|	
Full name:	
Immatriculation Number:	

Day 4: Substructure Searches In DrugBank

In today's protocol, we will use the ligand structures retrieved from PDB, ChEMBL, and PubChem to build a structural query and use it to screen DrugBank. The incentive is to find structural analogs of the retrieved ligands which could potentially show similar pharmacological activity on suggested COVID-19 drug targets. Since DrugBank contains a collection of marketed or withdrawn drugs, identified hits from these substructure searches could be considered for drug repurposing strategies. We will learn how to extract Murcko Scaffolds to get a quick overview of the structural diversity of available ligands. In the next step, we will hierarchically cluster available Murcko scaffolds on basis of their maximum common substructure, use loops to iterate over the clusters and create a maximum common substructure per each cluster, filter out substructures with too generic structures (i.e., scaffolds which do possess a single ring only), and use the substructures as a query for substructure searches in DrugBank.

4.1 Example Workflow

The extraction of Murcko Scaffolds for available ligand structures can be performed by using the 'RDKit Find Murcko Scaffolds' node in KNIME as follows:
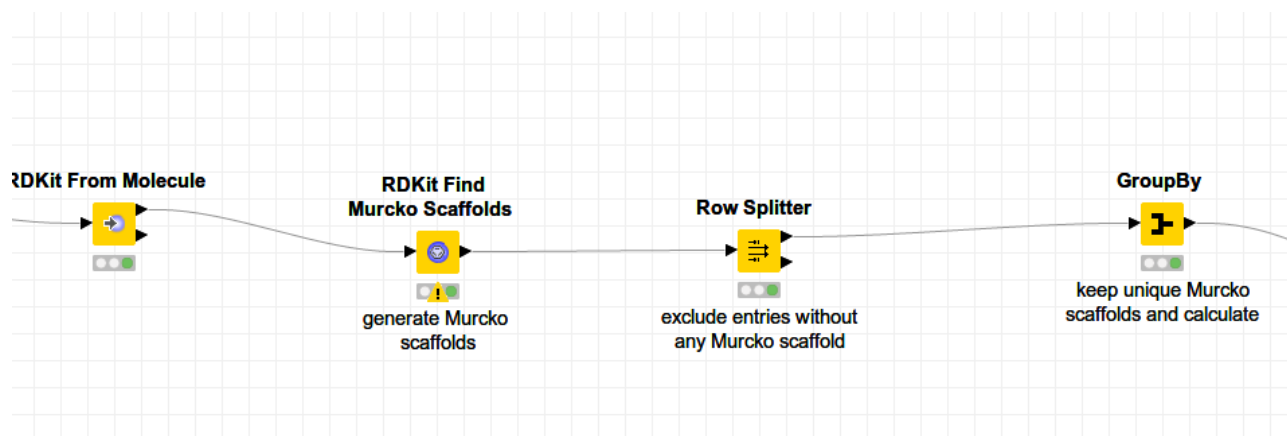
Unique Murcko scaffolds are inspected for their structural similarity. This step is done by *(1)* calculating molecular distances using the maximum common substructure approach as a metric of similarity, *(2)* hierarchical clustering, and *(3)* by assigning a threshold for cluster assignment as follows:
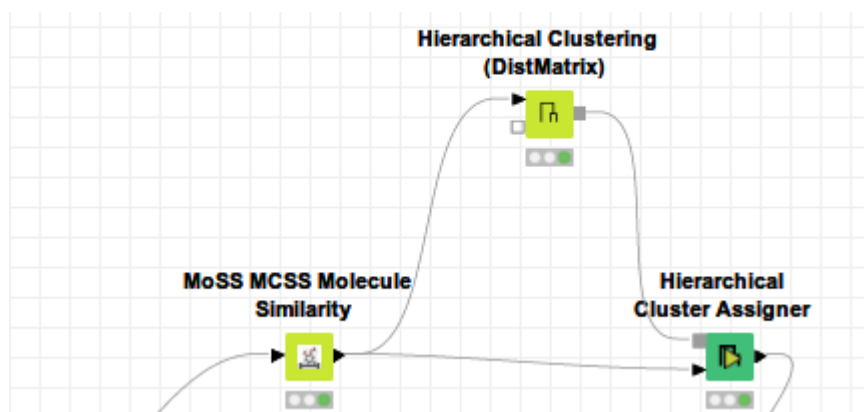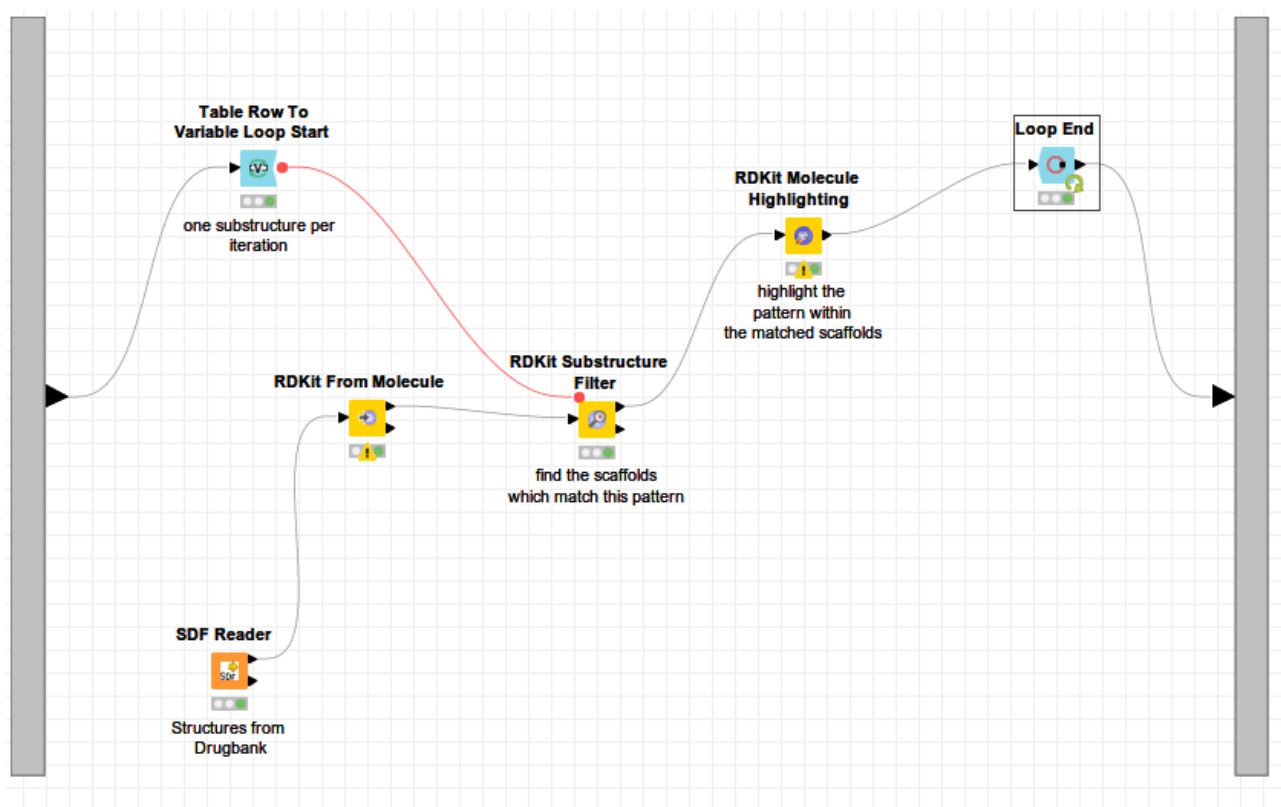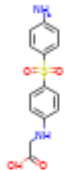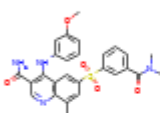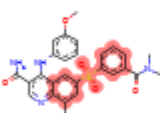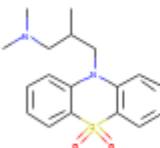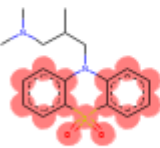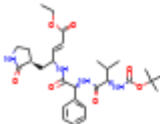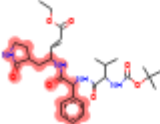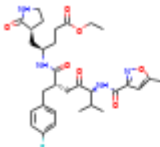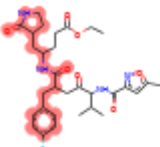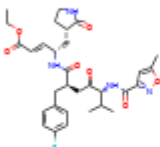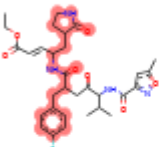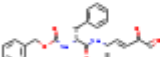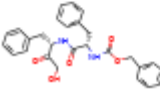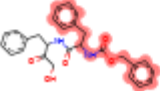

Loop nodes are handy if a particular operation is needed to be executed iteratively for each row/column in a table. Recursive loops are extensions to the regular loops which can be used in conjunction with the 'Row Splitter' node to separate the current row and the rest of the table. After the termination of the current iteration, the rest of the table is forwarded to the loop start and the next row is used in the subsequent iteration, etc. See a screenshot below to get an idea about the recursive loops. In our workflow we loop over distinct clusters of associated Murcko scaffolds to create a maximum common substructure ('RDKit MCS' node) out of all associated Murcko scaffolds belonging to a respective cluster:
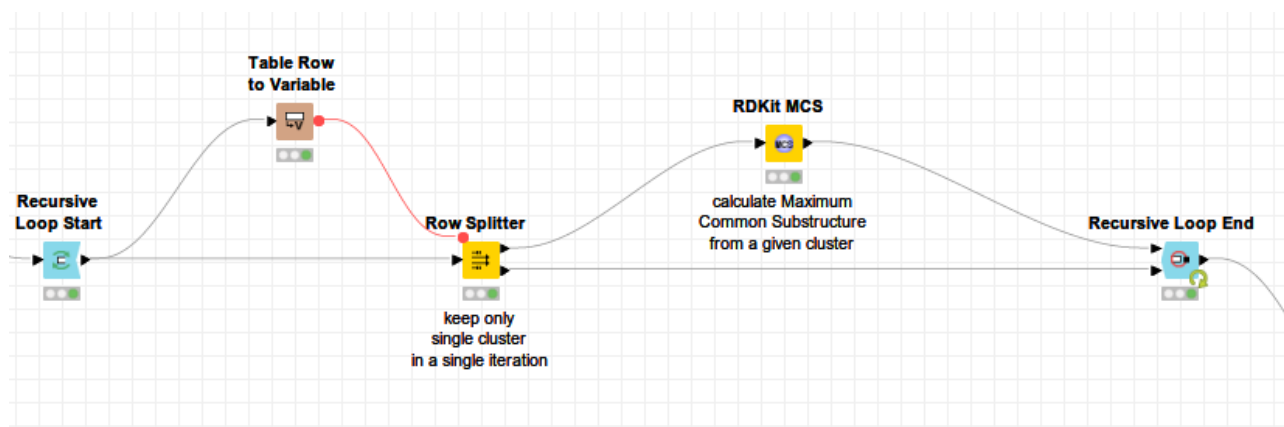
Maximum common substructures generated within a recursive loop are checked and too generic scaffolds (i.e. plane aromatic ring) are filtered out.

The substructure search in DrugBank is done in a loop. The 'Table Row To Variable Loop Start' accepts generated substructures from a previous step one-by-one, and each substructure is used as a structural query which is automatically forwarded to 'Rdkit Substructure Filter' as a flow variable. The 'Rdkit Substructure Filter' node takes input structures from DrugBank and checks whether a particular substructure is contained in the DrugBank dataset. If the condition is true, compounds from the dataset are forwarded to the 'RDKit molecule highlighting' node which visualizes an identified substructure.

Table Row To Variable Loop Start

one substructure per iteration

RDKit From Molecule

RDKit Substructure Filter

find the scaffolds which match this pattern

RDKit Molecule Highlighting

highlight the pattern within the matched scaffolds

Loop End

SDF Reader

Structures from Drugbank

Molecule ...	[...] Matchi...	Scaffold (Highlighti...	Iteration
	[12,9,7,...]		6
	[7,8,9,...]		6
	[1,0,5,...]		6
	[15,13,1...		7
	[14,13,1...		8
	[15,13,1...		8
	[11,12,1...		9
	[15,16,1...		9

4.2 Questions & Challenges

4.2.1 Generation & Analysis of Murcko scaffolds

1. Have a look at the ligands for which Murcko scaffold generation failed. What could be the reason for this? *(hint: check the definition for Murcko scaffolds[1])*

2. Which Murcko scaffold is the highest populated one? Provide the number of unique compounds which do possess this particular Murcko scaffold, as well as the canonical smiles of the Murcko scaffold *(hint: use the 'RDKit Canon SMILES' node to convert Murcko Scaffold structures into their corresponding canonical smiles form)*

4.2.2 Scaffold Clustering & Generating Maximum Common Substructures

1. How many distinct scaffold clusters do you get after hierarchical clustering?

1 BEMIS, Guy W.; MURCKO, Mark A. The properties of known drugs. 1. Molecular frameworks. *Journal of medicinal chemistry*, 1996, 39.15: 2887-2893.

2. Give the number of the highest populated scaffold cluster:

3. Calculate the average number of Murcko scaffolds which are clustered in distinct clusters.

4.2.3 Substructure Search In DrugBank

1. Calculate the number of identified compounds per target and fill in the table:

| Uniprot ID	Number of identified compounds from Drugbank
P0C6X7	
P0C6U8	
Q9BYF1	
P50052	

2. Is there a compound overlap between the protein targets? If yes, please write down the number of overlapping compounds per respective protein target:

4.2.4 Analysis of Identified Compounds from DrugBank

1. Calculate RDkit descriptors for the identified compounds and keep only those which obey Lipinski Rule of Five.[2] Name a few examples of such compounds:

2. Did you identify some known antiviral drugs? If yes, could you please provide some examples of such hits, including their DrugBank ID?

2 LIPINSKI, Christopher A., et al. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Advanced drug delivery reviews*, 1997, 23.1-3: 3-25.
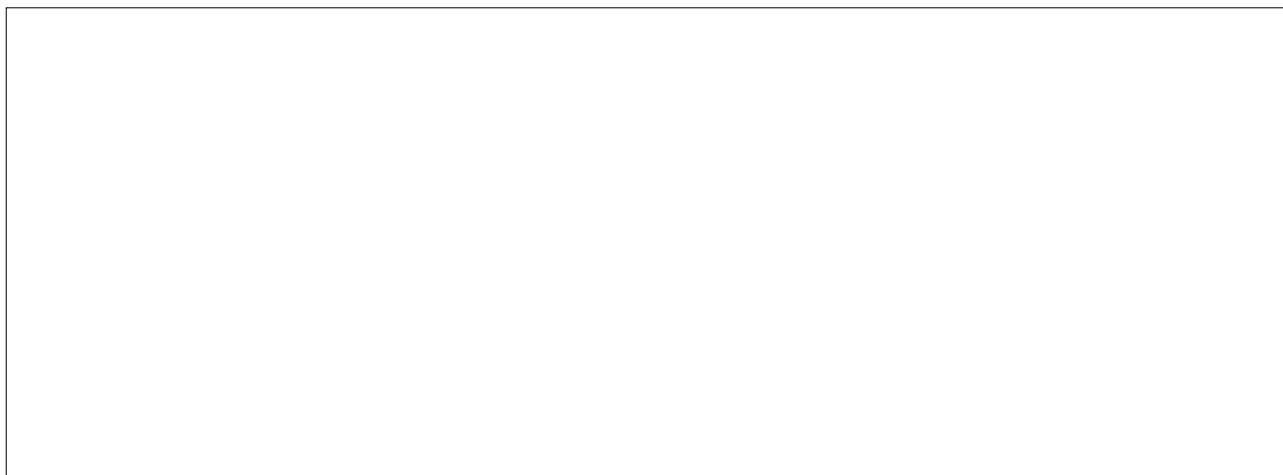
3. Choose one of the antiviral drugs from the previous question and indicate its name in the box below. On which protein target(s) of interest has the chosen drug been predicted to be active? Do a literature search and try to find some information whether the chosen drug has already been suggested for COVID-19 treatment. Is the chosen drug currently tested in clinical trials to fight COVID-19? Provide some reference (e.g. reference to a paper or link to a website). Has the chosen drug been identified via different experimental/computational approaches? Is there something known about the interaction of the chosen drug with COVID-19 target(s)? Write down a short summary (text limit: a single A4 page):