

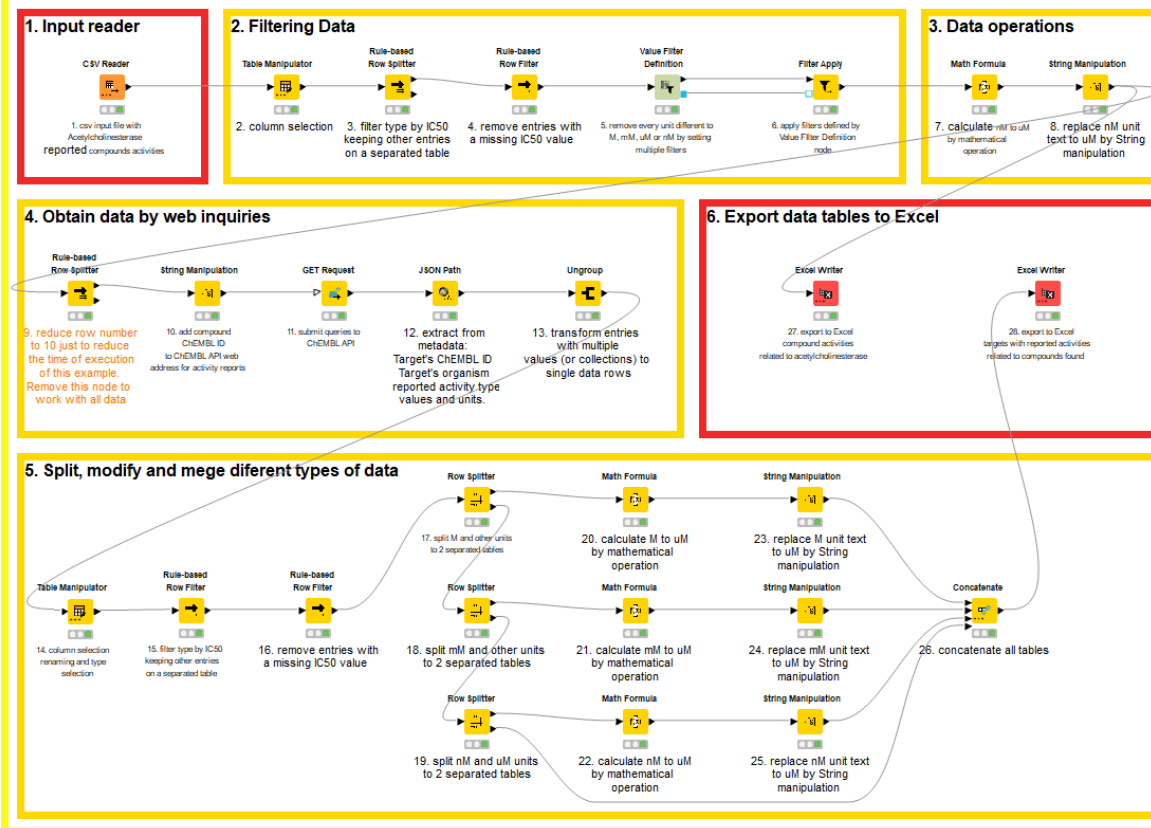
Knime Introduction tutorial – working with ChEMBL data using basic nodes – Knime 4.5

Knime tutorial - working with ChEMBL data using basic nodes - Knime 4.5

Red enclosed nodes need to be configured:

-Input node (red box 1) requires chembl220_compounds.csv file

-Outputs nodes (red box 6) requires local path and file name to export Excel files



Ramírez Lab

<https://ramirezlab.github.io/>

Carlos Peña Varas

carlospena.v@gmail.com

*This tutorial requires Knime 4.5 and a csv file with activity reports obtained from ChEMBL database.

For this exercise we need a ChEMBL data set of activity reports for compounds associated to a target, for example the compound activity reports related to acetylcholinesterase (target ChEMBL ID: chembl220).

ChEMBL Search in ChEMBL

Target Report Card

Name And Classification

ID: CHEMBL220

Type: SINGLE PROTEIN

Preferred Name: Acetylcholinesterase

Synonyms: Acetylcholinesterase, AChE, AChE

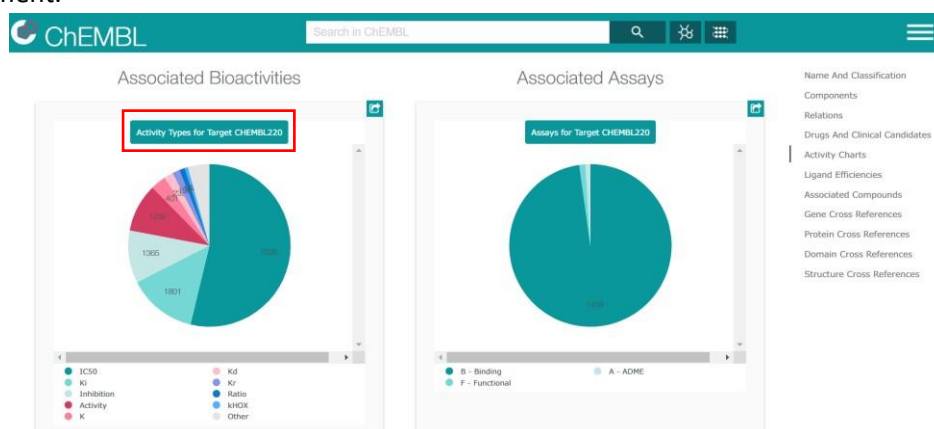
Organism: Homo sapiens

Species Group: No

Protein Target Classification: Enzyme > Hydrolase

- Name And Classification
- Components
- Relations
- Drugs And Clinical Candidates
- Activity Charts
- Ligand Efficiencies
- Associated Compounds
- Gene Cross References
- Protein Cross References
- Domain Cross References
- Structure Cross References

On the Target report Card of acetylcholinesterase we can find an Associated Bioactivities pie plot with the number of compounds with activity reported on this target and the type of activity measurement.



By clicking on Activity Types for Target CHEMBL2020 we can get the full list of compounds with information about the molecule, activity values, assays, and publication documentation. All the information can be downloaded by clicking on the download csv button.



ChEMBL Search in ChEMBL

13,480 Activities
0 Selected - Select All
Browse Compounds

Download CSV **Download TSV**

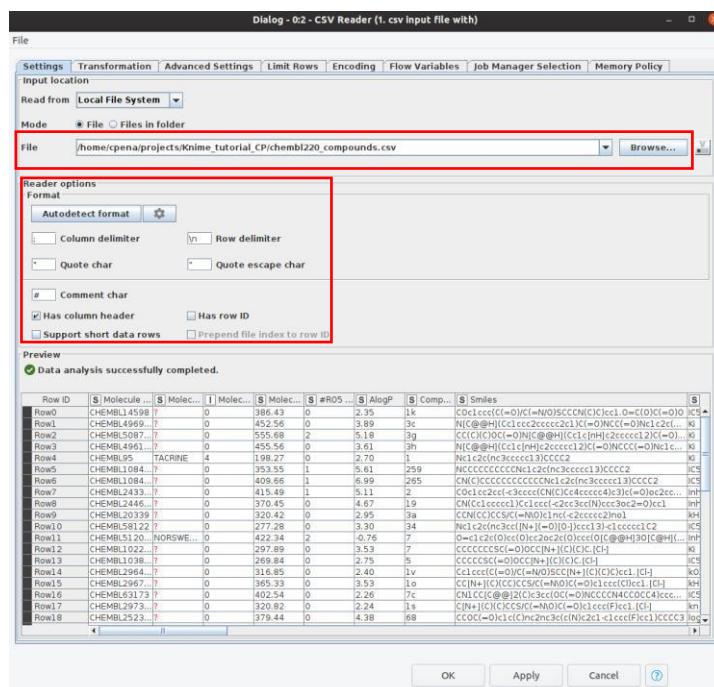
Records per page: 20

Showing 1-20 out of 13,480 records

Molecule ChEMBL ID	Compound Key	Standard Type	Standard Relation	Standard Value	Standard Units	pChEMBL Value	Comment	Assay ChEMBL ID	Assay Description	BAO Label
 CHEMBL1084092	259	IC50	=	147.0	nM	6.83	No Data	CHEMBL1119333	Inhibition of AChE	single protein format
 CHEMBL1084092	265	IC50	=	264.0	nM	6.58	No Data	CHEMBL1119333	Inhibition of AChE	single protein format

Node 1, CSV Reader.

Now this csv file can be read by Knime using CSV Reader node. On File you can browse for the csv file. In Reader options you can chose the column delimiter (in this case “;”). If the first line of the csv file contains column headers you have to check “Has column header” checkbox. Below you can previsualize data already separated by columns with their respective header names. On transformation tab you can also change column properties, but to show different node capabilities will be done in the next node.



Node 2, Table Manipulation.

On this node we can select which columns will be used for the rest of the workflow and change their names and data type. For this example, we reduce the number of columns selecting only the ones with a check mark. Below you can previsualize the table.



Node 3, Rule-based Row Splitter

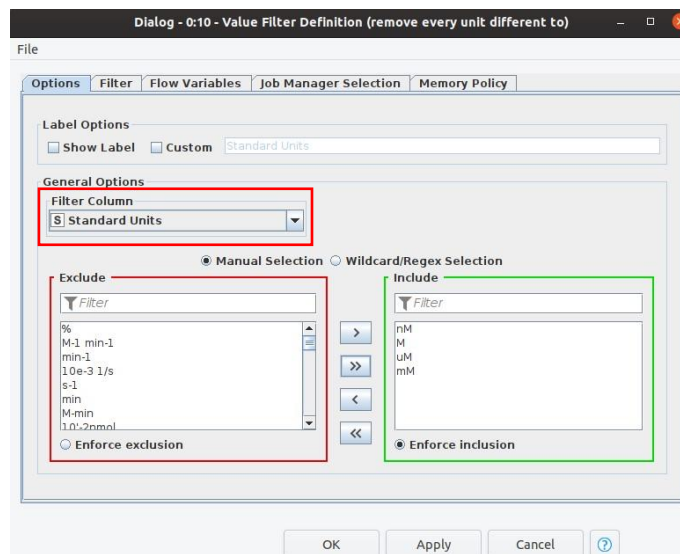
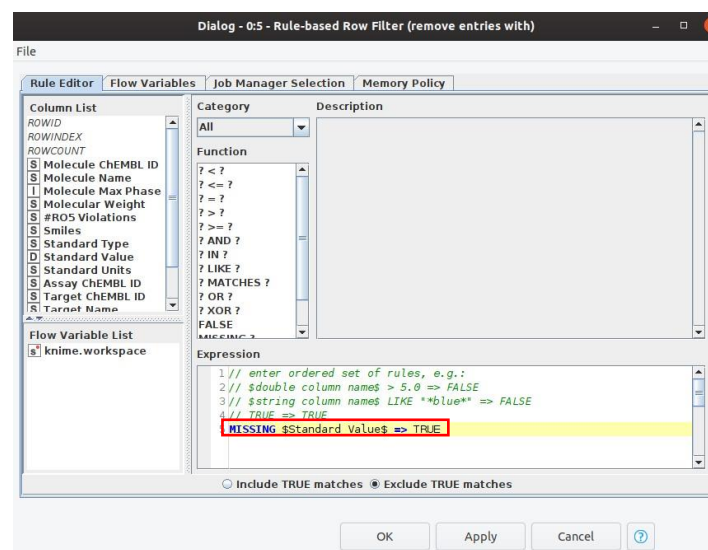
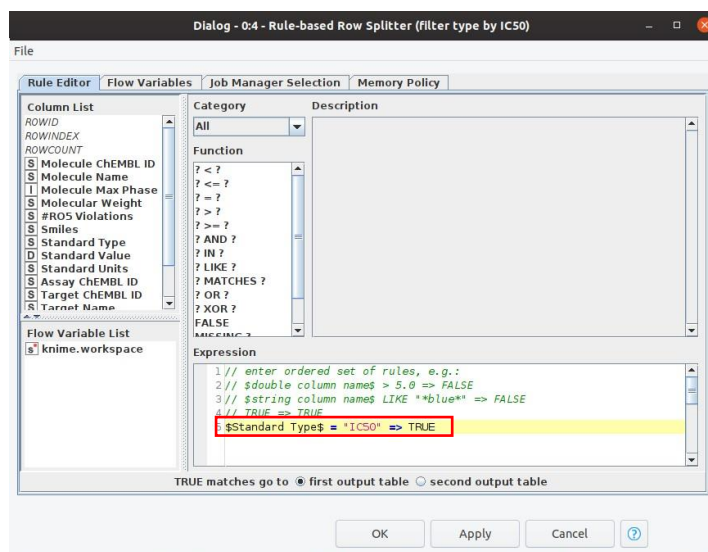
To reduce more the data, we will work only with the activities reported as IC50. To do this we will use a Rule-based Row Splitter node that allow us to separate all entries that meet a condition from those that do not on two separated tables. To use this node we chose, with double click, from Column Lists the column that we want to compare, in this case “Standar Type”, a function, “=” and between bouble quotation marks the value that we are looking for, “IC50”. To finish the configuration, we add => TRUE and check on first output table, to classify Standard Type entries with a value equal to IC50 as TRUE and separate them on the first output table. Remember that this node will create two separated tables.

Node 4, Rule-based Row Filter

Now we need to remove those entries that have empty values related to the activity, but this time let’s say that we don’t want to keep another table with the entries that has missing values because is a waste of resources a memory. So, this time we will use Rule-based Row Filter node, that will include or exclude from the table those entries that meet certain condition. For our data we will check “Exclude TRUE matches” and write the expression “MISSING \$Standard Value\$ => TRUE” that will classify as TRUE all Standard Values that have an empty value and will exclude them from our table without create a second table.

Node 5 and 6, Value Filter Definition and Filter Apply

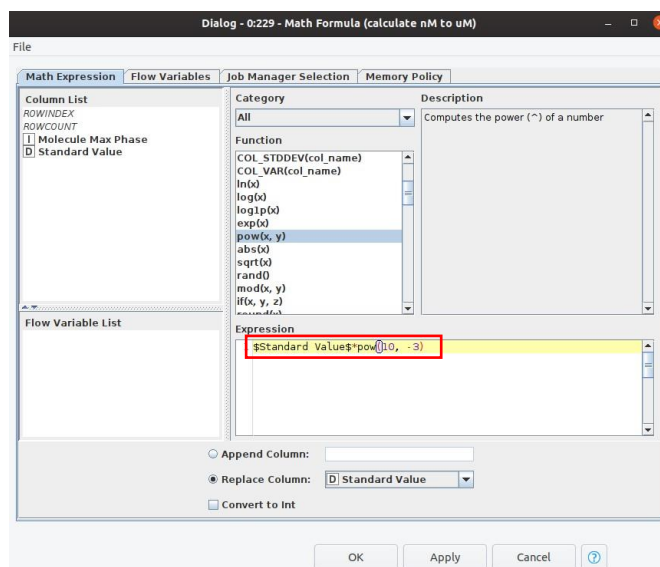
We have all IC50 activities that have a any value reported but now we need to choose the units that we are interested on. For this we will use Value Filter Definition that allow us to choose multiple conditions to filter a column. On Filter Column let’s choose Standard Units, on include green box we select all molar scale units (M, mM, uM and nM), and exclude all other units. Finally check enforce inclusion. Node 6 (Filter Apply) will execute Node 5 filter definitions and do not need any configuration, just black and blue connections from node 5.



Node 7, Math Formula

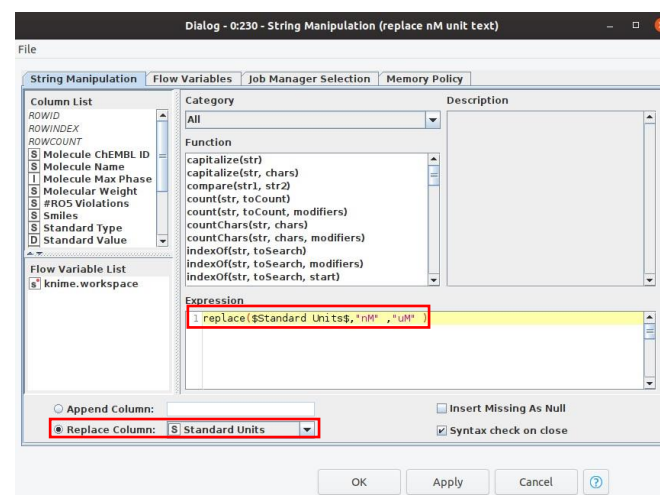
If we look the table that is generated on node 6, we will find that for this data set all activity units were measured in nM, but what if we want this values on uM units instead. The expression “\$Standard Value*\$pow(10,-3)” will multiply the value by 10⁻³ to perform this particular unit transformation.

With Math Formula node we can do all kind of mathematical operations but only with numerical columns as int and double types. This can be changed with table manipulation node.



Node 8, String Manipulation

We already transformed the value from nM to uM, however we still must change the unit column from nM to uM. For this we will be using the String Manipulation node that is capable of transform text. In this case we will write the expression “replace(\$Standard Unit, “nM”, “uM”)”, that will replace in the Standard Unit Column all values equal to nM to uM. Finally, we can choose to append a new column or replace the original column with the changes.

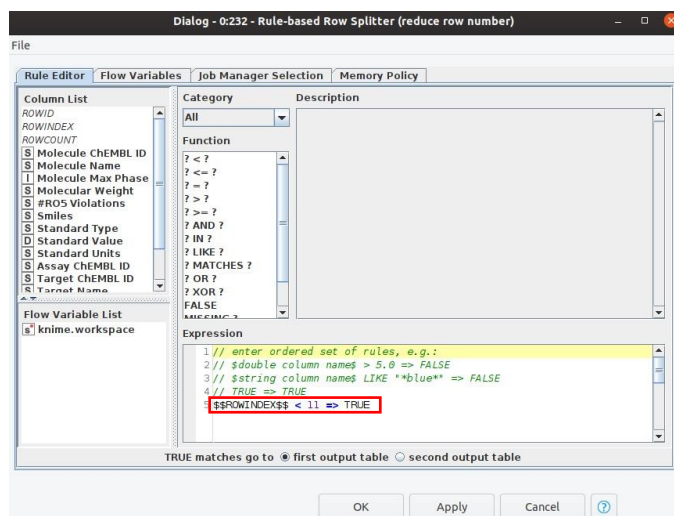


With the table generated on Node8 that contains all compounds with reported activity on acetylcholinesterase as IC50 and uM units we can export this table, on node 27, to be used in other programs like plotting software or other data manipulation on Excel.

Now we have a list of all compounds that interacts with acetylcholinesterase, but those compounds probably also interact with other targets, but there is too many to go to ChEMBL and look for targets one by one, therefore we are going to automatize the search and retrieve of this information from ChEMBL servers using Knime.

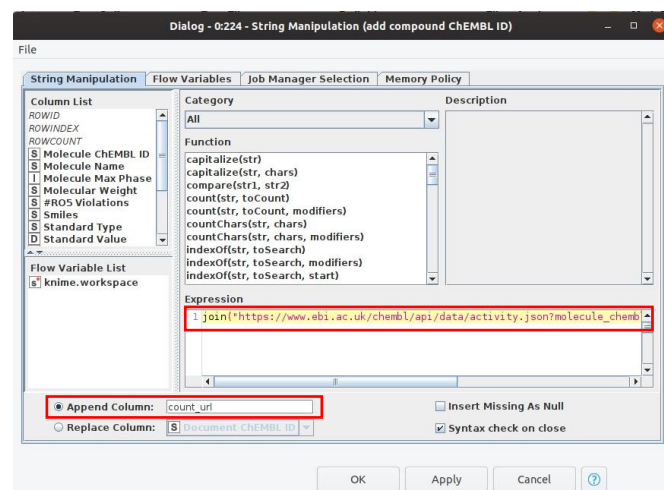
Node 9, Rule-based Row Splitter

Same as node 3. Submit queries to web servers could be a time-consuming task, therefore, to keep this tutorial within a demonstration time, this optional node creates a table only with the first 10 rows of from node 8.



Node 10, String manipulation

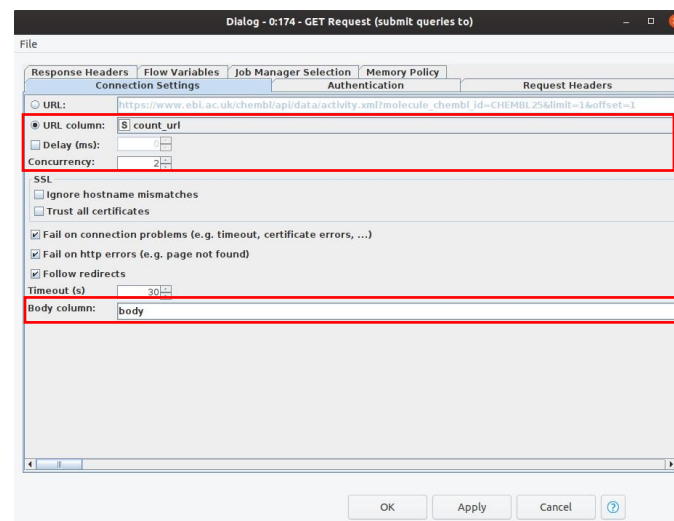
Same as node 8. We will use the join function from String manipulation to merge ChEMBL api web address with the compound ChEMBL ID with the expression below. The joined columns will be appended with the name count_url.



`join("https://www.ebi.ac.uk/chembl/api/data/activity.json?molecule_chembl_id=", $Molecule ChEMBL ID$)`

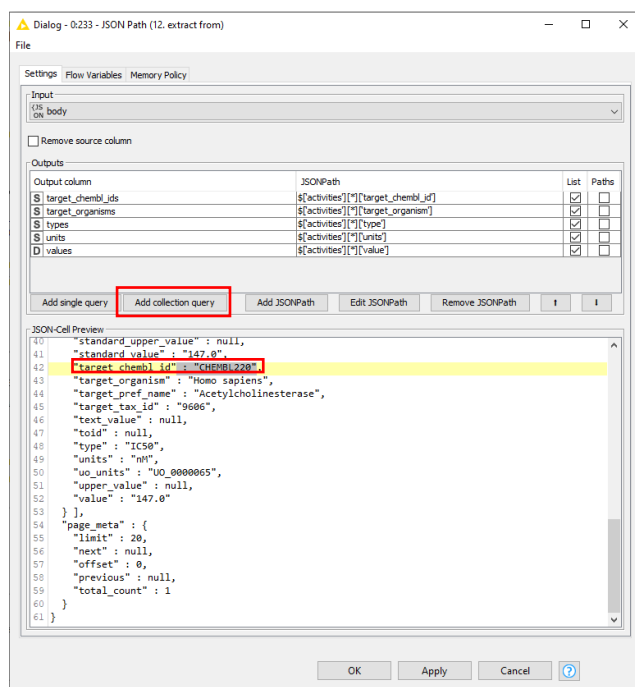
Node 11, GET Request

This node takes a column with a web address and returns information as metadata, in this case in Jason format. URL column is the column with web address from node 10 and concurrency is number of web request done at the same time (be careful, high numbers may lead to intentional loss of connection by servers). The information received will be stored as metadata on a new body column.



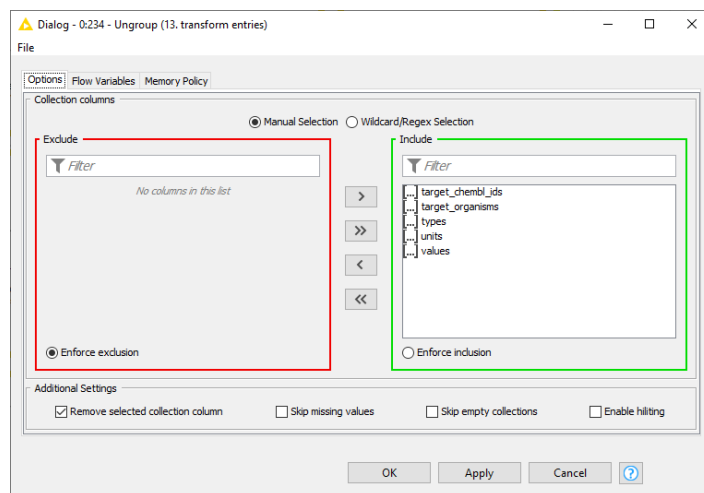
Node 12, Json Path

With Json Path node we can extract information from the metadata retrieved from ChEMBL via web request. To select data, click on the values need (for example “CHEMBL220” the target_chembl_id value) and then on add collection query, this way the node will find the path related to that data and will extract all the values from similar paths.



Node 13, Ungroup

The extracted data from metadata will be formatted as lists or collections, where every entry is separated by a “,” in a single row per request, to separate the information on multiple rows we can use Ungroup node. To configure this node we have to move all entries that we want to separate to the green include side.



Node 14, Table manipulation

Same as node 2. To make the table more manageable we will remove the columns that are no longer useful, transform the activity values from string to double type, and also, we can change some column names.

Dialog - 0.255 - Table Manipulator (14, column selection)

File Settings Flow Variables Memory Policy

Row ID handling
☐ Use existing row ID ☐ Prepend table index to row ID

Transformations
Reset actions Move up Move down Enforce types Take column from: Union Intersection

Column	New name	Type
<input type="checkbox"/> Assay CHEMBL ID		[S] String
<input type="checkbox"/> Target CHEMBL ID		[S] String
<input type="checkbox"/> Target Name		[S] String
<input type="checkbox"/> Target Organism		[S] String
<input type="checkbox"/> Target Type		[S] String
<input type="checkbox"/> Document CHEMBL ID		[S] String
<input type="checkbox"/> count_url		[S] String
<input checked="" type="checkbox"/> Status		[I] Number (integer)
<input type="checkbox"/> Content type		[S] String
<input type="checkbox"/> body		[JSON OR]
<input checked="" type="checkbox"/> target_chembl_ids	Other Targets	[S] String
<input checked="" type="checkbox"/> target_organisms		[S] String
<input checked="" type="checkbox"/> types		[S] String
<input checked="" type="checkbox"/> units		[S] String
<input checked="" type="checkbox"/> values		[D] Number (double)
<input type="checkbox"/> <any unknown new column>		[?] Default

Preview
Data analysis successfully completed.

Row ID	[S] Molecule ...	[I] Status	[S] Other T...	[S] target_...	[S] types	[S] units	[D] values
Row0	CHEMBL1084092	200	CHEMBL220	Homo sapiens	IC50	nM	147
Row1	CHEMBL1084368	200	CHEMBL220	Homo sapiens	IC50	nM	264
Row2	CHEMBL58122	200	CHEMBL220	Homo sapiens	Log IC50		-4.17
Row3	CHEMBL103873	200	CHEMBL220	Homo sapiens	IC50	M	0
Row4	CHEMBL103873	200	CHEMBL4768	Bos taurus	IC50	M	0
Row5	CHEMBL103873	200	CHEMBL220	Homo sapiens	IC50	M	0
Row6	CHEMBL103873	200	CHEMBL220	Homo sapiens	KI	M	0
Row7	CHEMBL103873	200	CHEMBL4768	Bos taurus	KI	M	0
Row8	CHEMBL103873	200	CHEMBL220	Homo sapiens	KI	M	0
Row9	CHEMBL63173	200	CHEMBL220	Homo sapiens	IC50	uM	21
Row10	CHEMBL433041	200	CHEMBL220	Homo sapiens	IC50	uM	0.29
Row11	CHEMBL433041	200	CHEMBL220	Homo sapiens	IC50	uM	0.009
Row12	CHEMBL433041	200	CHEMBL1951	Homo sapiens	IC50	uM	?
Row13	CHEMBL433041	200	CHEMBL2039	Homo sapiens	IC50	uM	?
Row14	CHEMBL433041	200	CHEMBL220	Homo sapiens	Inhibition	%	33.2
Row15	CHEMBL433041	200	CHEMBL1951	Homo sapiens	Inhibition	%	?
Row16	CHEMBL433041	200	CHEMBL2039	Homo sapiens	Inhibition	%	?

OK Apply Cancel ?

Node 15, Rule-based Row Filter

Same as node 4. Filter activity type by IC50.

Dialog - 0.259 - Rule-based Row Filter (15, filter type by IC50)

File Rule Editor Flow Variables Memory Policy

Column List
ACTIVE?
ACTIVITY
ACTIVITYCOUNT
[S] Molecule CHEMBL ID
[I] Status
[S] Other Targets
[S] target_organisms
[S] units
[S] types
[D] values

Flow Variable List
[S] Home, workspace

Category: All
Function:
? < ?
? <= ?
? = ?
? > ?
? >= ?
? AND ?
? IN ?
? LIKE ?
? MATCHES ?
? OR ?
? XOR ?
? FALSE
? MISSING ?
? NOT ?

Expression
// enter ordered set of rules, e.g.:
1 // \$double column names > 5.0 => FALSE
2 // \$string column names LIKE "Blue" => FALSE
3 // \$double column names
types\$ LIKE "IC50" => TRUE

☒ Include TRUE matches ☐ Exclude TRUE matches

OK Apply Cancel ?

Node 16, Rule-based Row Filter

Same as node 4. Exclude entries with empty values.

Dialog - 0.250 - Rule-based Row Filter (16, remove entries with)

File Rule Editor Flow Variables Memory Policy

Column List
ACTIVE?
ACTIVITY
ACTIVITYCOUNT
[S] Molecule CHEMBL ID
[I] Status
[S] Other Targets
[S] target_organisms
[S] units
[S] types
[D] values

Flow Variable List
[S] Home, workspace

Category: All
Function:
? < ?
? <= ?
? = ?
? > ?
? >= ?
? AND ?
? IN ?
? LIKE ?
? MATCHES ?
? OR ?
? XOR ?
? FALSE
? MISSING ?
? NOT ?

Expression
// enter ordered set of rules, e.g.:
1 // \$double column names > 5.0 => FALSE
2 // \$string column names LIKE "Blue" => FALSE
3 // \$double column names
MISSING \$values => TRUE

☐ Include TRUE matches ☒ Exclude TRUE matches

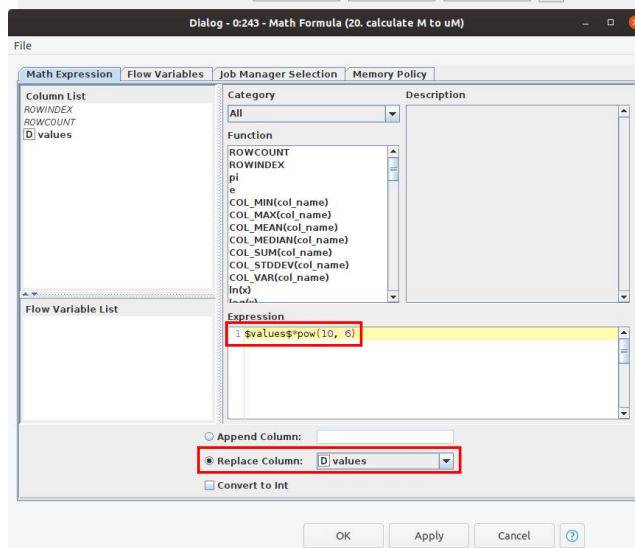
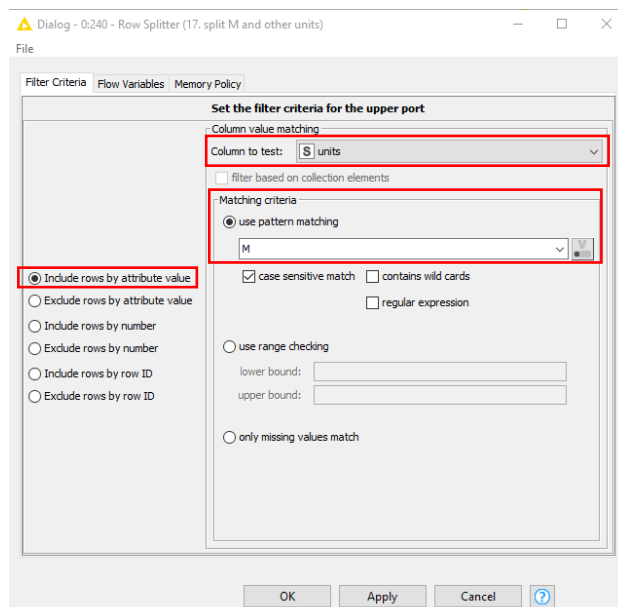
OK Apply Cancel ?

Nodes 17, 18 and 19 Row Splitter

If we look closely at table from node 16, and with the sample of 10 compounds from node 9, we should find that all the units are on molarity scale (M, mM, uM or nM). A way to get all values as uM is split the data, transform it to uM and then merge it again. For that we will use Row Splitter node, very similar in function to Rule-based Row Splitter, this node separates all values that matches a criterion on a new table, and all the other values goes to another table. First, we choose the function of the node, in this case include rows by attribute value, second, we chose the units column as the column to test, and finally, we write the pattern to match, M, mM, nM on 17, 18 and 19 nodes respectively.

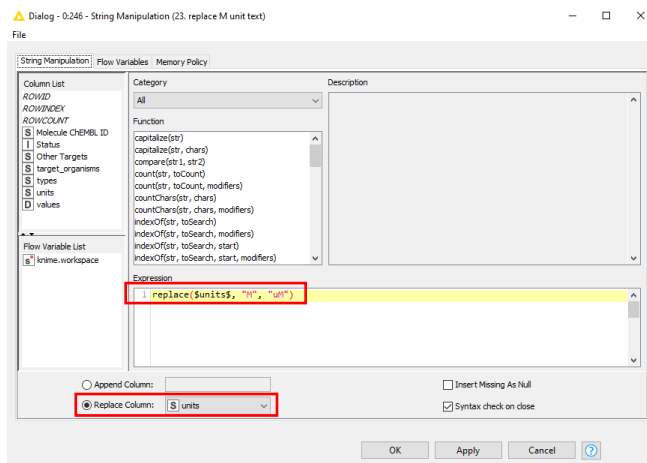
Nodes 20, 21 and 22 Math Formula

Same as node 7. Mathematical operation to transform from M, mM and nM to uM. $\$values\$ * pow(10, 6)$, $\$values\$ * pow(10, 3)$ and $\$values\$ * pow(10, -3)$ for 20, 21 and 22 nodes respectively. New values replace values column.



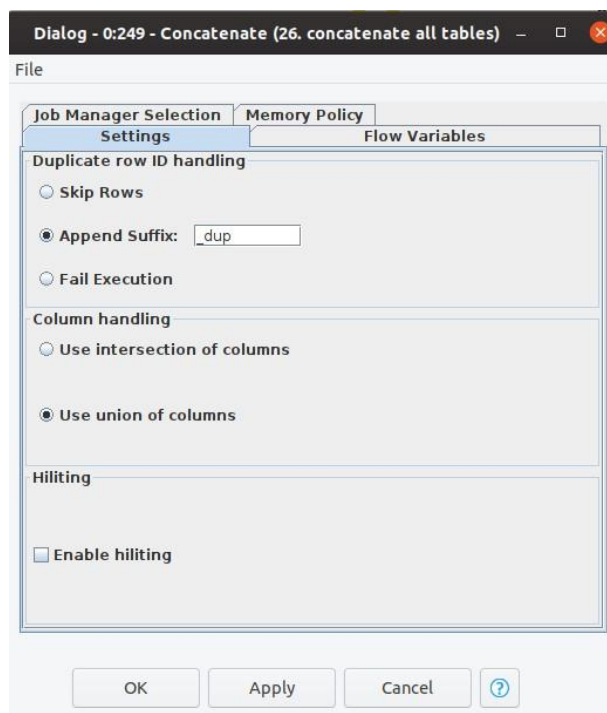
Nodes 23, 24 and 25, String Manipulation

Same as node 8. Replace the units M, mM and nM to uM on nodes 23, 24 and 25. New units replace the units column.



Node 26, Concatenate

With all the activities as IC50 and transformed to uM units, we can merge all the tables one after others using the concatenate node. The amount of input ports from of the node can be augmented by clicking con the three dots of the node outside the configuration. There is no need to configure nothing else in this node.



Nodes 27 and 28, Excel Writer

To export our data sets to Excel we can use Excel Writer node. We must choose a path and file name using the browse button. If we want to keep our column names we can check mark "write column headers", and if we pretend to execute the workflow multiple times we can choose to overwrite or append the output file.

