Table of Content

Terminology

Foreground: The set of chemicals on which enrichments will be computed
Background: The set of chemicals used by the fisher-test during the enrichment computation
CID: A chemical ID as found in one of the chemical universe
Chemical universe: Set of chemical names associated to CIDs. We use two chemical universe : Stitch and Pubchem
Database: Database used during the enrichment step. Each database contains specific biological terms associated to different chemicals
${CART_INSTALLATION_DIRECTORY} : CART installation directory

Note: By default, if you do not upload and select a background, a default background will be chosen. The default background is the set of chemicals that are in the database you will chose during the enrichment step.

?????
What happens if the user moves the tool?
Need to take that into account.
?????

Introduction

What is CART ?

CART is a software that retrieves biological annotations of chemical sets and computes which are enriched. In a nutshell, the input consists in a list of chemical names and the output is a table of enriched biological terms that are associated to these chemical names. CART consists of three modules: Name Matching, Enrichment, Visualization.

How to use it ?

In order to retrieve the enriched biological terms of a list of chemical names, you can either:

1-Run everything at once
2-Run it step by step

Those two modes will produce the same results (output files) for a given list of chemical names (input). Both have pros and cons.

	Pros	cons
Run all at once	Siet variables in one file and execute that file. Does not require the execution of several scripts. It is very straight.	You cannot analyse intermediate outputs of different modules and thus cannot modify them to refine results and then pursue the workflow.
Run step by step	You can analyse intermediate outputs of different modules and thus can modify them to refine results and then pursue the workflow.	Need to execute three different script in a rows before getting the final results (html output files)

Table 1

How does it work ?

All in all, CART functions as follow:

1- Input your chemical names
2- CART will fetch their CIDs by matching them against a chemical universe
3- With these CIDs, CART will retrieve which biological terms are associated to it
4- CART will compute which of these terms are enriched in the inputted chemicals
5- CART will produce a table with links linking terms to external web resources as well as an interactive network linking chemicals to their enriched terms

CART's workflow

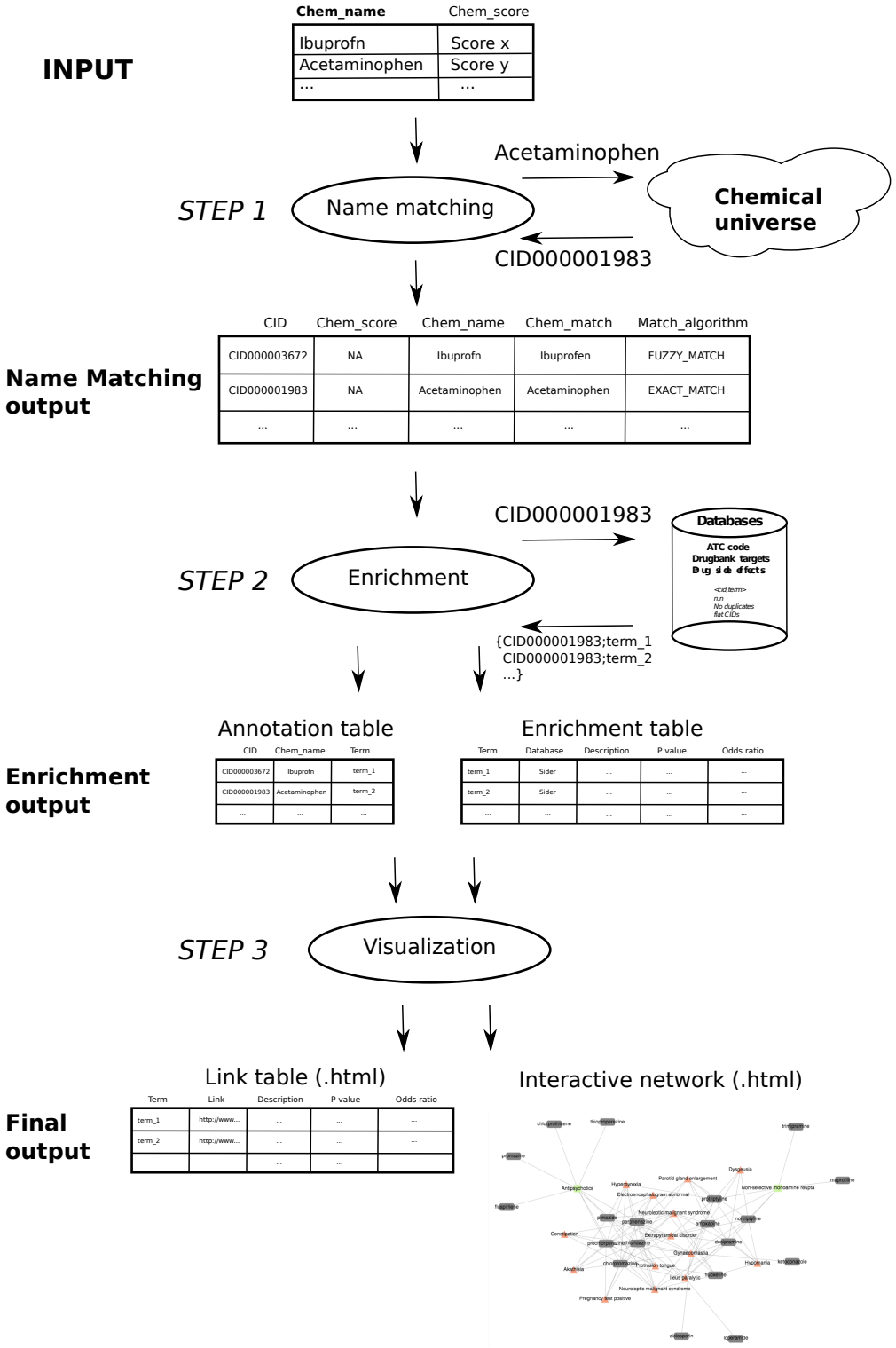

Figure 1 : Schematic representation of CART's workflow

1- Run everything at once

If you wish to run the entire workflow displayed in the Figure 1 all at once, follow the next step.

Running CART

Script to run : RUN_EXAMPLE.sh
Script location : ${CART_INSTALLATION_DIRECTORY}/test

```
> cd ${CART_INSTALLATION_DIRECTORY}/test
```

Open the script RUN_EXAMPLE.sh and set the appropriate variables according to the table 1.

```
> sh ./RUN_EXAMPLE.sh
```

Results

Your results are in the directory:
${CART_INSTALLATION_DIRECTORY}/test/${out_dir}
NB : **${out_dir}** corresponds to the variable `**out_dir**` defined in the script RUN_EXAMPLE.sh.

Variable name	REQUIRED ?	Default value	Description
fn_fg	YES	No default value	The absolute path of your foreground
fn_bg	OPTIONAL	NULL	The absolute path of your background.
out_nm_fg	OPTIONAL	"out-nm-${fn_fg}"	Name matching output of foreground chemicals
out_nm_bg	OPTIONAL	''NULL"	Name matching output of Background chemicals
fuzzy	OPTIONAL	true	Fuzzy name matching algorithm
heuristic	OPTIONAL	true	Heuristic name matching algorithm
universe	OPTIONAL	STITCH	Chemical universe
dbs	OPTIONAL	drug-ATC-code-L3,STITCH-drug-targets,drug-side-effects	The databases that you wish to use during the enrichment step
alpha	OPTIONAL	0.05	Enrichment significance level

method	OPTIONAL	Fisher	Method used for the enrichment calculation
correction	OPTIONAL	FDR	Correction test used for enrichment calculation
out_enr_enr_final	OPTIONAL	out_enr_enr_final	Enrichment table outputted y enrichment module
out_enr_ann_final	OPTIONAL	out_enr_an_final	Annotation table outputted by enrichment module
cart_interactive_table	OPTIONAL	cart-interactive-table.html	HTML file representation of the Enrichment table containing HTML links
cart_interactive_network	OPTIONAL	cart-interactive-network.html	HTML file containing the interactive network
verbose_level	OPTIONAL	2	Verbose level
synonym_option	OPTIONAL	False	Synonym option
out_dir	OPTIONAL	example_out	Output results directory

Table 2

2- Run step by step

If you wish to run the entire workflow displayed in the Figure 1 step by step, follow the next step.

Run the name matching module

What does it do?

The Name Matching aims at fetching CIDs so that the Enrichment module can use them in order to retrieve annotation terms from any incorporated database. The Name Matching matches chemical names against a universe of chemical names according to three different matching algorithms.

Script to run : name_matching.py
Script location : ${CART_INSTALLATION_DIRECTORY}/src
The name matching module expects 6 arguments:

```
> cd ${CART_INSTALLATION_DIRECTORY}/src
```

Argument name	REQUIRED?	Example	Default value	Description
-n	YES	fg_input	No default value	The absolute path of a **chemical name file properly formatted**
-o	YES	nm_fg_output	No default value	Name matching output (corresponding to background)
-u	OPTIONAL	STITCH	STITCH	Universe used for the name matching
-a	OPTIONAL	true	true	Fuzzy name matching algorithm
-e	OPTIONAL	true	true	Heuristic name matching algorithm
-s	OPTIONAL	true	false	Additional synonym output
--verbose	OPTIONAL	2	2	Verbose level

Table 3

> python name_matching.py -n <chemical_name_file_foreground> -o <chemical_name_file_output_foreground> -a <fuzzy_option_value> -e $<heuristic_option_value> -s $<synonym_option_value> --verbose $<verbose_level_option_value>

NB: If you which to perform your analysis with a background, repeat this step by replacing the name matching input argument with the proper background and by replacing the name matching output argument by a different name than the one used for the foreground name matching output.

Run the enrichment module

What does it do ?

The fetched CIDs of the foreground are matched against the collected databases and corresponding annotation terms are retrieved. If background is provided, the same operation is repeated. If no background is provided, the default background is set to the CID found in the database(s) chosen for the enrichment tool, and thus, all the enrichment terms contained in the databases are retrieved. A fisher test is then computed on each annotation term retrieved and a correction method for multiple hypotheses is applied. Annotation terms yielding a p-value inferior to the type I error cut off (by default 0.05) are retained and displayed to the user.

Script to run : enrichment_calculation.py
Script location : ${CART_INSTALLATION_DIRECTORY}/src
The enrichment module expects 9 arguments:

Argument name	REQUIRED?	Example	Default value	Description
-f	YES	nm_fg_output	No default value	name matching output corresponding to foreground
-b	OPTIONAL	ALL	ALL	Either name matching output corresponding to background, or "ALL"
-d	YES	drug-side-effects	No default value	Database form which biological terms will be extracted
-o	YES	enr-enr-output	No default value	Enrichment module output containing the enriched biological terms
-p	YES	enr-ann-output	No default value	Enrichment module output containing the chemical/biological terms annotations
-m	OPTIONAL	fisher	fisher	Method used to perform the enrichment
-a	OPTIONAL	0.05	0.05	Enrichment significance level
-c	OPTIONAL	FDR	FDR	FDR correction method

--versbose	OPTIONAL	2	2	Verbose level

Table 4

```
>  python enrichment_calculation.py -f <nm_fg_output> -b <'ALL'> -d
<'drug-side-effects'> -o <"enr-enr-output"> -p <"enr-ann-output"> -a
<"0.05"> --verbose <"2">
```

Run the visualization module

What does it do?

It produces a table of the found enrichments with links to external resources. It also produces an interactive networks

Generate annotation table

Script to run: result_annotator.py
Script location: ${CART_INSTALLATION_DIRECTORY}/src
The table generator expects 2 arguments

Argument name	REQUIRED	Example	Default value	Description
-i	YES	enr-enr-output	No default value	The enrichment table outputted by the enrichment module
-o	YES	res-viz-tab.html	No default value	File name (.html) where the table containing links to external resources will be generated

Table 5

```
> python result_annotator.py -i <nm_fg_output> -o <'ALL'>
```

Generate network visualization

Script to run : network_generator.py
Script location : ${CART_INSTALLATION_DIRECTORY}/src
The network generator expects 6 arguments

Argument name	Needs to be set?	Example	Default value	Description
-a	REQUIRED	enr-ann-output	No default value	The annotation table outputted by the enrichment module
-e	REQUIRED	enr-enr-output	No default value	The enrichment table outputted by the enrichment module
-o	OPTIONAL	res-viz-network.html	No default value	File name (.html) where the dynamic network will be generated. Open this file with a web browser to see the results.

> python network_generator.py –a <enr-ann-output> –e <enr-enr-output> –d <description_file> –k <head_js> –f <footer_js> -o <res-viz-network.html>

More about each step

Data input

Each uploaded file must contains at at least one column and at most 2 columns separated by a tab characted. The first column holds the chemical names. The second column can hold a numerical value.

Name Matching

What it does: The Name Matching aims at fetching CIDs so that the Enrichment module can use them in order to retrieve annotation terms from any incorporated database. The Name Matching matches chemical names against a universe of chemical names according to three different matching algorithms.

The exact matching looks for chemicals corresponding exactly to the inputted name and return a NA if no matches has been found.
The fuzzy matching tolerates mistakes including missing misspelt chemicals. For instance, if the user inputs 'acetominophen' instead of 'acetaminophen', the name matching will still return the correct CID.
The heuristic matching removes part(s) of the chemical names that are found in many chemicals and considered as non-informative. For instance "hcl", "dihydrochloride", "hydrochloride", "salt", "potassium", "dehydrate, "acid", "oxide" and "chloride".

Input : A file which you have uploaded with the Data upload module respecting the format described in Data upload.
Output : A file containing 5 columns delimited by tab characters. The first column holds the fetched CID. The second column corresponds to the second column of the input file. This column is replaced by NA values if no second column is found in the input file. The third column corresponds to the first column of the input file. The fourth column corresponds to the matched chemicals. The fifth column corresponds to the matching algorithm used to fetch the CID

Note: At least one file (foreground) needs to be run through the name matching before pursuing to the enrichment step ! If you which to use a background during the enrichment step, repeat the Name Matching step with a background

Enrichment

What it does: the fetched CIDs of the foreground are matched against the collected databases and corresponding annotation terms are retrieved. If background is provided, the same operation is repeated. If no background is provided, the default background is set to the CID found in the database(s)

chosen for the enrichment tool, and thus, all the enrichment terms contained in the databases are retrieved. A fisher test is then computed on each annotation term retrieved and a correction method for multiple hypotheses is applied. Annotation terms yielding a p-value inferior to the type I error cut off (by default 0.05) are retained and displayed to the user.

Input: A first input corresponding to the output of the name matching tool is provided. This is the foreground. Optionally, a second input can be provided and will be considered as the background.
You then need to select one database of interest.

Output: Two outputs files are provided. The first output is a tab-delimited file, this is the `enrichment output file`. It contains the enriched terms the enriched terms along with database of origin, links to external resources, p-values and odd-ratios. The second output is another tab-delimited file, this is the `annotation output file`. It contains all the annotation terms that have been retrieved for the foreground chemicals using the selected resources

Visualization

What it does: It produces a table of the found enrichments with links to external resources. It also produces an interactive networks

Input: The two outputed files from the enrichment module
Output: An html table with links of the enriched terms to external resources. A link to a web page displaying an interactive networks of the foreground chemicals associated to their enriched terms.

Synonyms

What it does: It retrieves the synonyms of an inputed file formatted as mentioned in the Data Upload part.

Input: An input file containing chemical names
Output: A tab delimited file contaiing three column. The first column contains the user's chemical names. The second column contains the synonym found. The third column contains the CIDs associated to these chemicals.