

Full Name:
Immatriculation Number:

Day 3: Integrative Data Mining of Ligand Bioactivity Data from ChEMBL and PubChem

An orthogonal approach to retrieving ligand information from available protein structures is to download and analyze ligand bioactivity data from public resources (such as K_m , K_i , IC_{50} values, or percentage inhibition). In today's protocol, we aim to integrate and analyze ligand bioactivity data from ChEMBL and PubChem. The motivation for curating data sets from different sources is to enhance the particular data sets not only in terms of the number of unique enumerated compounds but also in terms of chemical space. Since the different data sources might cover different parts of the chemical space, a greater variety in some key molecular properties of pharmaceutical interest (e.g., lipophilicity, molecular weight, topological polar surface area, and the number of rotatable bonds) can be expected when integrating various data sources. Ligand properties might provide ideas for the development of new treatments for COVID-19.

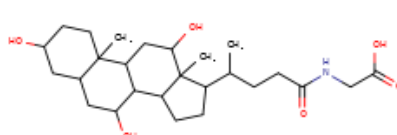
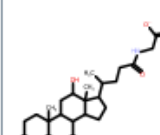

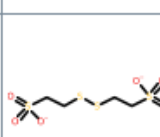
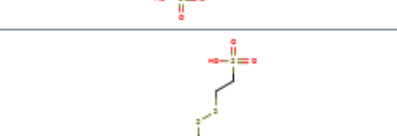
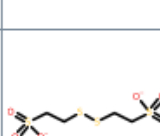
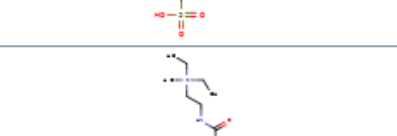
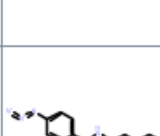
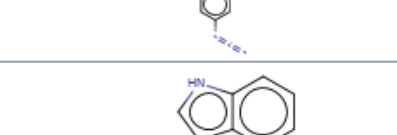

Since we aim to integrate data from different databases, we have to account for a unified molecular representation which would help us to, e.g., detect duplicate compounds in a data set. This task can be achieved by performing so-called compound "standardization". In an example workflow, we got inspired from the work of Gadaleta et al.¹ and applied a multi-step procedure for ligand curation:

1. Characters encoding stereoisomerism in SMILES format (@; \; /) are removed. Compound stereochemistry is one of the reasons which is likely to cause inconsistency between different datasets. Furthermore, most of the cheminformatic methods, such as QSAR or classification modeling, are only working with 2D descriptors. Therefore, stereochemistry is removed in the first step of the standardization procedure.

¹ Gadaleta, D., Lombardo, A., Toma, C. et al. A new semi-automated workflow for chemical data retrieval and quality checking for modeling applications. J Cheminform 10, 60 (2018) doi:10.1186/s13321-018-0315-6

2. Salts are stripped from the main compound by using the 'RDkit Salt Stripper' node. This node works with the pre-defined sets of different salts/salt mixtures by default. If requested, additional salt definitions can be forwarded to the node.

3. All salt components and other fragments are listed in a Table by using the 'Connectivity' node in combination with 'Split Collection Column' node as shown in the screenshot below:

SMI Molecule_CanonicalSmiles	Split Value 1	Split Value 2	Split Value 3	Split Value 4	Split Value 5
		NH ₃	NH ₃	Cl ⁻	Pt ²⁺
		Na ⁺	Na ⁺	Object missing ('	Object missing ('
		Na ⁺	Na ⁺	Object missing ('	Object missing ('
		I ⁻	Object missing ('	Object missing ('	Object missing ('
		Object missing ('	Object missing ('	Object missing ('	Object missing ('

4. The 'RDKit Structure Normalizer' attempts to neutralize charges and checks for the atomic clashes, etc. Additional criteria for compound quality check can be adjusted in the 'Advanced' section of the node configuration.

5. Curated compounds are checked to contain the following elements only: H,C,N,O,F,Br,I,Cl,P,S.

6. Optionally, different tautomeric forms of the single compound can be checked. However, we did not include this step into a workflow directly as it did not bring any substantial benefits. The 'Tautomers check' metanode is still included within the 'Standardization' metanode for potential usage/application.
7. The whole standardization procedure is completed by generating InChI, InChiKey, and Canonical smiles formats from the standardized compounds (which are included in *.sdf format).

For some applications, such as molecular docking into multiple binding sites, we aim to differentiate between the substrates and inhibitors. The rules applied here to define a compound as either (non)substrate or (non)inhibitor are the following: "bioactivity_type" was used as a criterion for classification as either a substrate or inhibitor. For substrates, data entries with either K_m or EC_{50} end points were considered. For inhibitors, data entries with K_i , IC_{50} were considered.

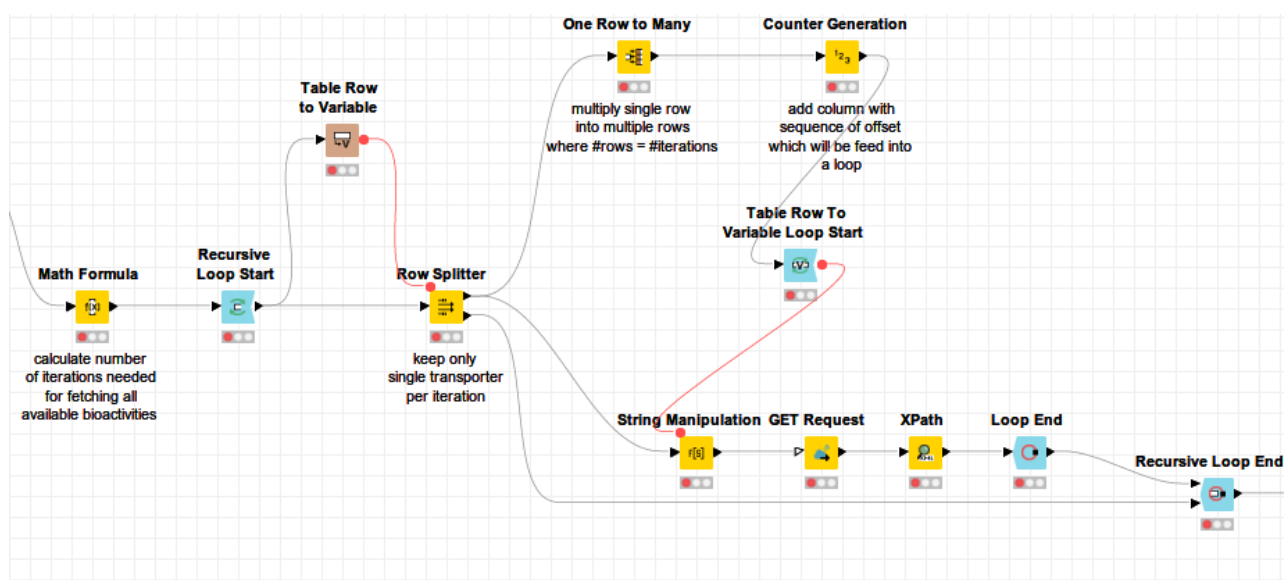
4.1 Example Workflow

4.1.1 Data curation from ChEMBL

Retrieval of ligand bioactivities from ChEMBL is done via ChEMBL webservice. A major challenge concerning fetching bioactivities from ChEMBL is the limited number of bioactivities (up to 1000 bioactivities) per single XML file. Since we do not know the number of bioactivities per target in advance, we have to make sure that our KNIME workflow will fetch all available data without any manual intervention. Therefore, we have created a workaround to download all available bioactivities per target. The corresponding part of the workflow is wrapped in a metanode called "Get bioactivities per target" and works as follows:

1. Download a single XML file per target and extract the information about total number of bioactivities
2. Calculate the number of iterations needed to fetch all available bioactivities per target: divide the number of bioactivities by 1,000 and then round up ('ceil' function in 'Math Formula' node)

3. Start a recursive loop where protein targets are processed one-by-one within a single iteration
4. Start another loop within a recursive loop where the API call is modified in a way that it dynamically changes the “off-set” parameter per each iteration; the “off-set” parameter determines which number of bioactivities should be skipped to download another portion of bioactivities for a given target. After the loop ended, all information needed is extracted from the collected XML files by the ‘XPath’ node. A following Figure illustrates the step 3 and 4 from the ‘Get bioactivities per target’ metanode:



Example: There are 2,410 bioactivities for protein X available. By dividing 2,410 by 1,000 and rounding the final value up (function “ceil” in ‘Math formula’ node) we get 3, i.e., three iterations within a loop (which is an inner part of the recursive loop) are needed to fetch all data available for transporter X. At each out of the three iterations, a column is appended to a table containing an API call with the corresponding off-set parameter, i.e.

https://www.ebi.ac.uk/chembl/api/data/activity?target_chembl_id=ChEMBL1743122&limit=1000&offset=0 (1st iteration)

https://www.ebi.ac.uk/chembl/api/data/activity?target_chembl_id=ChEMBL1743122&limit=1000&offset=1000 (2nd iteration)

https://www.ebi.ac.uk/chembl/api/data/activity?target_chembl_id=ChEMBL1743122&limit=1000&offset=2000 (3rd iteration)

At the end of the loop, 2,410 bioactivities are collected for protein X and these are processed further on. At the end, compound standardization is performed on basis of the procedure outlined in the introductory part of this protocol.

4.1.2 Data curation from PubChem

The workflow for querying data from PubChem is based on the well-established PubChem webservices and works as follows:

1. PubChem bioassay IDs ('AID' column) per requested target are downloaded.
2. Compounds linked to a specific bioassay (encoded by 'CID' identifier) are extracted from the record; more specifically, activity name (IC50, Km,...), activity value, as well as Pubmed ID (if available) is extracted from the records.
3. CID for a respective compound is used to fetch structural format from PubChem.
4. CID for a respective compound is used to fetch a compound name from PubChem; in some cases, compound names in PubChem are included as ChEMBL IDs. If this is true, the ChEMBL database is additionally queried to download a compound name.

At the end of the workflow execution, a table containing the following columns is created:

1. **Molecule_name** (if available)
2. **Uniprot_ID** of a requested protein target
3. **Target primary name**
4. **Activity_type**:
 1. Km, EC50, Ki, and IC50 endpoints
 2. For the sake of full workflow automatization, percentage inhibition values, which often represent a majority of data points, were omitted. If you are interested to re-include them into the pipeline, you can curate these type of data manually by looking into a primary publication.
 3. In some data sources there are additional activity end-points available (such as FC, Vmax, etc). These could potentially be included in the workflow as well.
5. **Pubmed_ID** (if available)

6. **Activity_pValue** (i.e., negative log of the IC50, Ki, or Km value when converted to molar)
7. **Activity_label** (“1” for active, “0” for inactive. So far, we have used a threshold of 10 uM for binary labeling. There is a possibility to adjust your workflow in a way that the user specifies his/her own activity cut-off. We did this in case of ChEMBL (see ‘Binary labeling’ section in a workflow).
8. **Standardized Molecule** (SDF format) according to an updated standardization procedure)
9. **Molecule_InChI** generated from standardized molecule
10. **Molecule_InChIKey** generated from standardized molecule
11. **Molecule_CanonicalSmiles** generated from standardized molecule
12. **Source** which indicated a database which has been used to fetch the compound information (i.e., ChEMBL or PubChem)
13. **Salt, ions, and/or different components** (each single component, is listed in a separate column which is named as “Split value x, where x = 0,1,2,3.....”)

4.2 Questions & Challenges

1. Which number of *bioactivity data points* did we retrieve per single protein target?
2. Which number of *unique compounds* did we retrieve per single protein target?
3. What different types of bioactivity endpoints (e.g., Km, Ki, IC50) did we retrieve? Write down a list of different bioactivities type and include the number of data points per given bioactivity type.
4. How many unique compounds do possess a bioactivity mismatch / a clash (activity label = 0.5)?
5. Check the compound overlap between PubChem and ChEMBL database (*hint: use an appropriate molecular structural format for checking the data overlap*).