
    _devStatus = BETA

    def __init__(self, **kwargs):
        pass

    #--------------- DEFINE param functions ---------------

    def _defineParams(self, form):
        pass

    #--------------- INSERT steps functions ----------------

    def _insertAllSteps(self):
        pass

    #--------------- STEPS functions -----------------------

    def convertInputStep(self):
        pass

    def runMLStep(self, params):
        pass

    def createOutputStep(self):
        pass

    #--------------- INFO functions ------------------------

    def _validate(self):
        return []

    def _citations(self):
        return []

    def _summary(self):
        return []

    def _methods(self):
        return []

    #--------------- UTILS functions -----------------------

    ...
```

Figure 2. Skeleton code of a Protocol Class in Scipion.
Reproduced from Creating a protocol — Scipion 3.0.0 documentation. 2022. Github.Io. Retrieved 21 June 2022, from https://scipion-em.github.io/docs/docs/developer/creating-a-protocol

The following sections (3.4. and 3.5.) describe the steps corresponding to the designed workflow in order of execution.

All the protocols and scripts created for this work are at **https://github.com/AlbaLomas/TFM-LBVS-Scipion.**

3.4. INTEGRATED PROTOCOLS – STRUCTURE-BASED FILTERING

3.4.1. ChEMBL programmatic access - Compound data acquisition

ChEMBL is a public database which contains information of a variety of potentially drug-derivable compounds, extracted manually from the existing literature. The database contains functional, pharmacokinetic (ADMET) and affinity data of the compounds. After manual data extraction, raw results are standardized and curated for quality assurance. ChEMBL already contains more than 5.4 million bioactivity measurements for more than one million compounds and 5200 protein targets [20].

The first workflow protocol aims to obtain an initial set of compounds that present bioactivity recorded in ChEMBL against a target molecule. It is therefore recommended for cases where the user does not have a starting set of ligands to analyze.

The result of the protocol is the import of the obtained molecules, in SMILES format [21-22], into a *SetOfSmallMolecules* Scipion object. This import is key so that the molecules can be used in the following protocols.

The protocol-linked script code is an adaptation of the code provided by the TeachOpenCadd project called Compound data acquisition (https://projects.volkamerlab.org/teachopencadd/talktorials/T001_query_chembl.html) and enables a broader search in the database.

Programmatic access is provided by a Python library called ChEMBL webresource client, which is the only official Python client library developed and supported by the ChEMBL group [23] and avoids the user requirement of studying how to interact with the REST APIs.

The protocol filters the compounds according to the following parameters [24]:

- **Uniprot ID**: Uniprot code of the target molecule.

- **Assay type**: ChEMBL includes six different types of assays, “biological assays are experimental methods for assessing the presence, localization, or biological activity of a substance in living cells and biological matrices” [25].

a. Binding (B): it is based on the binding of ligand molecules to receptors, antibodies, or other macromolecules. This type of assays obtain data for measuring this binding, e.g. Ki, IC50, Kd.

b. Functional (F): a set of systematic in vivo experiments designed to measure the effect or biological role of a compound in a cellular pathway or biological process, e.g., cell death in a cell line.

c. ADMET (A): assays related to the pharmacokinetic characteristics of a compound e.g., t1/2, oral bioavailability.

d. Toxicity (T): Data measuring toxicity of a compound, e.g., cytotoxicity.

e. Physicochemical (P): this type of assay is performed in the absence of biological material and measures physicochemical properties of the compounds e.g., chemical stability, solubility.

f. Unclassified (U): those assays which cannot be included in one of the above categories e.g., ratio of binding vs efficacy.

In case that the chosen assay type is Binding, which is recommended, the next parameter is the Bioactivity measure.

- **Bioactivity measure:**

  a. IC50: (Half maximal inhibitory concentration) indicates the required concentration of a compound to inhibit a biological process by 50%. The lower the IC50 value, the higher the compound bioactivity.

  b. EC50: (Half maximal effective concentration) is the drug concentration required to produce half (50%) of the compound maximum effect. The lower the EC50 value, the higher the compound bioactivity. EC50 is equivalent to IC50, but this last is used in the case of working with inhibitors.

  c. Kd: dissociation constant, this term is generic and describes the binding affinity between a molecule and an enzyme or a receptor.

  d. Ki: inhibition constant, the term is equivalent to the dissociation constant (Kd), but it is used when the molecule for which bioactivity is measured is an inhibitor.

- **Target type:** it is highly advisable that the chosen target type is "SINGLE PROTEIN", as the aim of the LVBS is to find compounds that have activity on a single target. Although it is possible to choose between other options.

- **Organism of compound origin:** it is recommended to choose human proteins; since the ultimate goal is to obtain possible molecules to be used as human drugs.

The user will specify the search according to these parameters using the GUI of the protocol.

When the assay type is a Binding assay, the script undertakes the following steps: firstly, it retrieves the ChEMBL ID of the compound with the highest affinity on the target. Then, it obtains the bioactivity data of the affine compound similar molecules and filters them according to assay type, bioactivity measure, target type and organism of compound origin fields.

Next, for those molecules that have passed the screening, their structure is obtained in form of "canonical SMILES". After collecting the bioactivity and structural data for each molecule, the filter is performed according to the average bioactivity chosen. The adapted script is able to sort the obtained molecules in bioactivity ascending order independently of the bioactivity measure. In case the chosen bioactivity measure is IC50 or EC50, they are transformed into pIC50 and pIEC50 values (negative logarithm in M) and ordered from the highest to the lowest. On the other hand, Kd and Ki values are ordered from the lowest to the highest. In most cases, the measure of bioactivity chosen is the IC50 since the drugs sought usually correspond to inhibitors [26]). It finally filters the molecules according to their bioactivity value.

In the rest of cases the search is limited to filtering without screening for bioactivity.

The result of the protocol is a number of compounds, also determined by the user, that show activity against the target and have the highest bioactivity levels.

The following protocols (ADME, fingerprint-based LBVS and shape-based LBVS) are based on the Similar Property Principle (SPP) and structure-activity relationship (SAR) which state that structurally similar molecules also show similar properties [27].

3.4.2. Analysis of ADME properties

Another protocol included in the workflow is the analysis of ADME properties. The objective is the qualitative evaluation of the screening compounds pharmacokinetic properties. These properties give a notion of whether the analyzed compounds can fulfil their function as a drug when they are ingested by humans. This analysis is done using an empirical rule known as Lipinski's rule of five (Ro5) [28] as it was developed to estimate the bioavailability of a compound only based on its chemical structure. The most important pharmacokinetic properties (in the human body) are Absorption, Distribution, Metabolism and Excretion (ADME).

This filtering is particularly useful for testing compounds whose pharmacological properties/activity have already been proven.

Ro5 dictates that for a compound to be orally administered it can only violate one of the Lipinski rules, there are variants of the standard that include new rules that do not require the complete fulfillment of them by the compound. However, in this paper only the original Ro5 rules will be considered. The compound must:

- Contain no more than 5 hydrogen bond donors (the total number of nitrogen–hydrogen and oxygen–hydrogen bonds).

- Contain no more than 10 hydrogen bond acceptors (all nitrogen or oxygen atoms).

- Have a molecular mass between 160 and 500 Da, as large molecules (high molecular weight) might have more difficulties in passing phospholipid membranes.

- Not exceed 5 in its octanol-water partition coefficient (log P). This coefficient measures the distribution of a compound, usually between a hydrophobic (e.g., 1-octanol) and a hydrophilic (e.g., water) phase.

The protocol's script is an adaptation of the source code, provided by the TeachOpenCadd project called Molecular filtering, ADME and lead-likeness criteria (https://projects.volkamerlab.org/teachopencadd/talktorials/T002_compound_adme.html) and another source code form oddt github (https://oddt.readthedocs.io/en/latest/_modules/oddt/virtualscreening.html). They offer the user the possibility of filtering a set of compounds by the rule Lipinski's rule and the Rule of 3 (Ro3). This new rule has the same statements as the Ro5, the difference between them is that, in the case of Ro3, the values that define whether a compound satisfies the rule are multiples of 3 and not of 5.

The protocol accepts as input a *SetOfSmallMolecules* Scipion object and the result of the process are those molecules that have successfully passed the chosen filter in a new *SetOfSmallMolecules* object.

3.4.3. Compound similarity - Fingerprint-based LBVS

Molecular descriptors are defined as mathematical representations of molecular properties, and they are used to characterise molecular physicochemical information quantitatively [29].

There are different types of molecular descriptors:

- **1D descriptors**: one-dimensional or global descriptors, i.e., for a single molecule, they have a single value. Solubility or molecular weight are examples of 1D descriptors.

- **2D descriptors**: two-dimensional descriptors are based on the structure of the molecular formula, these include molecular graphs, trajectories, fragments, atomic environments, or fingerprints. Fingerprints are computational representations of molecules encoding chemical and molecular features in the form of 0 (absence of the feature) and 1 (presence of the feature).

- **3D descriptors**: 3D descriptors are based on the three-dimensional structure of the molecule. They are less robust than 2D representations because of molecular flexibility, as the true shape of a molecule cannot be fully established. 3D descriptors include PPP patterns (Potential Pharmacophore Points), distance matrices (containing all distances between atoms) and description of three-dimensional fields (e.g., steric or electrostatic).

In previous protocols, 1D descriptors are analyzed, e.g., in the filtering of ADME properties. 2D descriptors filters are also widely used in virtual screening and mapping chemical space. In this case, the workflow analyses the similarity of two molecules using their fingerprints since "Molecular fingerprints are essential cheminformatics tools for virtual screening and mapping chemical space" [30].

The protocol offers the user the option of filtering by two types of fingerprints:

- **MACCS** (Molecular ACCess System): these fingerprints consist of 166 bits representing predefined structural fragments (***Figure 3***). The presence or absence of a fragment is reported in each position [31].

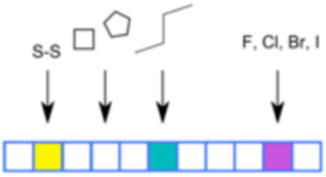


Figure 3. Illustration of MACCS fingerprint.
Reproduced from T004 · Ligand-based screening: compound similarity. 2019. Volkamerlab.Org. Retrieved 21 June 2022, from https://projects.volkamerlab.org/teachopencadd/talktorials/T004_compound_similarity.html

- **Morgan** (circular fingerprints): this type of fingerprint is based on the Morgan algorithm [32]. Each bit corresponds to a circular environment (with a defined radius) for each atom in the molecule (***Figure 4***). The number of counted neighboring bonds and atoms depends on the radius. The length of the fingerprint in this case is not limited because it can be modulated according to the radius studied.

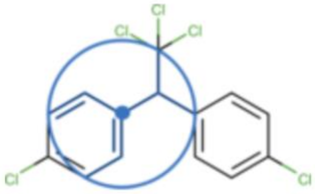
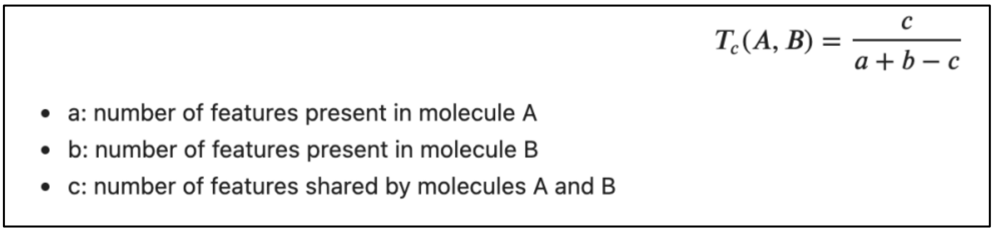


Figure 4. Illustration of Morgan circular fingerprint.
Reproduced from T004 · Ligand-based screening: compound similarity. 2019. Volkamerlab.Org. Retrieved 21 June 2022, from https://projects.volkamerlab.org/teachopencadd/talktorials/T004_compound_similarity.html

The similarity between both fingerprints is measured by different similarity coefficients. In this work, Tanimoto and Dice coefficients are chosen:

- **Tanimoto coefficient**: it is the most popular coefficient in both chemical informatics and computational medicinal chemistry because of its ease of implementation and quickness [33].

Figure 5 shows Tanimoto coefficient calculation:



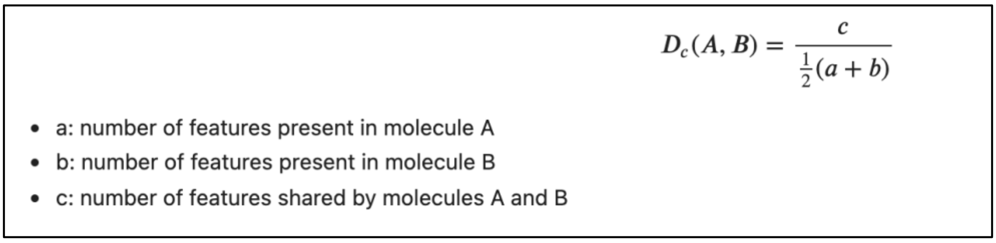
$$T_c(A, B) = \frac{c}{a + b - c}$$

- a: number of features present in molecule A
- b: number of features present in molecule B
- c: number of features shared by molecules A and B

Figure 5. Tanimoto coefficient formula and its composition.
Reproduced from T004 · Ligand-based screening: compound similarity. 2019. Volkamerlab.Org. Retrieved 21 June 2022, from https://projects.volkamerlab.org/teachopencadd/talktorials/T004_compound_similarity.html

- **Dice coefficient**: also known by other names such as Sørensen's index, Dice coefficient is a “statistic introduced to compare the similarity of two samples” [34].

Figure 6 shows the calculation of Dice coefficient.



$$D_c(A, B) = \frac{c}{\frac{1}{2}(a + b)}$$

- a: number of features present in molecule A
- b: number of features present in molecule B
- c: number of features shared by molecules A and B

Figure 6. Dice coefficient formula and its composition.
Reproduced from T004 · Ligand-based screening: compound similarity. 2019. Volkamerlab.Org. Retrieved 21 June 2022, from https://projects.volkamerlab.org/teachopencadd/talktorials/T004_compound_similarity.html

Both coefficients range from 0 to 1, being 0 the minimum similarity and 1 the maximum similarity. Although, because of each coefficient's calculation, Dice's coefficient tends to have higher values than those from Tanimoto one.

The script has been adapted from the existing script developed by the TeachOpenCadd project called Ligand-based screening: compound similarity (https://projects.volkamerlab.org/teachopencadd/talktorials/T004_compound_similarity.html). The result of the protocol is a new *SetOfSmallMolecules* object that would contain those that obtain a similarity result equal to or higher than the value set by the user specified in the protocol GUI.

3.4.4. Molecular filtering: unwanted substructures – PAINS search

PAINS (Pan-Assay INterference compoundS) are compounds that frequently show erroneous results (false positives) in high-throughput screening (HTS). This is a consequence of the functional groups they contain which cause non-specific binding to a wide range of targets [35].

PAINS concept is widely accepted by the research community, in fact, filtering by these compounds is already integrated in most virtual screening workflows. Moreover, several scientific publications, such as the Journal of Medicinal Chemistry, require this filtering for compounds presented in final papers [36].

Different filter scripts and PAINS catalogues are available, and the choice depends on the type of search or test desired. For increasing the filtering possibilities, in the protocol, it is possible to choose between the predefined PAINS catalogue offered by the RDKit library or to use a catalogue provided by the user. This user supplied file must be structured as follows: on each line, the first position is reserved for the PAINS molecule in SMART format, and the second position is for the PAINS name, both elements must be space separated.

The code corresponding to RDKit catalogue filtering is adapted from TeachOpenCadd and is called Molecular filtering: unwanted substructures (https://projects.volkamerlab.org/teachopencadd/talktorials/T003_compound_unwanted_substructures.html), and the code used for filtering by personal catalogue comes from the github repository (https://github.com/iwatobipen/rdkit_pains).

The required input file for this protocol is again a *SetOfSmallMolecules* object. The protocol GUI requests the user either to have his own PAINS catalogue or to use the

RDKit one. In case the user already has this file, the GUI will show a field to enter the catalogue absolute path.

As output archives, there are two files with txt extension and two *SetOfSmallMolecules* objects, one of them containing those molecules where no match has been found with any of the PAINS in the catalogue. The other one contains those molecules that have at least one match. The matching PAINS will be registered in the *SetOfSmallMolecules* in the form of an attribute for each molecule. Therefore, the user will be able to analyze any molecules that have not passed the filter through a viewer attached to *SetOfSmallMolecules* objects which displays a table with the attributes for each molecule in the set. This is helpful because some compounds that are considered interfering structures are not excluded for being used as drugs [37] as 5% of FDA-approved drugs have known PAINS. Hence, the user can use the protocol to discard all molecules with PAINS or perform a more exhaustive molecule-by-molecule analysis.

3.4.5. LBVS based on the shape

The next filter is based on the analysis of the shape of the query molecule/ligand used as a reference in the search for similar structures. The methods applied in the shape filtering are usually based on the calculation of distances, surface areas and/or volumes.

A commonly used measure for comparing the structure of two molecules in virtual screening programs is the calculation of the Root Mean Square Deviation (RMSD) between the atoms of two molecules [38]. RMSD is a distance which describes the structural difference between two topologies, this is detailed in ***Figure 7***. The lower the RMSD between two structures, the greater the similarity between them.

$$RMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\left(x_i - x'_i\right)^2 + \left(y_i - y'_i\right)^2 + \left(z_i - z'_i\right)^2\right]}$$

Figure 7. RMSD formula. Where N is the number of atoms in the system and (x,y,z) are the coordinates of the first structure atoms and (x',y',z') are the coordinates of the equivalent atoms of the second structure.

RDKit package allows the calculation of RMSD between a set of molecules and a reference with very high similarity. However, it is unable to calculate RMSD values for dissimilar compounds.

On the one hand, it is possible to estimate the optimally (minimum) RMSD between two molecules by “rdMolAlign()” module function “AlignMol()”. It calculates the necessary 3D transformation to align the probe molecule (in a specific conformation) to the query (in a specific conformation) so that the RMSD between a specified set of atoms is minimized. On the other hand, it is also possible to calculate the best RMSD (“GetBestRMS()”) by aligning all permutations of matching atoms orders in both molecules. In some cases, it can lead to a 'combinatorial explosion' when hydrogens molecules are present. The difference between both functions is that “AlignMol()” aligns molecules without changing atom order [39].

Both calculations have been integrated into the shape filtering protocol. However, the RMSD calculation is only effective when the selected molecules are very similar, small differences in the structure of the compared molecules will cause a response error. Therefore, filtering only by such functions would imply that a large part of the starting molecules would not be analyzed, as they commonly show structural differences with the query molecule. For this reason, it is necessary to calculate other distance measures. RDKit current tools for calculating distances between molecules of unequal size are Protrude Distance and Tanimoto distance. Protrude Distance focusses on the volume mismatch and is defined as the percentage of the larger molecule which protrudes/exceeds from the smaller molecule. Tanimoto Distance measures similarity between finite sample sets and is defined as the size of the intersection divided by the size of the union of the sample sets (***Figure 8***).

$$f(A,B) = \frac{A \cdot B}{||A||^2 + ||B||^2 - A \cdot B}$$

Figure 8. Tanimoto Distance formula. Where A corresponds to a finite sample set with n binary attributes, and B to the other one.

The protocol allows you to choose which coefficient you want to calculate and requires as input a *SetOfSmallMolecules* object. The analysis result will be another *SetOfSmallMolecules* object with those molecules that have obtained a coefficient range specified in the protocol GUI.

Before the calculation of RMSD and the Tanimoto and Protrude distances, it is necessary to superimpose the structures which would be compared. So far RDKit does not have a tool that performs a previous alignment to the calculation of similarity, so the results provided by these tools must be carefully analyzed by users.

The second part of the workflow (3.5.) consists of obtaining a consensus pharmacophore using ligands from PDB database. This structure is then used in the screening of the filtered molecules (first part of the analysis), by comparing the pharmacophoric characteristics of the consensus and those belonging to the analyzed molecules.

3.5. INTEGRATED PROTOCOLS – THE PHARMACOPHORE CONCEPT

As stated above, the ligand-based approach for the discovery of new drug targets becomes necessary when information on the 3D structure of the target receptor is not available or does not exist. This implies that the filtering is focused on the comparison between candidate molecules and a tested ligand. In this context, there is another type of ligand screening which measures the similarity between different molecules by their pharmacophoric characteristics.

One of the tools included in the Scipion workflow is the construction of a consensus pharmacophore model using known ligands. These ligands must fulfil a number of characteristics: obviously, they must bind to the target protein, at the same binding site and in the same orientation [40].

Pharmacophore is defined, according to IUPAC, as "the ensemble of steric and electronic features that is necessary to ensure the optimal supra-molecular interactions with a specific biological target structure and to trigger (or to block) its biological response". Therefore, a pharmacophore can be considered as an abstract set of stereo-electronic features that ensure the favorable interaction between a ligand and its target in energetic terms. These definitions are accepted by the scientific community so that molecules that share pharmacophoric characteristics or patterns are assumed to recognize the same binding sites on a biological receptor and have similar biological profiles [41].

Pharmacophore virtual screening solves the lack of structural diversity found in similarity analyses based on 2D structures. One of the advantages is that the spatial arrangement of the stereoelectronic molecular features [42] is the most relevant in pharmacophore analysis, and the chemical structure of the ligand remains in the background. Another advantage of this approach is the simplicity of the technique which also enables the analysis of huge databases in a time-efficient way, resulting in a selection of compounds that are very likely to show biological activity on the target [43-45].

3.5.1. PDB access and ligand extraction - Protein data acquisition

Protein Data Bank (PDB) is a public and supervised (Worldwide Protein Data Bank) database that stores the three-dimensional (3D) structure of different nucleic acids and proteins. These structures are provided by different biologists and biochemists from all over the world who obtain them through different techniques such as magnetic/nuclear resonance, X-ray crystallography or electron microscopy [46].

The source code from which the protocol script has been adapted comes from tutorial 008, also by Volkamer Lab (Protein data acquisition: Protein Data Bank).
Programmatic access to PDB is done using two Python packages: biotite (https://www.biotite-python.org/) and pypdb (https://github.com/williamgilpin/pypdb/).

The protocol GUI offers different fields that complete the filtering of the protein-ligand structures registered in PDB:

- **Uniprot ID** of the target protein.

- **Experimental method**: structure determination method, among the possible options are X-RAY DIFFRACTION, SOLUTION NMR, ELECTRON MICROSCOPY, etc or None, in case the experimental technique is not specified.

- **Minimum molecular weight** of the ligand associated with the structure.

- **Quality of the structure**: the lower the angstrom resolution of the structure, the higher the quality.

- **Number of structure chains**: it is recommended to set the value of this field to 1, as it simplifies the processing of the structure and the search.

- **Date of deposition**: this field is enabled to ensure the reproducibility of the screening; the user sets a date after which any structure that is uploaded on the database will not be considered. In the protocol the user must choose a year and the date that will be passed to the search criteria is January 1st of the year set.

- **Number of final extracted ligands**.

One of the disadvantages of the source code is that it does not classify the structures according to the binding site of the ligand to the protein, which does not allow the correct selection of ligands to create the consensus pharmacophore. According to the original code, the user must do a manual classification of the resultant ligands by visual inspection of the structure. However, although manual selection is a possibility, the new code facilitates the calculation of the centers of mass of the different ligands and their

subsequent clustering thanks to the DBSCAN algorithm [47] implemented by scikit-learn package.
The center of mass can be defined as the average position of all parts of the measured system, weighted according to their mass. Through this calculation, it is possible to know the point in space where each center of mass is located, which gives an idea of the ligand position with regard to the protein and allows the ligands to be grouped according to their location.
The centre of mass is estimated using the coordinates of the ligand atoms and the molecular weights from the information included in the HETATM lines of the PDB file.

DBSCAN is a non-supervised algorithm whose acronym stands for Density-Based Spatial Clustering of Applications with Noise. It is a density-based algorithm: it estimates the density distribution of the nodes in each cluster. DBSCAN has the advantage of not requiring the specification of a desired number of clusters, unlike k-means. Instead, it is necessary to specify the minimum number of points (ligands in this case) that each cluster must have and the maximum distance between one point and other to be part of the same group. It is desirable that the minimum number of points in the clusters is one: the aim is not to concentrate the ligands in a few clusters, but to know which ones bind to the target at the same binding site.
The output of the protocol is the generation of different *SetOfSmallMolecules* objects that will be composed by the ligands that have been grouped in the same cluster; in other words, there will be as many sets as clusters are obtained. Moreover, each molecule includes its canonical SMILES and the structure of its target protein as attributes, so that the binding site can be visualized and allows the user to check the clustering performed by the script.
Thanks to the DBSCAN algorithm implementation, the automation of the process is covered: the user can directly choose any of the *SetOfSmallMolecules* for the next protocol to obtain the consensus pharmacophore. Furthermore, it is also possible for the user to create an own *SetOfSmallMolecules*, via other protocols provided by Scipion-chem, ignoring the algorithm classification, therefore the manual mode is still possible.

3.5.2. Consensus pharmacophore generation and pharmacophore-based screening

The first part of the process involves the generation of the consensus pharmacophore from the ligands extracted in the previous protocol. Each ligand has different pharmacophore characteristics that are divided into the following families: donor, acceptor, aromatic, hydrophobe, lumpedhydrophobe, posionizable. The characteristics are obtained by RDKit "GetFeatureDefs()" function and classified into the above-mentioned families with "GetFamily()" function.

Up to this point, the script obtains different clusters, each of them represents one of the families and contains all the family characteristics of all the ligands examined.

Then the coordinates of the pharmacophore characteristics for each feature type are extracted by "GetPos()" function. These coordinates are represented in a 3D space, so each one has three points in a [x,y,z] form.

Now it is necessary to re-cluster these coordinates using the k-means algorithm, needless to say, the coordinates of each feature are grouped separately. ***Figure 9*** intends to show k-means clustering performance. The clustering is performed by the "clustering()" function, which is provided again by Volkamer Lab (Ligand-based pharmacophores) and requires the following parameters:

- kq: determines the number of clusters (k) per feature type
- K: number of clusters (number of features/kq)
- Minimum cluster size: minimum number of features that a cluster must have to be taken into account by the algorithm, it is intended that the clusters contain features of most of the molecules in our set of ligands.
- Top cluster number: number of clusters to be selected, the largest ones.

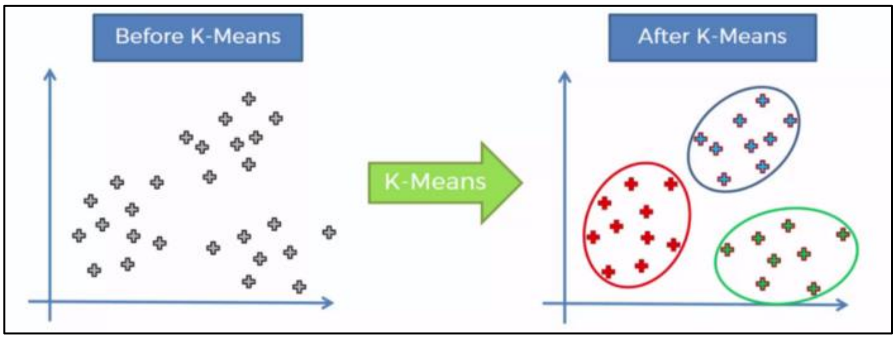


Figure 9. K-means clustering diagram.
The grey points correspond to the different characteristics from all starting ligands for a single family. After clustering, it is obtained groups of features that are spatially closest to each other.
Reproduced from T009 · ligand-based pharmacophores. 2019. Volkamerlab.Org. Retrieved 21 June 2022, from https://projects.volkamerlab.org/teachopencadd/talktorials/T009_compound_ensemble_pharmacophores.html

"clustering()" function returns the number of clusters obtained from the k-means analysis. It must be mentioned that kq should take a value that assures for all family features at least one cluster, but no more than 5 clusters. The code is adapted from TeachOpenCadd Ligand-based pharmacophores Jupyter Notebook (https://projects.volkamerlab.org/teachopencadd/talktorials/T009_compound_ensemble_pharmacophores.html) and sets the value of kq to 7; the value of minimum cluster size is 75% of the number of molecules in our set of ligands; and the clusters top number is 4.

The next step is the selection of clusters from a k-means clustering by using “get_clusters()” function. It returns the indices of these clusters; the selection is based on cluster size: the largest clusters are chosen.

After this step, “get_selected_cluster_center_coords()” function retrieves cluster center coordinates for selected clusters. These coordinates will finally be the consensus pharmacophore within the starting ligands.

Differing from the source code, in the protocol integrated in Scipion all feature families are taken into account to obtain a more accurate result.

Once the coordinates corresponding to the consensus pharmacophore are available, the last step of the protocol is filtering by this pharmacophore. The filtering is carried out using RDKit tools: (https://github.com/rdkit/UGM_2016/blob/master/Notebooks/Stiefl_RDKitPh4FullPublication.ipynb). The starting molecules are analyzed for their pharmacophore characteristics, and these are compared with the characteristics of the consensus pharmacophore. In other words, thanks to the RDKit package, the pharmacophoric characteristics of analyzed molecules are extracted per family and compared family by family with those for which the consensus pharmacophore has been constructed. The comparison is performed through the function “MatchPharmacophoreToMol()” which provides a Boolean response, *True* if there is a match between the pharmacophoric characteristics of the molecule and the model; *False*, if there is not. The user, in the protocol GUI, will select the minimum number of pharmacophoric characteristics/families which the filtered molecules must conform in order to pass the screening.

The inputs required by this protocol are two *SetOfSmallMolecules* objects, one of them containing the ligands which will build the pharmacophore (those obtained in the programmatic access or provided by the user in the form of PDB files); and another one containing the molecules that must match to the built pharmacophore. The output is a new object *SetOfSmallMolecules* containing only those molecules that have passed the filtering.

4. RESULTS

This part of the project corresponds to the results presentation, which, in this case, refers on the validation of the protocols integrated in Scipion-chem i.e., to verify that the integrated scripts work in the same way both inside and outside the Scipion platform.

For this purpose, the EGFR, ErbB1, HER1 or Epidermal Growth Factor Receptor, is selected as the target protein and the objective is to find small molecules that could work as human EGFR inhibitors. EGFR is a tyrosine kinase receptor-like protein that is located on the surface of some cells and interacts with Epidermal Growth Factor (EGF), which is involved in cell signaling pathways that control cell multiplication and survival. The interest for this protein comes from its relevance in cancer: mutations in EGFR gene cause a higher production of EGFR proteins than expected in some types of cancer cells, which leads to a faster multiplication of these cells. Therefore, the discovery of new drugs that block EGFR activity is of great importance in cancer treatments [48].

Validation is performed by executing separately the Scipion protocols and their source code in Jupyter Notebooks; most of them belonging to TeachOpenCadd platform and others from public code hosted in GitHub platform.

In the methodology section, seven different protocols have been explained, most of which are interchangeable. It is necessary to select a protocol order, so to design a specific workflow for the validation and to be able to compare the results of each step. In this case, the final workflow consists of two stages:

- The first phase of the protocol corresponds to the formation of the set of structures to be analyzed and their filtering by properties derived from their chemical structure.
    1. **Compound data acquisition**
    2. **ADME filtering**
    3. **PAINS filtering**
    4. **Compound similarity filtering**

- The second phase of the workflow corresponds to the filtering of the structures resulting from the first phase by applying a Pharmacophore based virtual screening (PBVS).
    5. **Ligand filtering and acquisition**
    6. **Pharmacophore generation and screening**

Additionally, the protocol for shape filtering is validated separately, as the source code is not integrated in any script ratified outside the Scipion platform.

1.1. STRUCTURE-BASED FILTERING

1.1.1. Compound data acquisition

Objective: to generate a set of initial molecules, based only in the Uniprot ID.

It is very important to define the parameters which, obviously, must be identical on both Scipion and the notebook to obtain comparable results. The parameters used for the validation of this workflow step are as follows:

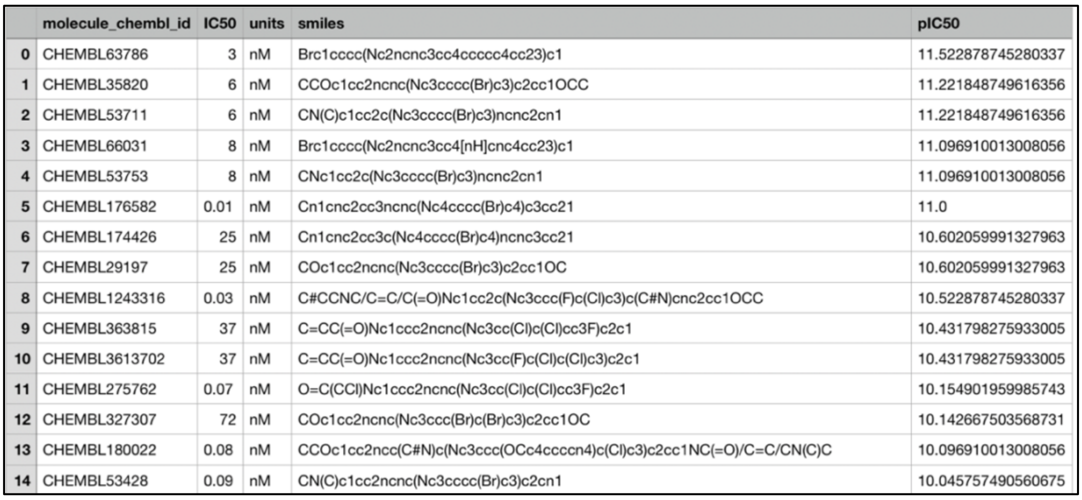
- **Uniprot ID**: "P00533"
- **Organism protein**: "Human protein"
- **Objective type**: "Single protein”
- **Bioactivity type**: "IC50”
- **Assay type**: "B"
- **Number of final structures**: “all”

The output obtained in both cases (Jupyter Notebook and Scipion) is the same: the ChEMBL target is ChEMBL203 which is a single protein and represents the human Epidermal growth factor receptor. The search returns 6283 total compounds, with pIC50 ratios ranging from 11,523 (CHEMBL63786) to 1,602 nM (CHEMBL45068).

In the case of the Scipion integrated protocol, the idea is that the user chooses the number of molecules (in SMILES format) to store at the end of the protocol in a *SetOfSmallMolecules* object, each of them having its pIC50 value as attribute. Therefore, the notebook case provides a file with csv extension and ***Table 1*** disposition; and Scipion provides smi extension file, with ***Table 2*** information, each line has these data separated by commas; and an object with ***Table 2*** structure.

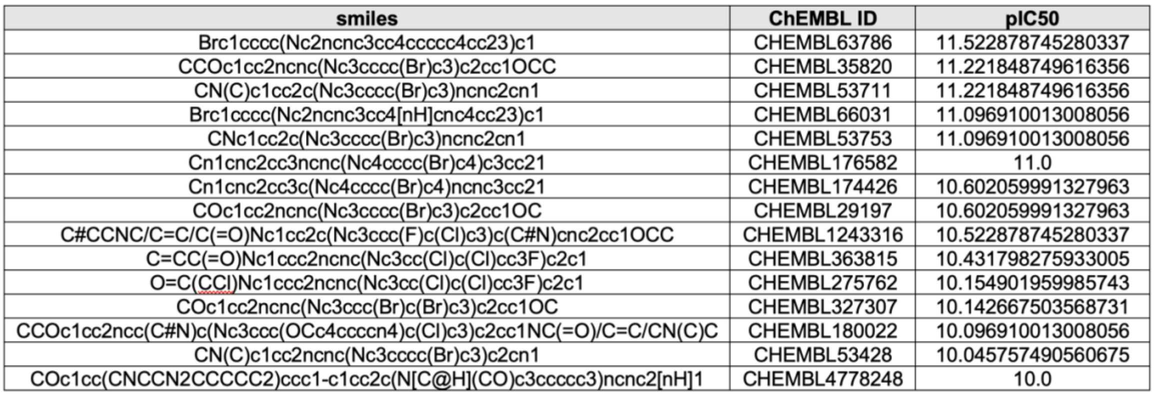
The purpose is that in the end, both processes will give identical results, not only in the number of compounds, but also in their identity and pIC50 value.

Table 1. Output csv file of "Compound data acquisition" Jupyter Notebook.
It is divided into five columns: compound ChEMBL ID; compound IC50 values, in case the value is lower than 0.01 the number is represented with a single digit (0.003 = 3); measure unit of IC50; compound canonical smiles and compound pIC50 values.



	molecule_chembl_id	IC50	units	smiles	pIC50
0	CHEMBL63786	3	nM	Brc1cccc(Nc2ncnc3cc4ccccc4cc23)c1	11.522878745280337
1	CHEMBL35820	6	nM	CCOc1cc2ncnc(Nc3cccc(Br)c3)c2cc1OCC	11.221848749616356
2	CHEMBL53711	6	nM	CN(C)c1cc2c(Nc3cccc(Br)c3)ncnc2cn1	11.221848749616356
3	CHEMBL66031	8	nM	Brc1cccc(Nc2ncnc3cc4[nH]cnc4cc23)c1	11.096910013008056
4	CHEMBL53753	8	nM	CNc1cc2c(Nc3cccc(Br)c3)ncnc2cn1	11.096910013008056
5	CHEMBL176582	0.01	nM	Cn1cnc2cc3ncnc(Nc4cccc(Br)c4)c3cc21	11.0
6	CHEMBL174426	25	nM	Cn1cnc2cc3c(Nc4cccc(Br)c4)ncnc3cc21	10.602059991327963
7	CHEMBL29197	25	nM	COc1cc2ncnc(Nc3cccc(Br)c3)c2cc1OC	10.602059991327963
8	CHEMBL1243316	0.03	nM	C#CCNC/C=C/C(=O)Nc1cc2c(Nc3ccc(F)c(Cl)c3)c(C#N)cnc2cc1OCC	10.522878745280337
9	CHEMBL363815	37	nM	C=CC(=O)Nc1ccc2ncnc(Nc3cc(Cl)c(Cl)cc3F)c2c1	10.431798275933005
10	CHEMBL3613702	37	nM	C=CC(=O)Nc1ccc2ncnc(Nc3cc(F)c(Cl)c(Cl)c3)c2c1	10.431798275933005
11	CHEMBL275762	0.07	nM	O=C(CCl)Nc1ccc2ncnc(Nc3cc(Cl)c(Cl)cc3F)c2c1	10.154901959985743
12	CHEMBL327307	72	nM	COc1cc2ncnc(Nc3ccc(Br)c(Br)c3)c2cc1OC	10.142667503568731
13	CHEMBL180022	0.08	nM	CCOc1cc2ncc(C#N)c(Nc3ccc(OCc4ccccn4)c(Cl)c3)c2cc1NC(=O)/C=C/CN(C)C	10.096910013008056
14	CHEMBL53428	0.09	nM	CN(C)c1cc2ncnc(Nc3cccc(Br)c3)c2cn1	10.045757490560675

Table 2. "Compound data acquisition" Scipion protocol output as Scipion object.
It is divided into three columns: compound canonical smiles; compound ChEMBL ID and compound pIC50 values.



smiles	ChEMBL ID	pIC50
Brc1cccc(Nc2ncnc3cc4ccccc4cc23)c1	CHEMBL63786	11.522878745280337
CCOc1cc2ncnc(Nc3cccc(Br)c3)c2cc1OCC	CHEMBL35820	11.221848749616356
CN(C)c1cc2c(Nc3cccc(Br)c3)ncnc2cn1	CHEMBL53711	11.221848749616356
Brc1cccc(Nc2ncnc3cc4[nH]cnc4cc23)c1	CHEMBL66031	11.096910013008056
CNc1cc2c(Nc3cccc(Br)c3)ncnc2cn1	CHEMBL53753	11.096910013008056
Cn1cnc2cc3ncnc(Nc4cccc(Br)c4)c3cc21	CHEMBL176582	11.0
Cn1cnc2cc3c(Nc4cccc(Br)c4)ncnc3cc21	CHEMBL174426	10.602059991327963
COc1cc2ncnc(Nc3cccc(Br)c3)c2cc1OC	CHEMBL29197	10.602059991327963
C#CCNC/C=C/C(=O)Nc1cc2c(Nc3ccc(F)c(Cl)c3)c(C#N)cnc2cc1OCC	CHEMBL1243316	10.522878745280337
C=CC(=O)Nc1ccc2ncnc(Nc3cc(Cl)c(Cl)cc3F)c2c1	CHEMBL363815	10.431798275933005
O=C(CCl)Nc1ccc2ncnc(Nc3cc(Cl)c(Cl)cc3F)c2c1	CHEMBL275762	10.154901959985743
COc1cc2ncnc(Nc3ccc(Br)c(Br)c3)c2cc1OC	CHEMBL327307	10.142667503568731
CCOc1cc2ncc(C#N)c(Nc3ccc(OCc4ccccn4)c(Cl)c3)c2cc1NC(=O)/C=C/CN(C)C	CHEMBL180022	10.096910013008056
CN(C)c1cc2ncnc(Nc3cccc(Br)c3)c2cn1	CHEMBL53428	10.045757490560675
COc1cc(CNCCN2CCCCC2)ccc1-c1cc2c(N[C@H](CO)c3ccccc3)ncnc2[nH]1	CHEMBL4778248	10.0

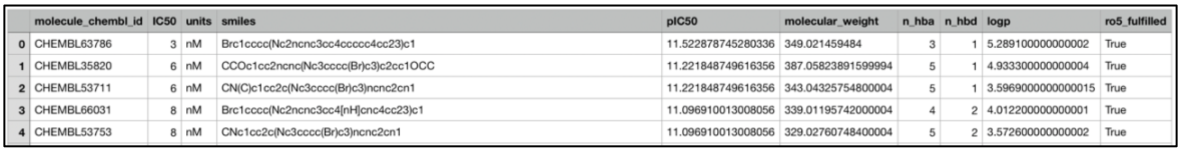
1.1.2. Molecular filtering: ADME and lead-likeness criteria

Objective: filtering those molecules obtained in the programmatic access to ChEMBL by using Lipinski's rule (Ro5).

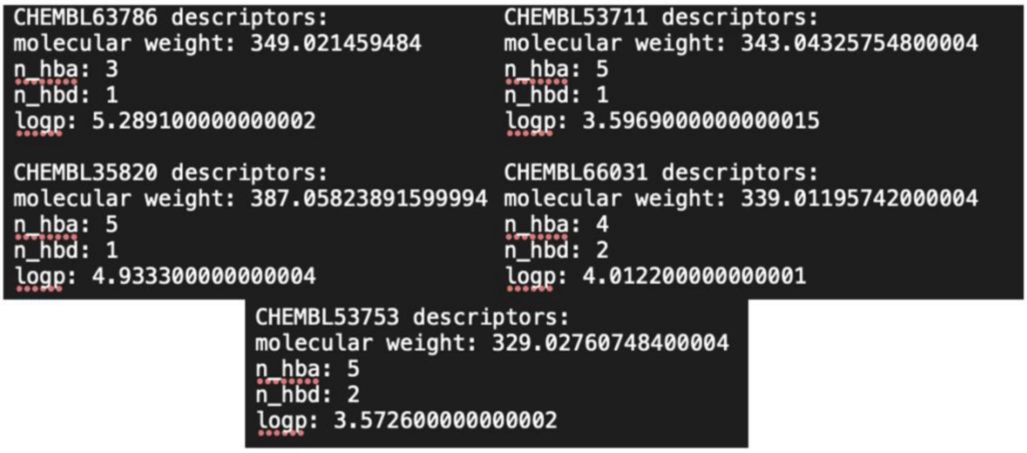
The result for both platforms is that 5091 of the 6283 total molecules have passed the filter, i.e., they fulfil at least 3 of the 4 Lipinski rules. The remaining structures, 1192 molecules, are eliminated from the workflow, being discarded for the following steps.

The Notebook again produces as output a csv file with the structure of ***Table 3***. In the case of the Scipion protocol, the main output is an object with the molecules that have passed the filter. However, for validation, it is included a function that creates a file, called "check.txt" which is shown in the ***Figure 10***, and it saves the result of the Lipinski rules for each molecule. The calculation of these values verifies that on both platforms the Ro5 have identical results, this way the implementation of the protocol is validated.

Table 3. Output csv file of "ADME and lead-likeness criteria" Jupyter Notebook.
It is divided into ten column: compound ChEMBL ID; compound IC50 values, in case the value is lower than 0.01 the number is represented with a single digit (0.003 = 3); measure unit of IC50; compound canonical smiles; compound pIC50 values; compound molecular weight; compound number of hydrogen bond acceptors; compound number of hydrogen bond donors; compound logp; and whether the compound fulfil the Ro5.



	molecule_chembl_id	IC50	units	smiles	pIC50	molecular_weight	n_hba	n_hbd	logp	ro5_fulfilled
0	CHEMBL63786	3	nM	Brc1cccc(Nc2ncnc3cc4ccccc4cc23)c1	11.522878745280336	349.021459484	3	1	5.289100000000002	True
1	CHEMBL35820	6	nM	CCOc1cc2ncnc(Nc3cccc(Br)c3)c2cc1OCC	11.221848749616356	387.05823891599994	5	1	4.933300000000004	True
2	CHEMBL53711	6	nM	CN(C)c1cc2c(Nc3cccc(Br)c3)ncnc2cn1	11.221848749616356	343.04325754800004	5	1	3.5969000000000015	True
3	CHEMBL66031	8	nM	Brc1cccc(Nc2ncnc3cc4[nH]cnc4cc23)c1	11.096910013008056	339.01195742000004	4	2	4.012200000000001	True
4	CHEMBL53753	8	nM	CNc1cc2c(Nc3cccc(Br)c3)ncnc2cn1	11.096910013008056	329.02760748400004	5	2	3.572600000000002	True



```
CHEMBL63786 descriptors:             CHEMBL53711 descriptors:
molecular weight: 349.021459484      molecular weight: 343.04325754800004
n_hba: 3                             n_hba: 5
n_hbd: 1                             n_hbd: 1
logp: 5.289100000000002              logp: 3.5969000000000015

CHEMBL35820 descriptors:             CHEMBL66031 descriptors:
molecular weight: 387.05823891599994 molecular weight: 339.01195742000004
n_hba: 5                             n_hba: 4
n_hbd: 1                             n_hbd: 2
logp: 4.933300000000004              logp: 4.012200000000001
                  CHEMBL53753 descriptors:
                  molecular weight: 329.02760748400004
                  n_hba: 5
                  n_hbd: 2
                  logp: 3.572600000000002
```

Figure 10. Part of "check.txt" file.
It contains the ChEMBL ID of the compounds that pass the ADME filtering, its molecular weight, number of hydrogen bond acceptors (n_hba), number of hydrogen bond donors (n_hbd) and logp values.

1.1.3. Molecular filtering: unwanted substructures

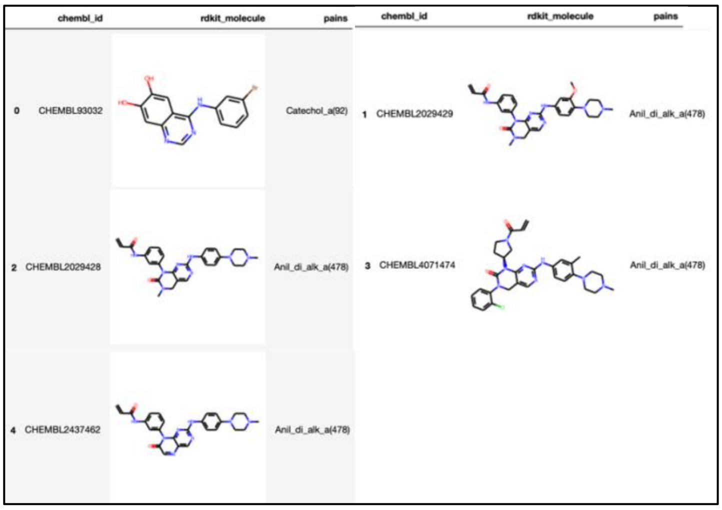
Objective: filter the structures obtained after ADME filtering by searching for PAINS.

In this project, the PAINS protocol validation focuses on the case of using the catalogue of pains included in RDKit application.

From the initial 5091 molecules, at least 453 molecules are found to have a minimum of one PAINS structure, in both platforms.

On one hand, the result obtained in the Jupyter Notebook are two data frames: one with those molecules that have passed the filtering, and another with the molecules that contain PAINS with the structure of ***Figure 11***.

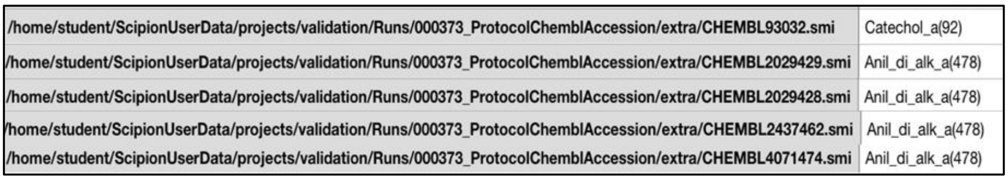
On the other hand, in the case of the Scipion protocol, the output are two objects, one containing the molecules that do not contain PAINS from which the analysis will continue. The other 453 molecules will be discarded and included in a *SetOfSmallMolecules* object called "SmallMoleculesPains". This object has the matching PAINS type associated to each molecule as an attribute, an example is shown in the ***Table 4***.



	chembl_id	rdkit_molecule	pains		chembl_id	rdkit_molecule	pains
0	CHEMBL93032		Catechol_a(92)	1	CHEMBL2029429		Anil_di_alk_a(478)
2	CHEMBL2029428		Anil_di_alk_a(478)	3	CHEMBL4071474		Anil_di_alk_a(478)
4	CHEMBL2437462		Anil_di_alk_a(478)				

Figure 11. Output file of "Unwanted substructures" Jupyter Notebook that includes those compounds that have PAINS in their structures. The file is divided into three columns: ChEMBL ID, RDKit molecule and PAINS structure found.

Table 4. Representation of "SmallMoleculesPains" object which is an output of "Unwanted substructures" Scipion Protocol. It contains those compounds that have PAINS in their structures. The file is divided into two fields: compound smi file path, and PAINS structure found.



/home/student/ScipionUserData/projects/validation/Runs/000373_ProtocolChemblAccession/extra/CHEMBL93032.smi	Catechol_a(92)
/home/student/ScipionUserData/projects/validation/Runs/000373_ProtocolChemblAccession/extra/CHEMBL2029429.smi	Anil_di_alk_a(478)
/home/student/ScipionUserData/projects/validation/Runs/000373_ProtocolChemblAccession/extra/CHEMBL2029428.smi	Anil_di_alk_a(478)
/home/student/ScipionUserData/projects/validation/Runs/000373_ProtocolChemblAccession/extra/CHEMBL2437462.smi	Anil_di_alk_a(478)
/home/student/ScipionUserData/projects/validation/Runs/000373_ProtocolChemblAccession/extra/CHEMBL4071474.smi	Anil_di_alk_a(478)

1.1.4. Ligand-based screening: compound similarity

Objective: filtering the remaining molecules after Ro5 and PAINS screening by 2D descriptors.

Next filter focuses on molecular similarity: similarity is calculated by comparing the fingerprints of the analyzed molecules and an inhibitor drug tested against the target protein, Gefitinib (***Figure 12***).

This protocol is validated by calculating, for all molecules, MACCS and Morgan fingerprints, and by calculating the similarity to Gefitinib molecule using Tanimoto and Dice coefficients.
In both options, the values obtained are exactly the same for the fingerprints as is shown in ***Tables 5 y 6***.

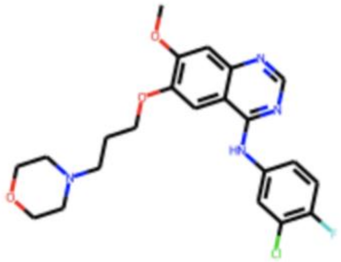
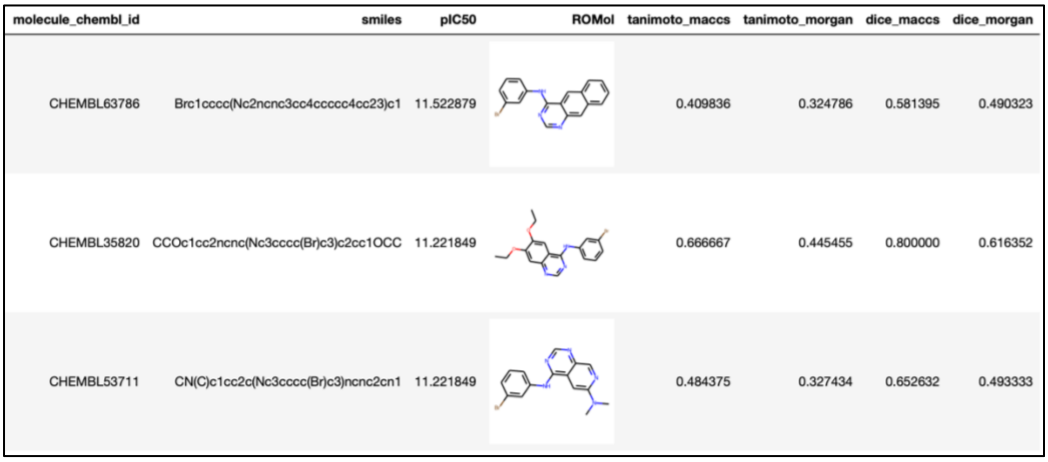


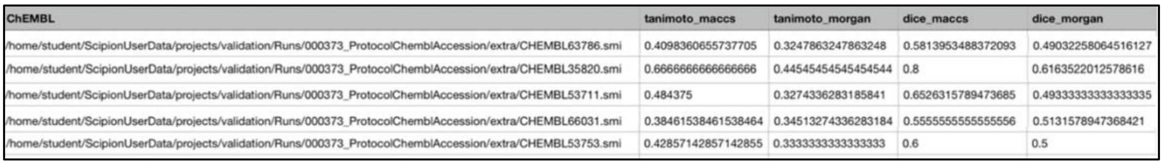
Figure 12. Gefitinib molecule structure.

Table 5. Output file of “Compound similarity” Jupyter Notebook.
It is divided into eight columns: molecule ChEMBL ID, molecule canonical smiles, molecule pIC50 value, RDKit molecule, Tanimoto-MACCS coefficient, Tanimoto-Morgan coefficient, Dice-MACCS coefficient and Dice-Morgan coefficient.



molecule_chembl_id	smiles	pIC50	ROMol	tanimoto_maccs	tanimoto_morgan	dice_maccs	dice_morgan
CHEMBL63786	Brc1cccc(Nc2ncnc3cc4ccccc4cc23)c1	11.522879		0.409836	0.324786	0.581395	0.490323
CHEMBL35820	CCOc1cc2ncnc(Nc3cccc(Br)c3)c2cc1OCC	11.221849		0.666667	0.445455	0.800000	0.616352
CHEMBL53711	CN(C)c1cc2c(Nc3cccc(Br)c3)ncnc2cn1	11.221849		0.484375	0.327434	0.652632	0.493333

Table 6. Representation of the *SetOfSmallMolecules* object which is the output of “Compound similarity” Scipion protocol. It is divided into five fields: compound smi file path, Tanimoto-MACCS coefficient, Tanimoto-Morgan coefficient, Dice-MACCS coefficient and Dice-Morgan coefficient.



ChEMBL	tanimoto_maccs	tanimoto_morgan	dice_maccs	dice_morgan
/home/student/ScipionUserData/projects/validation/Runs/000373_ProtocolChemblAccession/extra/CHEMBL63786.smi	0.4098360655737705	0.3247863247863248	0.5813953488372093	0.49032258064516127
/home/student/ScipionUserData/projects/validation/Runs/000373_ProtocolChemblAccession/extra/CHEMBL35820.smi	0.6666666666666666	0.44545454545454544	0.8	0.6163522012578616
/home/student/ScipionUserData/projects/validation/Runs/000373_ProtocolChemblAccession/extra/CHEMBL53711.smi	0.484375	0.3274336283185841	0.6526315789473685	0.49333333333333335
/home/student/ScipionUserData/projects/validation/Runs/000373_ProtocolChemblAccession/extra/CHEMBL66031.smi	0.38461538461538464	0.34513274336283184	0.5555555555555556	0.5131578947368421
/home/student/ScipionUserData/projects/validation/Runs/000373_ProtocolChemblAccession/extra/CHEMBL53753.smi	0.42857142857142855	0.3333333333333333	0.6	0.5

After obtaining the results of the comparison of both fingerprints by Dice and Tanimoto coefficients, it is possible to analyze the quality of the filtering by comparing how are the different outputs. ***Figure 13*** shows the different filter results and that the choice of MACCS filtering calculation and Tanimoto coefficient comparison is the option which present a more likely Gaussian distribution. Therefore, this one seems to have filtered in the most appropriate way; since, near half of the molecules have similarities greater than or equal to 0.5, and the rest, less than or equal to 0.5. For this reason, as for continue with the filtering it is necessary to choose a fingerprint type and a similarity coefficient, the final choice is Tanimoto-MACCS duo.

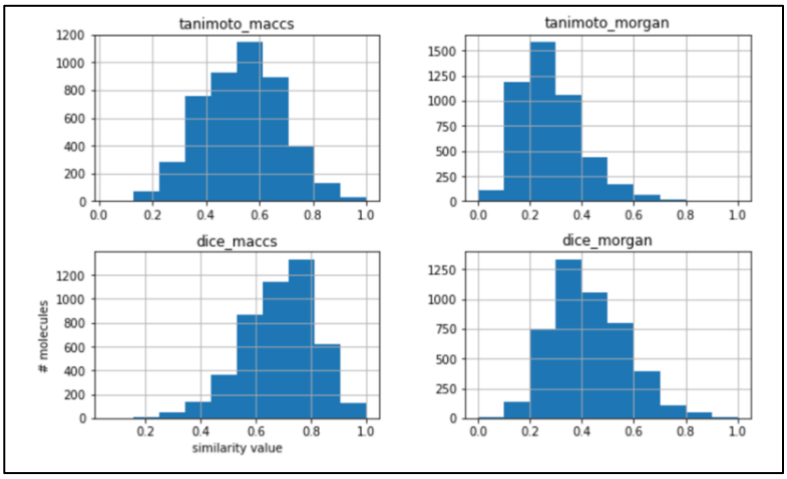


Figure 13. Matplotlib histograms showing the filtering by fingerprints results.
The X-axis shows the similarity values between the analysed molecules and Gefitinib which range from 0 to 1. The Y-axis indicates the number of molecules that are related to each similarity value. The upper title of each plot shows which type of fingerprint and similarity coefficient is being represented.

The molecules that continue for subsequent analysis are those that, after comparison of Maccs fingerprints using the Tanimoto similarity coefficient, have obtained a value equal to or greater than 0.5 with respect to the fingerprint of the Gefitinib molecule.

Specifically, this filtering concluded with the discarding of 1,786 molecules from the 4637 initial ones, leaving 2851.

1.2. PHARMACOPHORE BASED VIRTUAL SCREENING (PBVS)

The second part of the pharmacophore ligand virtual screening workflow starts with ligand extraction to obtain the consensus pharmacophore.

1.2.1. Protein data acquisition: Protein Data Bank (PDB)

Objective: obtain different target protein ligands in pdb files.

The parameters used in the search on both platforms are as follows:

- Uniprot code: P00533
- Experimental method: X-RAY-DIFFRACTION
- Minimum molecular weight of the ligand: 100.0
- Structure quality: 3.0
- Number of chains in the structure: 1
- Deposition date: 2020
- Number of final ligands to be obtained: 4
- Maximum distance between instances: 0.8 angstrom
- Minimum samples of a cluster: 1

The programmatic search with these parameters in ChEMBL results in the structures 5UG9, 5HG8, 5UG8, 5UGC and their respective ligands (8AM, 634, 8BP and 8BS).
The Jupyter Notebook only performs the ligand search without going into where they bind to the EGFR.
In the case of the Scipion protocol, a classification according to the protein-ligand binding site is obtained, the result is that all the ligands are included in a cluster, so it is interpreted that they all bind to the protein at the same binding site.

To check if the algorithm has done a good clustering, the pdb files are overlaid using the "MatchMaker" function of Chimera [49] (***Figure 14.A***), and all structures that do not correspond to the ligands are removed (***Figure 14.B***). This analysis shows that the ligands overlap in the same area of the protein.

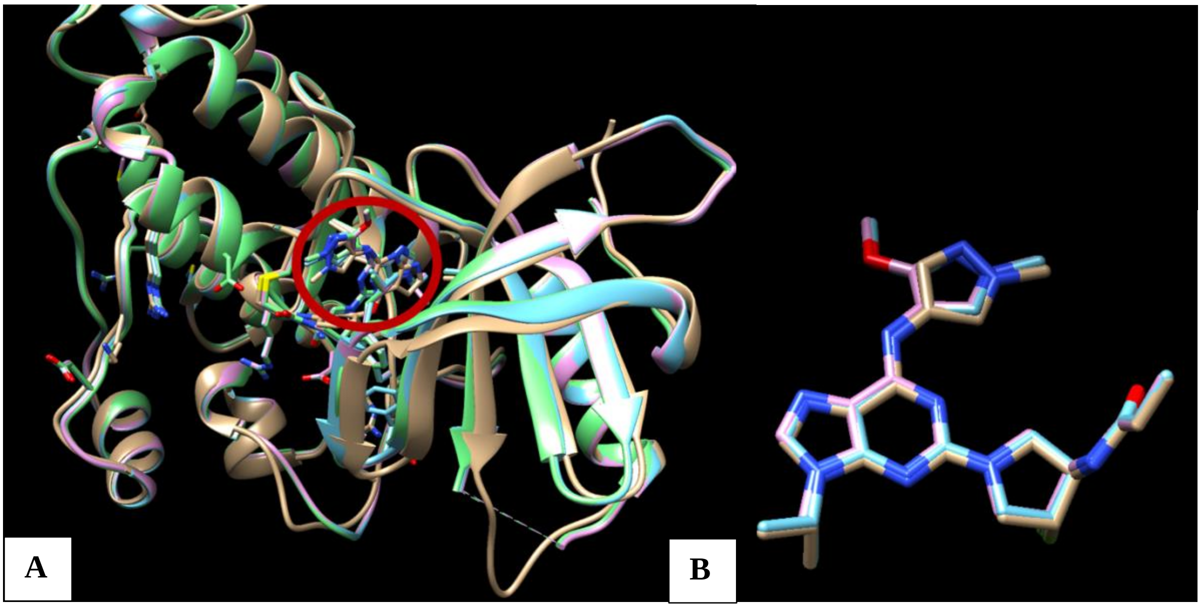


Figure 14. **A)** 8AM, 634, 8BP and 8BS ligands structures attached to the EGFR protein in the same binding site.
B) Superimposed 8AM, 634, 8BP and 8BS structures.

1.2.2. Ligand-based pharmacophores

Objective: generate a pharmacophore from the ligands that are known to interact with the EGPR.

Using the 8AM, 634, 8BP and 8BS ligands, the script, by means of the K-means algorithm, obtains the coordinates of the consensus pharmacophore with respect to the overlapping ligands.

First, it is obtained an overview of the total number of features per family and per ligand. In both cases (Scipion, Jupyter Notebook) the same results are obtained (***Table 7***). Then the coordinates of each feature characteristic are clustered (family by family) and the consensus pharmacophore is constructed with the largest clusters (***Table 8***).

Table 7. Listing of the pharmacophoric characteristics of each family per ligand.
Families are represented in rows and the ligands in columns.

	Mol1	Mol2	Mol3	Mol4
Donor	4	2	2	2
Acceptor	5	6	7	7
Aromatic	4	3	3	3
Hydrophobe	2	1	1	1
LumpedHydrophobe	1	1	1	0
PosIonizable	0	1	1	1

Table 8. Coordinates of the pharmacophoric features that constitute the consensus pharmacophore.

FEATURES	X	Y	Z
Donor1	-14.271666666666667	15.282833333333334	-25.706166666666668
Donor2	11.75675	-2.9387500000000006	-32.5585
Acceptor1	-12.479111111111111	14.526555555555555	-28.126666666666665
Acceptor2	-15.7845	18.182333333333332	-24.858166666666666
Acceptor3	13.724800000000002	-3.4705999999999992	-32.4134
Acceptor4	-16.3844	10.3812	-26.9424
Hydrophobic1	-14.052666666666667	18.13	-21.531666666666666
Aromatic1	-13.385348148148147	13.480403703703704	-27.773292592592593
Aromatic2	12.531766666666666	-3.456149999999999	-33.11249166666667

Figure 15 shows the consensus pharmacophore features, embodied in the structure of the starting ligands, that are obtained by using the Pharmagist and Pharmit tools.

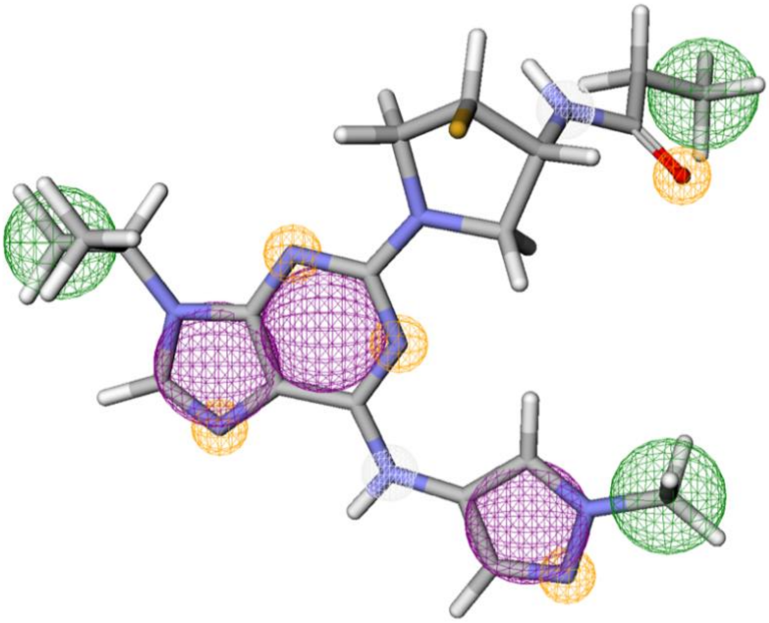


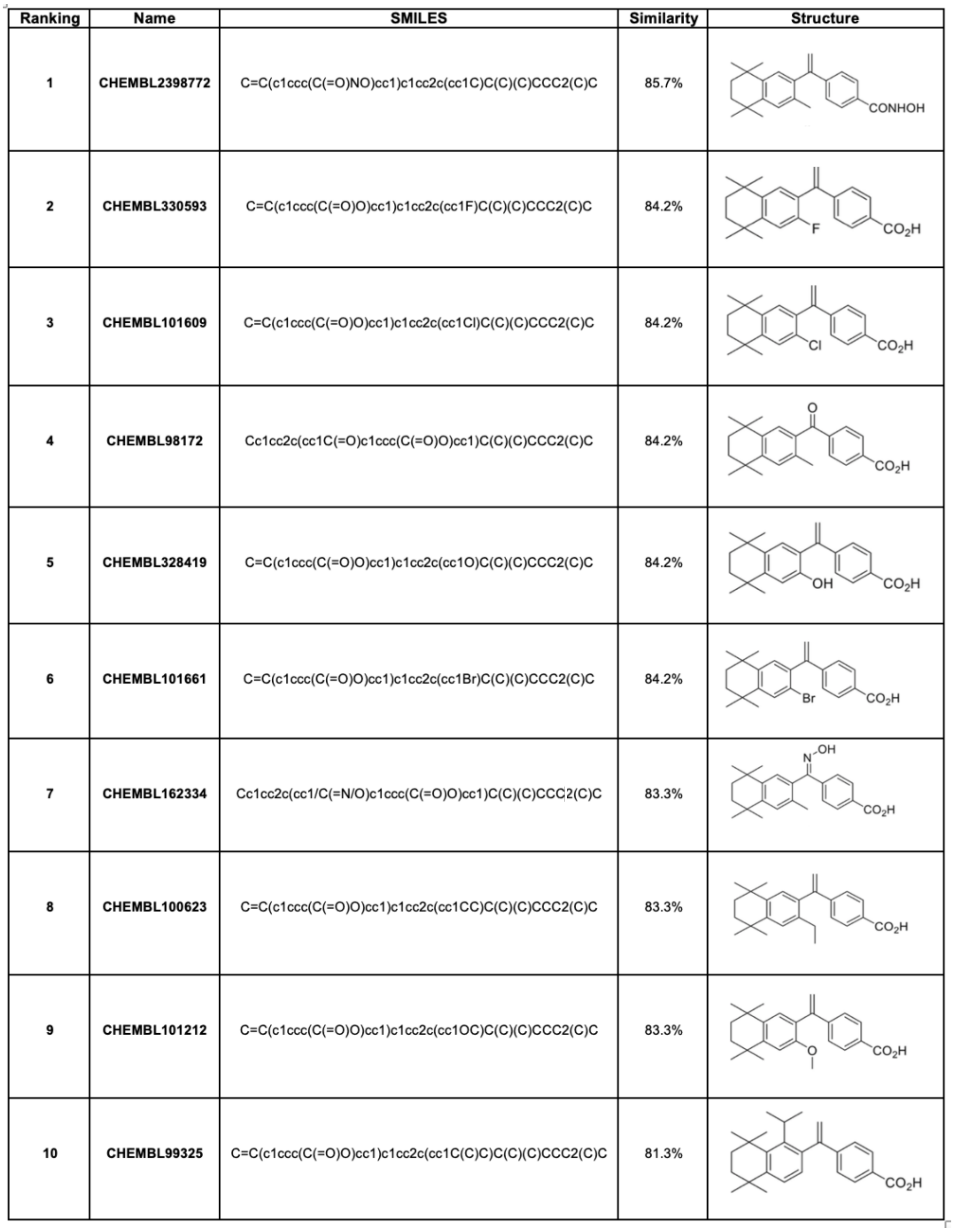
Figure 15. Diagram of the consensus pharmacophore obtained with 8AM, 634, 8BP and 8BS ligands from Pharmit. White spheres correspond to the donor family characteristics; yellow spheres correspond to the acceptor family characteristics; green spheres correspond to the hydrophobic family characteristics and purple ones correspond to the aromatic family.

1.3. SHAPE-FILTERING VALIDATION

The shape filtering protocol validation is done separately, as the RDKit functions that are implemented are still under development and there is no source code with tested functionality.

Validation focuses on analyzing how the RDKit distance calculation tools behave with molecules that are classified as very similar.

Table 9. Chemical structures of the 10 most similar database molecules to Bexarotene.
Re-edited from Peón, A., Naulaerts, S., & Ballester, P. J. (2017). Predicting the reliability of drug-target interaction predictions with maximum coverage of target space. Scientific Reports, 7(1), 3820. https://doi.org/10.1038/s41598-017-04264-w



Ranking	Name	SMILES	Similarity	Structure
1	CHEMBL2398772	C=C(c1ccc(C(=O)NO)cc1)c1cc2c(cc1C)C(C)(C)CCC2(C)C	85.7%	CONHOH
2	CHEMBL330593	C=C(c1ccc(C(=O)O)cc1)c1cc2c(cc1F)C(C)(C)CCC2(C)C	84.2%	F CO_2H
3	CHEMBL101609	C=C(c1ccc(C(=O)O)cc1)c1cc2c(cc1Cl)C(C)(C)CCC2(C)C	84.2%	Cl CO_2H
4	CHEMBL98172	Cc1cc2c(cc1C(=O)c1ccc(C(=O)O)cc1)C(C)(C)CCC2(C)C	84.2%	O CO_2H
5	CHEMBL328419	C=C(c1ccc(C(=O)O)cc1)c1cc2c(cc1O)C(C)(C)CCC2(C)C	84.2%	OH CO_2H
6	CHEMBL101661	C=C(c1ccc(C(=O)O)cc1)c1cc2c(cc1Br)C(C)(C)CCC2(C)C	84.2%	Br CO_2H
7	CHEMBL162334	Cc1cc2c(cc1/C(=N/O)c1ccc(C(=O)O)cc1)C(C)(C)CCC2(C)C	83.3%	N OH CO_2H
8	CHEMBL100623	C=C(c1ccc(C(=O)O)cc1)c1cc2c(cc1CC)C(C)(C)CCC2(C)C	83.3%	CO_2H
9	CHEMBL101212	C=C(c1ccc(C(=O)O)cc1)c1cc2c(cc1OC)C(C)(C)CCC2(C)C	83.3%	O CO_2H
10	CHEMBL99325	C=C(c1ccc(C(=O)O)cc1)c1cc2c(cc1C(C)C)C(C)(C)CCC2(C)C	81.3%	CO_2H

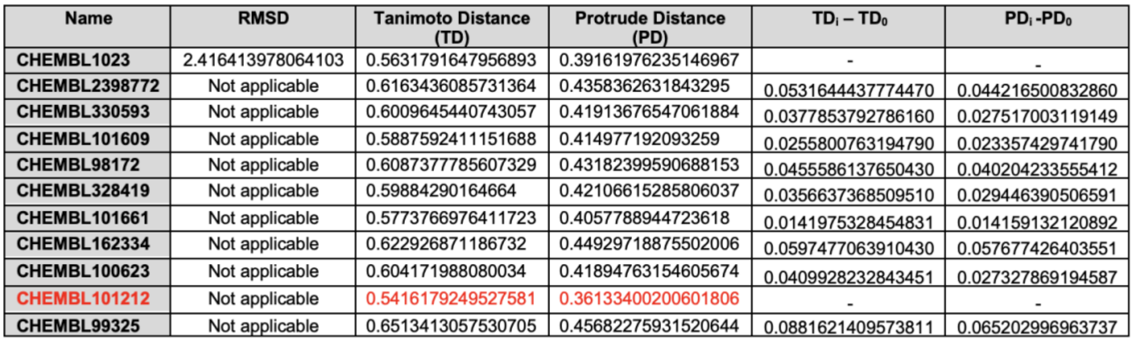
It is selected the 10 molecules from the ChEMBL database that are most structurally similar to Bexarotene (ChEMBL1023) according to Peón *et al*., 2017 [50]. These structures and the percentage of similarity with the Bexarotene molecule are shown in ***Table 9***.

The first step is the comparison of the Bexarotene molecule with itself in order to evaluate the results provided by the RMSD calculation, Tanimoto and Protrude distances. It is important to remember that RDKit performs all calculations without the process of a previous superposition between the compared structures. RDKit will run the process using the best fitting conformers according to the algorithm applied. The results obtained from this first comparison are an RMSD value of 2.416 and Tanimoto and Protrude distances values of 0.563 and 0.392 respectively. These values indicate that the conformers chosen, and the alignment performed by the algorithm (in this case) is not optimal as the result of all values should be 0 for the comparison of the same molecule.

Table 10 shows the results obtained with molecules whose target similarity ranges from 85.7% to 81.3%. It is noticed that the values for Tanimoto and Protrude distances are quite similar to those obtained in the query self-comparison. However, the calculation of the RMSD with these structures is impossible, i.e., the protocol does not find any substructure match between the probe and the query molecule. This fact is logical, since, as mentioned above, when there are small differences between the objective structure and the analyzed one, the RMSD does not obtain any response; and in this case, all the structures differ by at least 15% from Bexarotene.

By focusing on the values of distances calculation, all of them present higher values than the result of the query self-comparison, except in the case of CHEMBL101212, which presents a higher similarity value. This molecule has an alkoxy group that Bexarotene does not have. Regardless this result, the rest of the molecules present a Tanimoto distance that ranges between 0.577 and 0.651.

Table 10. Results of the shape comparison between Bexarotene and its 10 most similar molecules.
The table is divided into 6 columns: molecule ChEMBL ID, RMSD value, Tanimoto Distance value, Protrude Distance value, TD_I -TD_0 and PD_I – PD_0.
Last two columns correspond to the standardization result of the distances in relation to the results obtained by the comparison of Bexaroten with itself.



Name	RMSD	Tanimoto Distance (TD)	Protrude Distance (PD)	$TD_i – TD_0$	$PD_i - PD_0$
CHEMBL1023	2.416413978064103	0.5631791647956893	0.39161976235146967	-	-
CHEMBL2398772	Not applicable	0.6163436085731364	0.4358362631843295	0.0531644437774470	0.044216500832860
CHEMBL330593	Not applicable	0.6009645440743057	0.41913676547061884	0.0377853792786160	0.027517003119149
CHEMBL101609	Not applicable	0.5887592411151688	0.414977192093259	0.0255800763194790	0.023357429741790
CHEMBL98172	Not applicable	0.6087377785607329	0.43182399590688153	0.0455586137650430	0.040204233555412
CHEMBL328419	Not applicable	0.59884290164664	0.42106615285806037	0.0356637368509510	0.029446390506591
CHEMBL101661	Not applicable	0.5773766976411723	0.4057788944723618	0.0141975328454831	0.014159132120892
CHEMBL162334	Not applicable	0.622926871186732	0.44929718875502006	0.0597477063910430	0.057677426403551
CHEMBL100623	Not applicable	0.604171988080034	0.41894763154605674	0.0409928232843451	0.027327869194587
CHEMBL101212	Not applicable	0.5416179249527581	0.36133400200601806	-	-
CHEMBL99325	Not applicable	0.6513413057530705	0.45682275931520644	0.0881621409573811	0.065202996963737

The conclusion that can be drawn from these data is that this filter should not be used to definitively discard molecules in a screening. Instead, it can be used in a predictive way, first comparing the molecule with itself to see how accurately the “optimal” conformers are chosen and aligned by RDKit, and then analyzing the results of the rest of the molecules taking into account this first result.

The concept is correct as these protocol-implemented distances and the RMSD calculation are widely used measures to compare the structure of two molecules, but it is clear that there is still a lot of work to do before the RDKit tools can achieve suitable results.

2. CONCLUSIONS

An overview of the platform and the different protocols shows that the integration of different LBVS protocols in Scipion offers the user several advantages:

- As Scipion-chem branch has already integrated a conda environment with all the necessary tools to perform the protocols, it is only compulsory to install Scipion tool itself. This saves a lot of time and reduces the complexity of the process, offering the possibility to use different tools without the need to install different software packages separately.

- In contrast to other tools, and besides some protocols in Scipion that include programs that have pay-per-use software; the protocols that are the subject of this work include only open-source tools facilitating access to science for everyone.

- Because of the structure of the platform and the protocols: they contain information/help messages and warnings that provide information to the user about both the type of data being processed and the analyses performed by each protocol. These functions help the user to make the best decisions and reduces the requirement to have detailed knowledge of the bioinformatics tools used. In addition, Scipion tries to offer a standard workflow that the user can employ without having to design it oneself, facilitating the initiation of new users and simplifying the combined use of all tools.

- Almost all protocols accept a variety of molecular structure files through the Scipion objects usage. This fact gives considerable flexibility for the user, without being limited by a specific type of data, and provides interconnectivity between the different analyses.

- Since it is possible to use the different protocols independently, it also increases the flexibility of the tool by allowing the user to choose how and in which order to process the data.

All these advantages are only relative to the properties of the platform itself. However, the originality of the work also lies in the implementation of tools that achieve virtual screening process automation:

- The programmatic access to PDB and ChEMBL enables the analysis to be started from scratch, without the requirement of initial data, only the UNIPROT code of the target protein is needed. This also implies a timesaving for the user that is not required to create his own library of compounds to be filtered, and provides a reliable source, such as the selected databases, which is generally used in the scientific community.

- Ligand extraction from PDB database includes the clustering of the filtered ligands into groups according to the active site where they interact with the target molecule. Thus, the program is able to give the user a notion of the binding sites without the need for a bibliographic or visual inspection by the user. However, it does allow the user to review the results so that the manual mode is still available.

- In some protocols, output files are generated in order to analyze the results and to check whether the analysis has produced plausible findings.

Furthermore, it is necessary to emphasize the usefulness of using this type of screening, which saves time and money in the experimental part of the trials.

In the specific example used for the validation section, the analysis starts with 6283 molecules and after the first part of the workflow, without incorporating shape filtering and screening by pharmacological characteristics, around 3400 compounds are discarded, leaving 2851. This means that the final number of compounds is reduced by almost 55% compared to the total number of initial compounds. This reduction is quite considerable and facilitates in a simple and fast way the following steps in any type of research that involves the discovery of new molecules that have similar activities to a specific target.

To summarize, Scipion brings together the most important LBVS tools in a single platform that interconnects them, while being completely open source. In addition, new features are incorporated into the protocols that automate the process.

Even with the advantages of LBVS and its integration into a single framework, it is important to understand that the results of a LBVS are only predictions and the next natural step for successful drug discovery and development is experimental validation.

3. FUTURE WORK

Regarding the next steps that could be undertaken to improve the different protocols, we have the following:

- Creating new viewers to make processing more dynamic and easier for the user to follow, i.e., introducing graphics for better visualization of the results or the visualization of the consensus pharmacophore ("Consensus pharmacophore generation and pharmacophore-based screening" protocol).

- Include other tools, new protocols for adding new steps to the existing LBVS. For example, for those related to the creation of the pharmacophore, by considering other types of classification algorithms to improve the results.

- In particular, improving the protocol for shape filtering; for the moment only the RMSD calculations and Tanimoto and Protrude distances are implemented, which, as mentioned above, present difficulties when comparing molecules with strong dissimilarities. One way to improve the protocol is to implement another tool known as Shape-it [51] which is widely used in the superposition of molecular structures and is employed by SwissSimilairty.

- Enhance in all protocols, and in particular the structural similarity study protocol, the number of files that can be used as input, i.e., enable a wider variety of molecular files to be used as input files.

7. BIBLIOGRAPHY

[1]. Dai, W., & Guo, D. (2019). A ligand-based virtual screening method using direct quantification of generalization ability. Molecules (Basel, Switzerland), 24(13), 2414. https://doi.org/10.3390/molecules24132414

[2]. Zhang, W., Ji, L., Chen, Y., Tang, K., Wang, H., Zhu, R., Jia, W., Cao, Z., & Liu, Q. (2015). When drug discovery meets web search: Learning to Rank for ligand-based virtual screening. Journal of Cheminformatics, 7(1), 5. https://doi.org/10.1186/s13321-015-0052-z

[3]. Arora, T., & Malik, A. A. (2021). An introduction to BLAST. En Chemoinformatics and Bioinformatics in the Pharmaceutical Sciences (pp. 423–453). Elsevier.

[4]. Bajorath, J. (2002). Integration of virtual and high-throughput screening. Nature Reviews. Drug Discovery, 1(11), 882–894. https://doi.org/10.1038/nrd941

[5]. Leelananda, S. P., & Lindert, S. (2016). Computational methods in drug discovery. Beilstein Journal of Organic Chemistry, 12, 2694–2718. https://doi.org/10.3762/bjoc.12.267

[6]. Willett, P., Barnard, J. M., & Downs, G. M. (1998). Chemical similarity searching. Journal of Chemical Information and Computer Sciences, 38(6), 983–996. https://doi.org/10.1021/ci9800211

[7]. Karthikeyan, M., & Vyas, R. (2015). Role of open source tools and resources in virtual screening for drug discovery. Combinatorial Chemistry & High Throughput Screening, 18(6), 528–543. https://doi.org/10.2174/1386207318666150703111911

[8]. Haga, J. H., Ichikawa, K., & Date, S. (2016). Virtual screening techniques and current computational infrastructures. Current Pharmaceutical Design, 22(23), 3576–3584. https://doi.org/10.2174/1381612822666160414142530

[9]. Goh, G. B., Hodas, N. O., & Vishnu, A. (2017). Deep learning for computational chemistry. Journal of Computational Chemistry, 38(16), 1291–1307. https://doi.org/10.1002/jcc.24764

[10]. Wójcikowski, M., Zielenkiewicz, P., & Siedlecki, P. (2015). Open Drug Discovery Toolkit (ODDT): a new open-source player in the drug discovery field. Journal of Cheminformatics, 7(1), 26. doi:10.1186/s13321-015-0078-2

[11]. "RDKit: Open-source cheminformatics. https://www.rdkit.org"

[12]. Bragina, ME., Daina, A., Perez, MAS., Michielin, O. & Zoete, V. SwissSimilarity 2021 Web Tool: Novel Chemical Libraries and Additional Methods for an

Enhanced Ligand-Based Virtual Screening Experience., Int. J. Mol. Sci, 2022, 23(2), 811.

[13]. Zoete, V., Daina, A., Bovigny, C., & Michielin, O. SwissSimilarity: A Web Tool for Low to Ultra High Throughput Ligand-Based Virtual Screening., J. Chem. Inf. Model., 2016, 56(8), 1399.

[14]. Schneidman-Duhovny, D., Dror, O., Inbar, Y., Nussinov, R., & Wolfson, H. J. (2008). PharmaGist: a webserver for ligand-based pharmacophore detection. Nucleic Acids Research, 36(Web Server issue), W223-8. https://doi.org/10.1093/nar/gkn187

[15]. Sunseri, J., & Koes, D. R. (2016). Pharmit: interactive exploration of chemical space. Nucleic Acids Research, 44(W1), W442–W448. https://doi.org/10.1093/nar/gkw287

[16]. Sydow, D., Morger, A., Driller, M., & Volkamer, A. (2019). TeachOpenCADD: a teaching platform for computer-aided drug design using open source packages and data. Journal of Cheminformatics, 11(1), 29. https://doi.org/10.1186/s13321-019-0351-x

[17]. de la Rosa-Trevín, J. M., Quintana, A., Del Cano, L., Zaldívar, A., Foche, I., Gutiérrez, J., Gómez-Blanco, J., Burguet-Castell, J., Cuenca-Alba, J., Abrishami, V., Vargas, J., Otón, J., Sharov, G., Vilas, J. L., Navas, J., Conesa, P., Kazemi, M., Marabini, R., Sorzano, C. O. S., & Carazo, J. M. (2016). Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy. Journal of Structural Biology, 195(1), 93–99. https://doi.org/10.1016/j.jsb.2016.04.010

[18]. Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B., & Lindahl, E. (2015). GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. SoftwareX, 1–2, 19–25. https://doi.org/10.1016/j.softx.2015.06.001

[19]. Creating a protocol — Scipion 3.0.0 documentation. 2022. Github.Io. Retrieved 21 June 2022, from https://scipion-em.github.io/docs/docs/developer/creating-a-protocol

[20]. Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., & Overington, J. P. (2012). ChEMBL: a large-scale bioactivity database for drug discovery. Nucleic Acids Research, 40(Database issue), D1100-7. https://doi.org/10.1093/nar/gkr777

[21]. Anderson, E., G.D. Veith, and D. Weininger. 1987. SMILES: A line notation and computerized interpreter for chemical structures. Report No. EPA/600/M-87/021. U.S. EPA, Environmental Research Laboratory-Duluth, Duluth, MN 55804

[22]. Weininger, D. (1988), 'SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules', J. Chem. Inf. Comput. Sci. 28, 31 - 36.

[23]. Davies, M., Nowotka, M., Papadatos, G., Dedman, N., Gaulton, A., Atkinson, F., Bellis, L., & Overington, J. P. (2015). ChEMBL web services: streamlining access to drug discovery data and utilities. Nucleic Acids Research, 43(W1), W612-20. https://doi.org/10.1093/nar/gkv352

[24]. General Questions. 2021. Gitbook.io. Retrieved 21 June 2022, from https://chembl.gitbook.io/chembl-interface-documentation/frequently-asked-questions/general-questions

[25]. Biological assay development and validation. (2022, enero 24). SRI International. https://www.sri.com/biological-assay-development-and-validation/

[26]. Spektor, A. C. (2019, noviembre 21). Por qué utilizar el pIC50 en lugar del IC50 le cambiará la vida - Collaborative Drug Discovery Inc. (CDD). Collaborative Drug Discovery Inc. (CDD). https://www.collaborativedrug.com/es/why-using-pic50-instead-of-ic50-will-change-your-life/

[27]. Klopmand, G. (1992). Concepts and applications of molecular similarity, by Mark A. Johnson and Gerald M. Maggiora, eds., John Wiley & Sons, New York, 1990, 393 pp. Price: $65.00. Journal of Computational Chemistry, 13(4), 539–540. https://doi.org/10.1002/jcc.540130415

[28]. Lipinski, C. A., Lombardo, F., Dominy, B. W., & Feeney, P. J. (1997). Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Advanced Drug Delivery Reviews, 23(1–3), 3–25. https://doi.org/10.1016/s0169-409x(96)00423-1

[29]. Chandrasekaran, B., Abed, S. N., Al-Attraqchi, O., Kuche, K., & Tekade, R. K. (2018). Computer-aided prediction of pharmacokinetic (ADMET) properties. En Dosage Form Design Parameters (pp. 731–755). Elsevier.

[30]. Capecchi, A., Probst, D., & Reymond, J.-L. (2020). One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome. Journal of Cheminformatics, 12(1), 43. https://doi.org/10.1186/s13321-020-00445-4

[31]. rdkit.Chem.MACCSkeys module — The RDKit 2022.03.1 documentation. 2021. Rdkit.org. Retrieved 21 June 2022, from http://rdkit.org/docs/source/rdkit.Chem.MACCSkeys.html

[32]. Rogers, D., & Hahn, M. (2010). Extended-connectivity fingerprints. Journal of Chemical Information and Modeling, 50(5), 742–754. https://doi.org/10.1021/ci100050t

[33]. Maggiora, G., Vogt, M., Stumpfe, D., & Bajorath, J. (2014). Molecular similarity in medicinal chemistry: Miniperspective. Journal of Medicinal Chemistry, 57(8), 3186–3204. https://doi.org/10.1021/jm401411z

[34]. Dice, L. R. (1945). Measures of the amount of ecologic association between species. Ecology, 26(3), 297–302. https://doi.org/10.2307/1932409

[35]. Baell, J. B., & Holloway, G. A. (2010). New substructure filters for removal of pan assay interference compounds (PAINS) from screening libraries and for their exclusion in bioassays. Journal of Medicinal Chemistry, 53(7), 2719–2740. https://doi.org/10.1021/jm901137j

[36]. Author guidelines. 2022. Acs.Org. Retrieved 21 June 2022, from https://publish.acs.org/publish/author_guidelines?coden=jmcmar

[37]. Baell, J. B., & Nissink, J. W. M. (2018). Seven year itch: Pan-assay interference compounds (PAINS) in 2017—utility and limitations. ACS Chemical Biology, 13(1), 36–44. https://doi.org/10.1021/acschembio.7b00903

[38]. Leung, S., Bodkin, M., von Delft, F., Brennan, P., & Morris, G. (2019). SuCOS is better than RMSD for evaluating fragment elaboration and docking poses. En ChemRxiv. https://doi.org/10.26434/chemrxiv.8100203.v1

[39]. rdkit.Chem.rdMolAlign module — The RDKit 2022.03.1 documentation. 2021. Rdkit.org. Retrieved 21 June 2022, from https://www.rdkit.org/docs/source/rdkit.Chem.rdMolAlign.html

[40]. Kinghorn, A. D., Falk, H., Gibbons, S., Kobayashi, J., Asakawa, Y., & Liu, J.-K. (Eds.). (2020). Progress in the chemistry of organic natural products 112 (1a ed.). Springer Nature (page 105).

[41]. Kinghorn, A. D., Falk, H., Gibbons, S., Kobayashi, J., Asakawa, Y., & Liu, J.-K. (Eds.). (2020). Progress in the chemistry of organic natural products 112 (1a ed.). Springer Nature (page 119).

[42]. Martin, Y. C. (1992). ChemInform abstract: 3D database searching in drug design. ChemInform, 23(51), no-no. https://doi.org/10.1002/chin.199251310

[43]. Manallack, D. T. (1996). Getting that hit: 3D database searching in drug discovery. Drug Discovery Today, 1(6), 231–238. https://doi.org/10.1016/1359-6446(96)88990-2

[44]. Clark, D. E., Westhead, D. R., Sykes, R. A., & Murray, C. W. (1996). Active-site-directed 3D database searching: pharmacophore extraction and validation of hits. Journal of Computer-Aided Molecular Design, 10(5), 397–416. https://doi.org/10.1007/bf00124472

[45]. Good, A. C., & Mason, J. S. (2007). Three-dimensional structure database searches. En Reviews in Computational Chemistry (pp. 67–117). John Wiley & Sons, Inc.

[46]. Berman, H., Henrick, K., & Nakamura, H. (2003). Announcing the worldwide Protein Data Bank. Nature Structural Biology, 10(12), 980. https://doi.org/10.1038/nsb1203-980

[47]. Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei (1996). «A density-based algorithm for discovering clusters in large spatial databases with noise». En Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M., eds. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226-231. ISBN 1-57735-004-9.

[48]. Diccionario de cáncer del NCI. (2011, febrero 2). Instituto Nacional del Cáncer. https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/egfr

[49]. Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C., & Ferrin, T. E. (2004). UCSF Chimera--a visualization system for exploratory research and analysis. Journal of Computational Chemistry, 25(13), 1605–1612. https://doi.org/10.1002/jcc.20084

[50]. Peón, A., Naulaerts, S., & Ballester, P. J. (2017). Predicting the reliability of drug-target interaction predictions with maximum coverage of target space. Scientific Reports, 7(1), 3820. https://doi.org/10.1038/s41598-017-04264-w

[51]. Taminau, J., Thijs, G., & De Winter, H. (2008). Pharao: pharmacophore alignment and optimization. Journal of Molecular Graphics & Modelling, 27(2), 161–169. https://doi.org/10.1016/j.jmgm.2008.04.003