
ComPath Documentation

Release 0.1.3

Daniel Domingo-Fernández and Charles Tapley Hoyt

Sep 29, 2020

CONTENTS:

1	How to Use	3
2	Mapping across gene/protein identifiers	5
3	Repositories	7
4	Command Line Interface	9
4.1	compath	9
5	Manager	13
6	Models	19
7	Constants	23
8	Curation	25
9	Visualization	27
9.1	Network visualization	27
9.2	Dendrogram visualization	28
9.3	Venn Diagram visualization	29
10	Web	31
11	Utils	33
12	ComPath License	39
13	Indices and tables	41
	Python Module Index	43
	Index	45

An interactive and extensible web application for comparative analysis of pathway databases. ComPath is available at <https://compath.scai.fraunhofer.de/>.

This package exposes the Bio2BEL pathway packages into a web application containing multiple built-in visualization and analytics tools allowing for their analysis and comparison. By default, this packages wraps the following default packages:

- [Bio2BEL KEGG](#)
- [Bio2BEL Reactome](#)
- [Bio2BEL WikiPathways](#)
- [Bio2BEL MSig](#)

New pathway/gene signatures resources can be added by forking the [ComPath Template Repository](#).

HOW TO USE

1. Install and load the required packages

- If you just cloned the repo, you can run the sh script “load_compath.sh” by typing `sh load_compath.sh` in your terminal. This script will first install all packages and later populate the database.
- If you have already installed the packages, but not loaded the data. First, load **Bio2BEL HGNC** (see next section). Next, load all individual pathway database packages KEGG, Reactome, WikiPathways, and MSigDB with `python3 -m compath populate`. This command assumes that these packages are already installed in your Python environment. You can check the packages installed by running `python3 -m compath ls` in your terminal. Alternatively, you can populate each package independently by running: `python3 -m bio2bel_kegg populate`, `python3 -m bio2bel_reactome populate`, `python3 -m bio2bel_wikipathways populate`, or `python3 -m bio2bel_msig populate`.

2. The final step is to start the web application locally (runs by default on port 5000 -> <http://localhost:5000>) by running `python3 -m compath web`

MAPPING ACROSS GENE/PROTEIN IDENTIFIERS

In order to load the gene sets from default packages, ComPath assumes that [Bio2BEL HGNC](#) has been already installed and populated. This package is required to perform the mapping from multiple Protein/Gene identifiers to HGNC symbols. The following steps are needed to install Bio2BEL HGNC:

1. `python3 -m pip install bio2bel_hgnc`
2. `python3 -m bio2bel_hgnc populate`

REPOSITORIES

The following is an incomplete listing of the Bio2BEL repositories enabled for use with ComPath.

Source	Version	Documentation	Travis	DOI
ComPath HGNC				
KEGG				
Reactome				
WikiPathways				
MSigDB				

COMMAND LINE INTERFACE

4.1 compath

ComPath at `sqlite:///home/docs/.pybel/bio2bel/bio2bel.db`

```
compath [OPTIONS] COMMAND [ARGS]...
```

4.1.1 drop

Drop ComPath DB.

```
compath drop [OPTIONS]
```

Options

-v, --debug

Turn on debugging.

-y, --yes

-c, --connection <connection>

Defaults to `sqlite:///home/docs/.pybel/bio2bel/bio2bel.db`

4.1.2 drop-databases

Drop all databases.

```
compath drop-databases [OPTIONS]
```

Options

- v, --debug**
Turn on debugging.
- y, --yes**
- c, --connection** <connection>
Defaults to `sqlite:///home/docs/.pybel/bio2bel/bio2bel.db`

4.1.3 load-hierarchies

Load pathway databases with hierarchies.

```
compath load-hierarchies [OPTIONS]
```

Options

- c, --connection** <connection>
Defaults to `sqlite:///home/docs/.pybel/bio2bel/bio2bel.db`
- e, --email** <email>
Default curator: daniel.domingo.fernandez@scai.fraunhofer.de

4.1.4 load-mappings

Load mappings from template.

```
compath load-mappings [OPTIONS]
```

Options

- c, --connection** <connection>
Defaults to `sqlite:///home/docs/.pybel/bio2bel/bio2bel.db`

4.1.5 ls

Display registered Bio2BEL pathway managers.

```
compath ls [OPTIONS]
```

4.1.6 manage

Manage the database.

```
compath manage [OPTIONS] COMMAND [ARGS]...
```

Options

-c, --connection <connection>
Cache connection. Defaults to sqlite:///home/docs/.pybel/bio2bel/bio2bel.db

users

User group.

```
compath manage users [OPTIONS] COMMAND [ARGS]...
```

ls

List all users.

```
compath manage users ls [OPTIONS]
```

make-admin

Make a pre-existing user an admin.

```
compath manage users make-admin [OPTIONS] EMAIL
```

Arguments

EMAIL
Required argument

make-user

Create a pre-existing user an admin.

```
compath manage users make-user [OPTIONS] EMAIL PASSWORD
```

Arguments

EMAIL

Required argument

PASSWORD

Required argument

4.1.7 populate

Populate all registered Bio2BEL pathway packages.

```
compath populate [OPTIONS]
```

Options

-v, --debug

Turn on debugging.

-c, --connection <connection>

Defaults to sqlite:///home/docs/pybel/bio2bel/bio2bel.db

-d, --delete-first

4.1.8 web

Run web service.

```
compath web [OPTIONS]
```

Options

--host <host>

Flask host. Defaults to 0.0.0.0

--port <port>

Flask port. Defaults to 5000

--template-folder <template_folder>

Template folder. Defaults to 'templates'

--static-folder <static_folder>

Template folder. Defaults to 'static'

-v, --debug

Turn on debugging.

-c, --connection <connection>

Defaults to sqlite:///home/docs/pybel/bio2bel/bio2bel.db

MANAGER

This module the database manager of ComPath.

```
class compath.manager.Manager (engine, session)
    Database manager.

    Init ComPath manager.

    static from_connection (connection=None)
        Initialize manager from connection string

    create_all (check_first: bool = True)
        Create tables for ComPath.

    drop_all (check_first: bool = True)
        Drop all tables for ComPath.

    count_votes () → int
        Count the votes in the database.

    count_mappings () → int
        Count the mappings in the database.

    count_users () → int
        Count the Users in the database.

    count_mapping_user () → int
        Count the UsersMappings table in the database.

        Return type int

    get_all_mappings () → List[compath.models.PathwayMapping]
        Get all mappings in the database.

    get_all_accepted_mappings () → List[compath.models.PathwayMapping]
        Get all accepted mappings in the database.

    get_mappings_by_type (mapping_type: str) → List[compath.models.PathwayMapping]
        Get all mappings in the database.

        Parameters mapping_type – type of the mapping

    get_vote_by_id (vote_id: str) → Optional[compath.models.Vote]
        Get a vote by its id.

        Parameters vote_id – identifier

    get_vote (user: compath.models.User, mapping: compath.models.PathwayMapping) → Optional[compath.models.Vote]
        Get a vote.
```

Parameters

- **user** – User instance
- **mapping** – Mapping instance

get_user_by_email (*email: str*) → Optional[*compath.models.User*]

Get a user by their email address.

get_mapping (*service_1_name, pathway_1_id, pathway_1_name, service_2_name, pathway_2_id, pathway_2_name, mapping_type*)

Query mapping in the database.

Parameters

- **service_1_name** (*str*) – manager name of the service 1
- **pathway_1_id** (*str*) – pathway 1 id
- **pathway_1_name** (*str*) – pathway 1 name
- **service_2_name** (*str*) – manager name of the service 1
- **pathway_2_id** (*str*) – pathway 2 id
- **pathway_2_name** (*str*) – pathway 2 name
- **mapping_type** (*str*) – mapping type (isPartOf or equivalentTo)

Return type Optional[Mapping]

get_mapping_by_id (*mapping_id: int*) → Optional[*compath.models.PathwayMapping*]

Get a mapping by its id.

Parameters **mapping_id** – mapping id

get_or_create_vote (*user, mapping, vote_type=True*)

Get or create vote.

Parameters

- **user** (*User*) – User instance
- **mapping** (*PathwayMapping*) – Mapping instance
- **vote_type** (*Optional[Vote.type]*) – vote type

Return type *Vote*

get_or_create_mapping (*service_1_name, pathway_1_id, pathway_1_name, service_2_name, pathway_2_id, pathway_2_name, mapping_type, user*)

Get or create a mapping.

Parameters

- **service_1_name** (*str*) – manager name of the service 1
- **pathway_1_name** (*str*) – pathway 1 name
- **pathway_1_id** (*str*) – pathway 1 id
- **service_2_name** (*str*) – manager name of the service 1
- **pathway_2_name** (*str*) – pathway 2 name
- **pathway_2_id** (*str*) – pathway 2 id
- **mapping_type** (*str*) – type of mapping
- **user** (*User*) – the user

Returns PathwayMapping and boolean indicating if the mapping was created or not

Return type `tuple[PathwayMapping,bool]`

delete_all_mappings ()

Delete all the votes then all the mappings.

delete_mapping_by_id (*mapping_id*)

Delete a mapping by its id.

Parameters *mapping_id* (*int*) – mapping id

Return type `bool`

claim_mapping (*mapping*, *user*)

Check if user has already established the mapping, if not claims it.

Parameters

- **mapping** (*PathwayMapping*) – Mapping instance
- **user** (*User*) – User

Return type `bool`

Returns if mapping was assigned to user

accept_mapping (*mapping_id*)

Accept established mapping (from user or curator consensus).

Parameters *mapping_id* (*int*) – mapping id

Return type `tuple[PathwayMapping,bool]`

Returns mapping and boolean that indicates if transaction was made

get_mappings_from_pathway_with_relationship (*type*, *service_name*, *pathway_id*, *pathway_name*)

Get all mappings matching pathway and service name.

Parameters

- **type** (*str*) – mapping type
- **service_name** (*str*) – service name
- **pathway_id** (*str*) – original pathway identifier
- **pathway_name** (*str*) – pathway name

Return type `list[PathwayMapping]`

Returns

get_descendents_mappings_from_pathway_with_is_part_of_relationship (*service_name*, *pathway_id*, *pathway_name*)

Get all mappings matching pathway and service name.

Parameters

- **type** (*str*) – mapping type
- **service_name** (*str*) – service name
- **pathway_id** (*str*) – original pathway identifier

- **pathway_name** (*str*) – pathway name

Return type `list[PathwayMapping]`

Returns

get_ancestry_mappings_from_pathway_with_is_part_of_relationship (*service_name*,
pathway_id,
pathway_name)

Get all mappings matching pathway and service name.

Parameters

- **type** (*str*) – mapping type
- **service_name** (*str*) – service name
- **pathway_id** (*str*) – original pathway identifier
- **pathway_name** (*str*) – pathway name

Return type `list[PathwayMapping]`

Returns

get_all_mappings_from_pathway (*service_name*, *pathway_id*, *pathway_name*)

Get all mappings matching pathway and service name.

Parameters

- **service_name** (*str*) – service name
- **pathway_id** (*str*) – original pathway identifier
- **pathway_name** (*str*) – pathway name

Return type `list[PathwayMapping]`

Returns

get_all_pathways_from_db_with_mappings (*pathway_database*)

Get all mappings that contain a pathway from a given database.

Parameters

- **service_name** (*str*) – service name
- **pathway_id** (*str*) – original pathway identifier
- **pathway_name** (*str*) – pathway name

Return type `list[PathwayMapping]`

Returns

infer_hierarchy (*resource*, *pathway_id*, *pathway_name*)

Infer the possible hierarchy of a given pathway based on its equivalent mappings.

Parameters

- **type** (*str*) – mapping type
- **resource** (*str*) – service name
- **pathway_id** (*str*) – pathway original identifier
- **pathway_name** (*str*) – pathway name

Returns

MODELS

ComPath database model.

```
class compath.models.User (**kwargs)
```

User table.

A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and values in `kwargs`.

Only keys that are present as attributes of the instance's class are allowed. These could be, for example, any mapped columns or relationships.

```
property is_admin
```

Is this user an administrator?.

```
class compath.models.Role (**kwargs)
```

Role table.

A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and values in `kwargs`.

Only keys that are present as attributes of the instance's class are allowed. These could be, for example, any mapped columns or relationships.

```
class compath.models.PathwayMapping (**kwargs)
```

Mapping table.

A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and values in `kwargs`.

Only keys that are present as attributes of the instance's class are allowed. These could be, for example, any mapped columns or relationships.

```
service_1_name
```

service name (e.g., KEGG or Reactome)

```
service_1_pathway_id
```

pathway 1 id

```
service_1_pathway_name
```

pathway 1 name

```
service_2_name
```

service name (e.g., KEGG or Reactome)

```
service_2_pathway_id
```

pathway 2 id

service_2_pathway_name

pathway 2 name

type

Type of Mapping (isPartOf or equivalentTo)

accepted

accepted mapping by the admin/curator consensus

get_complement_mapping_info (*service_name, pathway_id, pathway_name*)

Return the info corresponding to the other pathway in a mapping.

Parameters

- **mapping** (*PathwayMapping*) –
- **service_name** (*str*) – reference service name
- **pathway_id** (*str*) – reference pathway id
- **pathway_name** (*str*) – reference pathway name

Return type *tuple*[*str*,*str*,*str*]

static has_pathway_tuple (*type, service_name, pathway_id, pathway_name*)

Return a filter to get all mappings matching type, service and pathway name and id.

static has_descendant_pathway_tuple (*type, service_name, pathway_id, pathway_name*)

Return a filter to get all the descendants mappings matching a isPartOf relationship (the predicates), service and pathway name and id.

static has_ancestry_pathway_tuple (*type, service_name, pathway_id, pathway_name*)

Return a filter to get all the ancestries mappings matching a isPartOf relationship (the subjects), service and pathway name and id.

static has_pathway (*service_name, pathway_id, pathway_name*)

Return a filter to get all mappings matching service and pathway name and id.

static has_database_pathway (*service_name*)

Return a filter to get all mappings matching service a name.

property count_votes

Return the number of votes for this mapping.

Return type *int*

property count_creators

Return the number of creator that claimed this mapping.

Return type *int*

property count_up_votes

Return the number of up votes for this mapping.

Return type *int*

property count_down_votes

Return the number of down votes for this mapping.

Return type *int*

property is_acceptable

Return true if the mapping has enough votes to be accepted.

Return type *bool*

get_user_vote (*user*)

Return votes given by the user.

class `compath.models.Vote` (***kwargs*)

Vote table.

A simple constructor that allows initialization from kwargs.

Sets attributes on the constructed instance using the names and values in *kwargs*.

Only keys that are present as attributes of the instance's class are allowed. These could be, for example, any mapped columns or relationships.

type

Type of vote, by default is up-vote

CONSTANTS

Constants for ComPath.

This module contains all the string constants used in ComPath.

```
compath.constants.REACTOME_GENE_SET = '/home/docs/checkouts/readthedocs.org/user_builds/compath/kegg/kegg.csv'
    CSV file for Reactome gene set

compath.constants.KEGG_GENE_SET = '/home/docs/checkouts/readthedocs.org/user_builds/compath/kegg/kegg.csv'
    CSV file for KEGG gene set

compath.constants.WIKIPATHWAYS_GENE_SET = '/home/docs/checkouts/readthedocs.org/user_builds/compath/kegg/kegg.csv'
    CSV file for WikiPathways gene set

compath.constants.MSIG_GENE_SET = '/home/docs/checkouts/readthedocs.org/user_builds/compath/kegg/kegg.csv'
    CSV file for MSigDB gene set

compath.constants.VOTE_ACCEPTANCE = 3
    Minimum number of votes to accept a mapping

compath.constants.KEGG = 'kegg'
    KEGG

compath.constants.REACTOME = 'reactome'
    Reactome

compath.constants.WIKIPATHWAYS = 'wikipathways'
    WikiPathways

compath.constants.MSIG = 'msig'
    MSigDB

compath.constants.KEGG_URL = 'http://www.kegg.jp/kegg-bin/show_pathway?map=map{}&show_desc=1'
    REST API to KEGG

compath.constants.REACTOME_URL = 'https://reactome.org/PathwayBrowser/#/{}'
    URL to pathways in Reactome

compath.constants.WIKIPATHWAYS_URL = 'https://www.wikipathways.org/index.php/Pathway:{}'.format('{}')
    URL to pathways in WikiPathways

compath.constants.HIERARCHY_MANAGERS = {'reactome'}
    Managers with hierarchical information

compath.constants.BLACK_LIST = {'compath_hgnc', 'hgnc'}
    Managers without pathway knowledge

compath.constants.MAPPING_TYPES = {'equivalentTo', 'isPartOf'}
    Possible mapping types between pathways
```

```
compath.constants.STYLED_NAMES = {'compath_neurommsig_ad': 'NeuroMMSig AD', 'compath_neur  
    Mappings for pathway database  
compath.constants.EQUIVALENT_TO = 'equivalentTo'  
    Possible mapping types between pathways
```

CURATION

Modules related with the ComPath curation exercise.

This module loads the hierarchical pathway databases into ComPath.

```
compath.curation.hierarchies.create_hierarchical_mappings (pathways,          com-  
                                                         path_manager,  
                                                         pathway_database,  
                                                         curator)
```

Iterate over pathway objects and creates hierarchies if exist.

Parameters

- **pathways** (*list*[*Pathway*]) – list of pathways
- **compath_manager** (*compath.manager.Manager*) – ComPath Manager
- **pathway_database** (*str*) – Name of the pathway database
- **curator** (*compath.models.User*) – Curator user

Returns number of created mappings

Return type *int*

```
compath.curation.hierarchies.load_hierarchy (*, connection=None, curator_email=None)
```

Load the hierarchical relationships for the managers containing them.

Parameters

- **connection** (*Optional*[*str*]) –
- **curator_email** (*Optional*[*str*]) – email of the admin

This module contains all the curation processing methods to integrate curated mappings to ComPath.

```
compath.curation.parser.get_users_from_emails (compath_manager, curator_emails)
```

Return a list with curator users.

Parameters

- **compath_manager** (*Manager*) –
- **curator_emails** (*list*[*str*]) –

Return type *list*[*User*]

```
compath.curation.parser.parse_special_mappings (path, curator_emails=None, connec-  
                                                tion=None)
```

Parse special mapping file.

Parameters

- **path** (*str*) – path of the excel sheet
- **curator_emails** (*Optional[list]*) – emails of the curators. Needs to be already in the database

```
compath.curation.parser.parse_curation_template(path, reference_pathway_db,
                                                compared_pathway_db, curator_emails=None, connection=None)
```

Load the curation template excel sheet into a pandas Dataframe.

Parameters

- **path** (*str*) – path of the excel sheet
- **reference_pathway_db** (*str*) – name of the manager of the reference pathway_db
- **compared_pathway_db** (*str*) – name of the manager of the compared pathway_db
- **curator_emails** (*Optional[list]*) – email of the curators. Needs to be already in the database
- **connection** (*str*) – database connection

This module contains all the utils for curation processing.

```
compath.curation.utils.remove_star_from_pathway_name(pathway_name)
```

Remove the star that label the reference pathway in isPartOf statements.

Parameters **statements** (*str*) – pathway name

```
compath.curation.utils.get_pathways_from_statement(mapping_statement, mapping_type)
```

Return the subject, object of the mapping.

Parameters

- **mapping_statement** (*str*) – statement
- **mapping_type** (*str*) – type of relationship

Return type *tuple[str,str]*

```
compath.curation.utils.is_valid_pathway_name(manager_dict, resource, pathway_name)
```

Check if pathway name exists in pathway database.

Parameters

- **manager_dict** (*dict*) – manager name to manager instances dictionary
- **resource** (*str*) – resource name
- **pathway_name** (*str*) – pathway name

Return type *bool*

```
compath.curation.utils.is_valid_pathway_by_id(manager_dict, resource, pathway_id)
```

Check if pathway identifier exists in pathway database.

Parameters

- **manager_dict** (*dict*) – manager name to manager instances dictionary
- **resource** (*str*) – resource name
- **pathway_id** (*str*) – pathway identifier

Return type *bool*

VISUALIZATION

All methods used to perform the visualizations.

9.1 Network visualization

Utils to generate the Cytoscape.js network.

`compath.visualization.cytoscape.filter_network_by_similarity` (*graph*,
min_similarity)

Remove edges with similarity less than the minimum given.

Parameters

- **graph** (*networkx.Graph*) – graph
- **min_similarity** (*float*) – minimum similarity required to keep an edge

`compath.visualization.cytoscape.pathways_to_similarity_network` (*manager_dict*,
pathways)

Create a graph with the given pathways related by their similarity

Parameters

- **manager_dict** (*dict*) –
- **pathways** (*list[tuple(str, str, str)]*) –

Return type `networkx.Graph`

`compath.visualization.cytoscape.enrich_graph_with_mappings` (*graph*, *mappings*)

Enrich a graph with the mapping information.

Parameters

- **networkx.Graph** (*graph*) –
- **mappings** (*iter*) –

`compath.visualization.cytoscape.networkx_to_cytoscape_js` (*graph*)

Convert a networkx graph to the cytoscape json format.

Parameters **graph** (*iter[PathwayMapping]*) –

Return type `dict`

9.2 Dendrogram visualization

Utils to generate the D3.js dendrogram. This module is adapted from <https://gist.github.com/mdml/7537455>.

`compath.visualization.d3_dendrogram.create_similarity_matrix(gene_sets)`

Create a similarity matrix for a given pathway-geneset dataset.

Parameters `gene_sets` (*dict*) – pathway gene set dictionary

Return type `pandas.DataFrame`

Returns similarity matrix

`compath.visualization.d3_dendrogram.add_node(node, parent)`

Create a nested dictionary from the ClusterNode's returned by SciPy.

Parameters

- **node** –
- **parent** (*dict*) –

`compath.visualization.d3_dendrogram.label_tree(id_name_dict, name_manager_dict, cluster_to_x, tree)`

Label the tree in a recursive way with names, resource and cluster information.

Parameters

- **id_name_dict** (*dict*[*str*, *str*]) – node_id to name dictionary
- **name_manager_dict** (*dict*[*str*, *str*]) – node name to resource dictionary
- **cluster_to_x** (*dict*[*tuple*[*int*, *int*], *float*]) – node_id tuple of the cluster to distance
- **tree** (*dict*) – tree like structure

Return type `list`

`compath.visualization.d3_dendrogram.get_dendrogram_tree(gene_sets, pathway_manager_dict)`

Return ready to plot json data.

Parameters

- **gene_sets** (*dict*[*str*, *set*[*str*]]) – pathway gene sets dict
- **pathway_manager_dict** (*dict*[*str*, *str*]) – pathway name to manager dictionary

Return type `tuple`[*dict*, *int*]

Returns json tree like structure

`compath.visualization.d3_dendrogram.get_descendants(manager, resource, pathway_id, pathway_name)`

Generate d3 dendrogram structure by using BFS starting from the starting from a parent (root) node to the last descendants.

Parameters

- **manager** – ComPath manager
- **resource** (*str*) – resource name
- **pathway_id** (*str*) – pathway identifier in the resource
- **pathway_name** (*str*) – pathway name

Returns parent-children data structure

Return type `list[dict]`

```
compath.visualization.d3_dendrogram.get_mapping_dendrogram(manager, resource,
                                                         pathway_id, pathway_name)
```

Generate d3 dendrogram structure by using BFS starting from the queried node in both directions of the hierarchy.

Parameters

- **manager** – ComPath manager
- **resource** (*str*) – resource name
- **pathway_id** (*str*) – pathway identifier in the resource
- **pathway_name** (*str*) – pathway name

Returns parent-children data structure

Return type `list[dict]`

```
compath.visualization.d3_dendrogram.add_mapping_node(id_name_dict,
                                                    name_manager_dict, cluster_to_x, tree)
```

Label the tree in a recursive way with names, resource and cluster information.

Parameters

- **id_name_dict** (*dict[str, str]*) – node_id to name dictionary
- **name_manager_dict** (*dict[str, str]*) – node name to resource dictionary
- **cluster_to_x** (*dict[tuple[int, int], float]*) – node_id tuple of the cluster to distance
- **tree** (*dict*) – tree like structure

Return type `list`

9.3 Venn Diagram visualization

Utils to generate the Venn Diagram.

```
compath.visualization.venn_diagram.process_overlap_for_venn_diagram(gene_sets,
                                                                    skip_gene_set_info=False)
```

Calculate gene sets overlaps and process the structure to render venn diagram -> <https://github.com/benfred/venn.js/>.

Parameters

- **gene_sets** (*dict[str, set]*) – pathway to gene sets dictionary
- **skip_gene_set_info** (*bool*) – include gene set overlap data

Returns `list[dict]`

This module contains the ComPath web application.

```
class compath.web.ComPathSQLAlchemy (app=None, use_native_unicode=True, session_options=None, metadata=None,  
                                     query_class=<class 'flask_sqlalchemy.BaseQuery'>, model_class=<class 'flask_sqlalchemy.model.Model'>)  
    ComPath.  
    init_app (app)  
        Overwrite init app method.  
compath.web.create_app (connection=None, template_folder=None, static_folder=None)  
    Create the Flask application.  
    Return type flask.Flask
```


UTILS

This module contains miscellaneous methods.

`compath.utils.log = <Logger compath.utils (WARNING)>`
General utils

`compath.utils.dict_to_pandas_df(d)`
Transform pandas df into a dict.

Parameters `d` (*dict*) –

Return type `pandas.DataFrame`

Returns pandas dataframe

`compath.utils.process_form_gene_set(text)`
Process the string containing gene symbols and returns a gene set.

Parameters `text` (*str*) – string to be processed

Return type `set[str]`

Returns gene set

`compath.utils.calculate_relative_enrichments(results, total_pathways_by_resource)`
Calculate relative enrichment of pathways (enriched pathways/total pathways).

Parameters

- **results** (*dict*) – result enrichment
- **total_pathways_by_resource** (*dict*) – resource to number of pathways

Return type `dict`

`compath.utils.count_genes_in_pathway(pathways_gene_sets, genes)`
Calculate how many of the genes are associated to each pathway gene set.

Parameters

- **pathways_gene_sets** (*dict*) – pathways and their gene sets
- **genes** (*set*) – genes queried

Return type `dict`

`compath.utils.apply_filter(results, threshold)`
Run one simulation with a given threshold.

Parameters

- **results** (*dict*) – resource with pathways

- **threshold** (*int*) – necessary genes to enrich a pathway

Return type *dict*

`compath.utils.simulate_pathway_enrichment` (*resource_gene_sets*, *gene_set_query*,
runs=200)

Simulate pathway enrichment.

Parameters

- **resource_gene_sets** – resource and their gene sets
- **gene_set_query** – shared genes between all resources
- **runs** – number of simulation

Return type *dict[list[tuple]]*

`compath.utils.get_genes_without_assigned_pathways` (*enrichment_results*, *genes_query*)

Return the genes without any known pathway assigned.

Parameters

- **gene_set** (*dict*) – list of managers
- **genes_query** (*set[str]*) – gene set queried

Returns

`compath.utils.get_enriched_pathways` (*manager_list*, *gene_set*)

Return the results of the queries for every registered manager.

Parameters

- **Manager** *manager_list* (*dict[str, ...]*) – list of managers
- **gene_set** (*set[str]*) – gene set queried

Return type *dict[str, dict[str, dict]]*

`compath.utils.get_gene_pathways` (*manager_list*, *gene*)

Return the pathways associated with a gene for every registered manager.

Parameters

- **Manager** *manager_list* (*dict[str, ...]*) – list of managers
- **gene** (*str*) – HGNC symbol

Return type *dict[str, dict[str, dict]]*

`compath.utils.get_mappings` (*compath_manager*, *only_accepted=True*)

Return a pandas dataframe with mappings information as an excel sheet file.

Parameters

- **compath_manager** (*compath.manager.Manager*) – ComPath Manager
- **only_accepted** (*bool*) – only accepted (True) or all (False)

`compath.utils.get_pathway_model_by_name` (*manager_dict*, *resource*, *pathway_name*)

Return the pathway object from the resource manager.

Parameters

- **manager_dict** (*dict*) – manager name to manager instances dictionary
- **resource** (*str*) – name of the manager
- **pathway_name** (*str*) – pathway name

Return type Optional[Pathway]

Returns pathway if exists

`compath.utils.get_pathway_model_by_id(app, resource, pathway_id)`

Return the pathway object from the resource manager.

Parameters

- **app** (*flask.Flask*) – current app
- **resource** (*str*) – name of the manager
- **pathway_id** (*str*) – pathway id

Return type Optional[Pathway]

Returns pathway if exists

`compath.utils.get_gene_sets_from_pathway_names(app, pathways)`

Return the gene sets for a given pathway/resource tuple.

Parameters

- **app** (*flask.Flask*) – current app
- **pathways** (*list[tuple[str, str]]*) – pathway/resource tuples

Return type *tuple[dict[str, set[str]], dict[str, str]]*

Returns gene sets

`compath.utils.get_pathway_info(app, pathways)`

Return the gene sets for a given pathway/resource tuple.

Parameters

- **app** (*flask.Flask*) – current app
- **pathways** (*list[tuple[str, str]]*) – pathway/resource tuples

Return type *list*

Returns pathway info

`compath.utils.get_last_action_in_module(module_name, action)`

Return the info about the last action in the given module.

Parameters **module_name** (*str*) –

Returns

`compath.utils.perform_hyergeometric_test(gene_set, manager_pathways_dict,
gene_universe, apply_threshold=False, thresh-
old=0.05)`

Perform hypergeometric tests.

Parameters

- **gene_set** (*set[str]*) – gene set to test against pathway
- **manager_pathways_dict** (*dict[str, dict[str, dict]]*) – manager to pathways
- **gene_universe** (*int*) – number of HGNC symbols
- **apply_threshold** (*Optional[bool]*) – return only significant pathways
- **threshold** (*Optional[float]*) – significance threshold (by default 0.05)

Return type `dict[str,dict[str,dict]]`

Returns `manager_pathways_dict` with p value info

`compath.utils.calculate_szymkiewicz_simpson_coefficient(set_1, set_2)`

Calculate Szymkiewicz-Simpson coefficient between two sets.

Parameters

- **set_1** (`set`) – set 1
- **set_2** (`set`) – set 2

Returns similarity of the two sets

Return type `float`

`compath.utils.calculate_similarity(name_1, name_2)`

Calculate the string based similarity between two names.

Parameters

- **name_1** (`str`) – name 1
- **name_2** (`str`) – name 2

Return type `float`

Returns Levenshtein similarity

`compath.utils.get_top_matches(matches, top)`

Order list of tuples by second value and returns top values.

Parameters

- **matches** (`list[tuple[str, float]]`) – list of tuples
- **top** (`int`) – top values to return

`compath.utils.filter_results(results, threshold)`

Filter a tuple based iterator given a threshold.

Parameters

- **results** (`list[tuple[str, float]]`) – list of tuples
- **threshold** (`float`) – thresholding

`compath.utils.get_most_similar_names(reference_name, names, threshold=0.4, top=5)`

Return the most similar names based on string matching.

Parameters

- **reference_name** (`str`) –
- **names** (`list[str]`) –
- **threshold** (`optional[float]`) –
- **top** (`optional[int]`) –

Returns

`compath.utils.to_csv(triplets, file=None, sep='\t')`

Writes triplets as a tab-separated.

Parameters

- **triplets** (`iterable[tuple[str, str, str]]`) – iterable of triplets

- **file** (*file*) – A writable file or file-like. Defaults to stdout.
- **sep** (*str*) – The separator. Defaults to tab.

COMPATH LICENSE

MIT License

Copyright (c) 2018 Fraunhofer-Gesellschaft e. V. Munich, Germany. All rights reserved.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`

PYTHON MODULE INDEX

C

- `compath, ??`
- `compath.constants`, [23](#)
- `compath.curation.hierarchies`, [25](#)
- `compath.curation.parser`, [25](#)
- `compath.curation.utils`, [26](#)
- `compath.manager`, [13](#)
- `compath.models`, [19](#)
- `compath.utils`, [33](#)
- `compath.visualization.cytoscape`, [27](#)
- `compath.visualization.d3_dendrogram`, [28](#)
- `compath.visualization.venn_diagram`, [29](#)
- `compath.web`, [31](#)

Symbols

--connection <connection>
 compath-drop command line option, 9
 compath-drop-databases command
 line option, 10
 compath-load-hierarchies command
 line option, 10
 compath-load-mappings command line
 option, 10
 compath-manage command line option,
 11
 compath-populate command line
 option, 12
 compath-web command line option, 12
 --debug
 compath-drop command line option, 9
 compath-drop-databases command
 line option, 10
 compath-populate command line
 option, 12
 compath-web command line option, 12
 --delete-first
 compath-populate command line
 option, 12
 --email <email>
 compath-load-hierarchies command
 line option, 10
 --host <host>
 compath-web command line option, 12
 --port <port>
 compath-web command line option, 12
 --static-folder <static_folder>
 compath-web command line option, 12
 --template-folder <template_folder>
 compath-web command line option, 12
 --yes
 compath-drop command line option, 9
 compath-drop-databases command
 line option, 10
 -c
 compath-drop command line option, 9
 compath-drop-databases command

 line option, 10
 compath-load-hierarchies command
 line option, 10
 compath-load-mappings command line
 option, 10
 compath-manage command line option,
 11
 compath-populate command line
 option, 12
 compath-web command line option, 12
 -d
 compath-populate command line
 option, 12
 -e
 compath-load-hierarchies command
 line option, 10
 -v
 compath-drop command line option, 9
 compath-drop-databases command
 line option, 10
 compath-populate command line
 option, 12
 compath-web command line option, 12
 -y
 compath-drop command line option, 9
 compath-drop-databases command
 line option, 10

A

accept_mapping() (*compath.manager.Manager*
 method), 15
 accepted (*compath.models.PathwayMapping* *at-*
 tribute), 20
 add_mapping_node() (*in module compath-*
 path.visualization.d3_dendrogram), 29
 add_node() (*in module compath-*
 path.visualization.d3_dendrogram), 28
 apply_filter() (*in module compath.utils*), 33

B

BLACK_LIST (*in module compath.constants*), 23

C

`calculate_relative_enrichments()` (in module *compath.utils*), 33

`calculate_similarity()` (in module *compath.utils*), 36

`calculate_szymkiewicz_simpson_coefficient()` (in module *compath.utils*), 36

`claim_mapping()` (*compath.manager.Manager* method), 15

compath

- module, 1
- compath.constants* module, 23
- compath.curation.hierarchies* module, 25
- compath.curation.parser* module, 25
- compath.curation.utils* module, 26
- compath.manager* module, 13
- compath.models* module, 19
- compath.utils* module, 33
- compath.visualization.cytoscape* module, 27
- compath.visualization.d3_dendrogram* module, 28
- compath.visualization.venn_diagram* module, 29
- compath.web* module, 31

compath-drop command line option

- `--connection <connection>`, 9
- `--debug`, 9
- `--yes`, 9
- `-c`, 9
- `-v`, 9
- `-y`, 9

compath-drop-databases command line option

- `--connection <connection>`, 10
- `--debug`, 10
- `--yes`, 10
- `-c`, 10
- `-v`, 10
- `-y`, 10

compath-load-hierarchies command line option

- `--connection <connection>`, 10
- `--email <email>`, 10
- `-c`, 10
- `-e`, 10

compath-load-mappings command line option

- `--connection <connection>`, 10
- `-c`, 10

compath-manage command line option

- `--connection <connection>`, 11
- `-c`, 11

compath-manage-users-make-admin command line option

- EMAIL, 11

compath-manage-users-make-user command line option

- EMAIL, 12
- PASSWORD, 12

compath-populate command line option

- `--connection <connection>`, 12
- `--debug`, 12
- `--delete-first`, 12
- `-c`, 12
- `-d`, 12
- `-v`, 12

compath-web command line option

- `--connection <connection>`, 12
- `--debug`, 12
- `--host <host>`, 12
- `--port <port>`, 12
- `--static-folder <static_folder>`, 12
- `--template-folder <template_folder>`, 12
- `-c`, 12
- `-v`, 12

ComPathSQLAlchemy (class in *compath.web*), 31

`count_creators()` (*compath.models.PathwayMapping* property), 20

`count_down_votes()` (*compath.models.PathwayMapping* property), 20

`count_genes_in_pathway()` (in module *compath.utils*), 33

`count_mapping_user()` (*compath.manager.Manager* method), 13

`count_mappings()` (*compath.manager.Manager* method), 13

`count_up_votes()` (*compath.models.PathwayMapping* property), 20

`count_users()` (*compath.manager.Manager* method), 13

`count_votes()` (*compath.manager.Manager* method), 13

`count_votes()` (*compath.models.PathwayMapping* property), 20

`create_all()` (*compath.manager.Manager* method),

- 13
 create_app() (in module *compath.web*), 31
 create_hierarchical_mappings() (in module *compath.curation.hierarchies*), 25
 create_similarity_matrix() (in module *compath.visualization.d3_dendrogram*), 28
- ## D
- delete_all_mappings() (*compath.manager.Manager* method), 15
 delete_mapping_by_id() (*compath.manager.Manager* method), 15
 dict_to_pandas_df() (in module *compath.utils*), 33
 drop_all() (*compath.manager.Manager* method), 13
- ## E
- EMAIL
 compath-manage-users-make-admin
 command line option, 11
 compath-manage-users-make-user
 command line option, 12
 enrich_graph_with_mappings() (in module *compath.visualization.cytoscape*), 27
 EQUIVALENT_TO (in module *compath.constants*), 24
- ## F
- filter_network_by_similarity() (in module *compath.visualization.cytoscape*), 27
 filter_results() (in module *compath.utils*), 36
 from_connection() (*compath.manager.Manager* static method), 13
- ## G
- get_all_accepted_mappings() (*compath.manager.Manager* method), 13
 get_all_mappings() (*compath.manager.Manager* method), 13
 get_all_mappings_from_pathway() (*compath.manager.Manager* method), 16
 get_all_pathways_from_db_with_mappings() (*compath.manager.Manager* method), 16
 get_ancestry_mappings_from_pathway_with_is_part_of_relationship() (*compath.manager.Manager* method), 16
 get_complement_mapping_info() (*compath.models.PathwayMapping* method), 20
 get_descendents_mappings_from_pathway_with_is_part_of_relationship() (*compath.manager.Manager* method), 15
 get_dendrogram_tree() (in module *compath.visualization.d3_dendrogram*), 28
 get_descendants() (in module *compath.visualization.d3_dendrogram*), 28
 get_enriched_pathways() (in module *compath.utils*), 34
 get_gene_pathways() (in module *compath.utils*), 34
 get_gene_sets_from_pathway_names() (in module *compath.utils*), 35
 get_genes_without_assigned_pathways() (in module *compath.utils*), 34
 get_last_action_in_module() (in module *compath.utils*), 35
 get_mapping() (*compath.manager.Manager* method), 14
 get_mapping_by_id() (*compath.manager.Manager* method), 14
 get_mapping_dendrogram() (in module *compath.visualization.d3_dendrogram*), 29
 get_mappings() (in module *compath.utils*), 34
 get_mappings_by_type() (*compath.manager.Manager* method), 13
 get_mappings_from_pathway_with_relationship() (*compath.manager.Manager* method), 15
 get_most_similar_names() (in module *compath.utils*), 36
 get_or_create_mapping() (*compath.manager.Manager* method), 14
 get_or_create_vote() (*compath.manager.Manager* method), 14
 get_pathway_info() (in module *compath.utils*), 35
 get_pathway_model_by_id() (in module *compath.utils*), 35
 get_pathway_model_by_name() (in module *compath.utils*), 34
 get_pathways_from_statement() (in module *compath.curation.utils*), 26
 get_top_matches() (in module *compath.utils*), 36
 get_user_by_email() (*compath.manager.Manager* method), 14
 get_user_vote() (*compath.models.PathwayMapping* method), 20
 get_users_from_emails() (in module *compath.curation.parser*), 25
 get_vote() (*compath.manager.Manager* method), 13
 get_vote_by_id() (*compath.manager.Manager* method), 13
 get_vote_by_id() (*compath.manager.Manager* method), 13
- ## H
- has_ancestry_pathway_tuple() (*compath.models.PathwayMapping* static method), 20
 has_database_pathway() (*compath.models.PathwayMapping* static method), 20
 has_descendant_pathway_tuple() (*compath.models.PathwayMapping* static method), 20

`has_pathway()` (*compath.models.PathwayMapping static method*), 20

`has_pathway_tuple()` (*compath.models.PathwayMapping static method*), 20

`HIERARCHY MANAGERS` (*in module compath.constants*), 23

I

`infer_hierarchy()` (*compath.manager.Manager method*), 16

`init_app()` (*compath.web.ComPathSQLAlchemy method*), 31

`is_acceptable()` (*compath.models.PathwayMapping property*), 20

`is_admin()` (*compath.models.User property*), 19

`is_valid_pathway_by_id()` (*in module compath.curation.utils*), 26

`is_valid_pathway_name()` (*in module compath.curation.utils*), 26

K

`KEGG` (*in module compath.constants*), 23

`KEGG_GENE_SET` (*in module compath.constants*), 23

`KEGG_URL` (*in module compath.constants*), 23

L

`label_tree()` (*in module compath.visualization.d3_dendrogram*), 28

`load_hierarchy()` (*in module compath.curation.hierarchies*), 25

`log` (*in module compath.utils*), 33

M

`Manager` (*class in compath.manager*), 13

`MAPPING_TYPES` (*in module compath.constants*), 23

`module`

- `compath`, 1
- `compath.constants`, 23
- `compath.curation.hierarchies`, 25
- `compath.curation.parser`, 25
- `compath.curation.utils`, 26
- `compath.manager`, 13
- `compath.models`, 19
- `compath.utils`, 33
- `compath.visualization.cytoscape`, 27
- `compath.visualization.d3_dendrogram`, 28
- `compath.visualization.venn_diagram`, 29
- `compath.web`, 31

`MSIG` (*in module compath.constants*), 23

`MSIG_GENE_SET` (*in module compath.constants*), 23

N

`networkx_to_cytoscape_js()` (*in module compath.visualization.cytoscape*), 27

P

`parse_curation_template()` (*in module compath.curation.parser*), 26

`parse_special_mappings()` (*in module compath.curation.parser*), 25

`PASSWORD`

- `compath-manage-users-make-user`
- command line option, 12

`PathwayMapping` (*class in compath.models*), 19

`pathways_to_similarity_network()` (*in module compath.visualization.cytoscape*), 27

`perform_hypergeometric_test()` (*in module compath.utils*), 35

`process_form_gene_set()` (*in module compath.utils*), 33

`process_overlap_for_venn_diagram()` (*in module compath.visualization.venn_diagram*), 29

R

`REACTOME` (*in module compath.constants*), 23

`REACTOME_GENE_SET` (*in module compath.constants*), 23

`REACTOME_URL` (*in module compath.constants*), 23

`remove_star_from_pathway_name()` (*in module compath.curation.utils*), 26

`Role` (*class in compath.models*), 19

S

`service_1_name` (*compath.models.PathwayMapping attribute*), 19

`service_1_pathway_id` (*compath.models.PathwayMapping attribute*), 19

`service_1_pathway_name` (*compath.models.PathwayMapping attribute*), 19

`service_2_name` (*compath.models.PathwayMapping attribute*), 19

`service_2_pathway_id` (*compath.models.PathwayMapping attribute*), 19

`service_2_pathway_name` (*compath.models.PathwayMapping attribute*), 19

`simulate_pathway_enrichment()` (*in module compath.utils*), 34

`STYLED_NAMES` (*in module compath.constants*), 23

T

`to_csv()` (in module *compath.utils*), [36](#)

`type` (*compath.models.PathwayMapping* attribute), [20](#)

`type` (*compath.models.Vote* attribute), [21](#)

U

`User` (class in *compath.models*), [19](#)

V

`Vote` (class in *compath.models*), [21](#)

`VOTE_ACCEPTANCE` (in module *compath.constants*), [23](#)

W

`WIKIPATHWAYS` (in module *compath.constants*), [23](#)

`WIKIPATHWAYS_GENE_SET` (in module *compath.constants*), [23](#)

`WIKIPATHWAYS_URL` (in module *compath.constants*), [23](#)