
Bio2BEL ExPASy Documentation

Release 0.2.0

Charles Tapley Hoyt

Nov 15, 2018

Contents:

1	Installation	3
1.1	Easiest	3
1.2	Get the Latest	3
1.3	For Developers	3
2	Testing	5
3	Manager	7
4	Models	11
5	Constants	13
6	Indices and tables	15
	Python Module Index	17

This library helps to download and parses the enzyme classes from the ExPASy ENZYME database.

1.1 Easiest

Download the latest stable code from [PyPI](#) with:

```
$ python3 -m pip install bio2bel_expasy
```

1.2 Get the Latest

Download the most recent code from [GitHub](#) with:

```
$ python3 -m pip install git+https://github.com/bio2bel/expasy.git
```

1.3 For Developers

Clone the repository from [GitHub](#) and install in editable mode with:

```
$ git clone https://github.com/bio2bel/expasy.git
$ cd expasy
$ python3 -m pip install -e .
```


CHAPTER 2

Testing

Bio2BEL ExPASy is tested with Python3 on Linux using [Travis CI](#).

Manager for Bio2BEL ExPASy.

```
class bio2bel_expasy.manager.Manager (*args, **kwargs)
    Creates a connection to database and a persistent session using SQLAlchemy.

    namespace_model
        alias of bio2bel_expasy.models.Enzyme

    id_enzyme = None
        Maps canonicalized ExPASy enzyme identifiers to their SQLAlchemy models

    is_populated () → bool
        Check if the database is already populated.

    count_enzymes () → int
        Count the number of enzyme entries in the database.

    count_enzyme_prosites () → int
        Count the number of enzyme-prosite annotations.

    count_prosites () → int
        Count the number of ProSite entries in the database.

    count_enzyme_proteins () → int
        Count the number of enzyme-protein annotations.

    count_proteins () → int
        Count the number of protein entries in the database.

    summarize () → Mapping[str, int]
        Return a summary dictionary over the content of the database.

    get_or_create_enzyme (expasy_id: str, description: Optional[str] = None) →
        bio2bel_expasy.models.Enzyme
        Get an enzyme from the database or creates it.

    get_or_create_prosite (prosite_id: str, **kwargs) → bio2bel_expasy.models.Prosite
        Get a prosite from the database or creates it.
```

get_or_create_protein (*accession_number: str, entry_name: str, **kwargs*) → `bio2bel_expasy.models.Protein`

Get a protein by its UniProt accession or create it.

Parameters

- **accession_number** –
- **entry_name** –
- **kwargs** –

populate (*tree_path: Optional[str] = None, database_path: Optional[str] = None*) → None

Populate the database..

Parameters

- **tree_path** –
- **database_path** –

populate_tree (*path: Optional[str] = None, force_download: bool = False*) → None

Download and populate the ExPASy tree.

Parameters

- **path** – A custom url to download
- **force_download** – If true, overwrites a previously cached file

populate_database (*path: Optional[str] = None, force_download: bool = False*) → None

Populate the ExPASy database.

Parameters

- **path** – A custom url to download
- **force_download** – If true, overwrites a previously cached file

get_enzyme_by_id (*expasy_id: str*) → `Optional[bio2bel_expasy.models.Enzyme]`

Get an enzyme by its ExPASy identifier.

Implementation note: canonicalizes identifier to remove all spaces first.

Parameters **expasy_id** – An ExPASy identifier. Example: 1.3.3.- or 1.3.3.19

get_parent_by_expasy_id (*expasy_id: str*) → `Optional[bio2bel_expasy.models.Enzyme]`

Return the parent ID of ExPASy identifier if exist otherwise returns None.

Parameters **expasy_id** – An ExPASy identifier

get_children_by_expasy_id (*expasy_id: str*) → `Optional[List[bio2bel_expasy.models.Enzyme]]`

Return a list of enzymes which are children of the enzyme with the given ExPASy enzyme identifier.

Parameters **expasy_id** – An ExPASy enzyme identifier

get_protein_by_uniprot_id (*uniprot_id: str*) → `Optional[bio2bel_expasy.models.Protein]`

Get a protein having the given UniProt identifier.

Parameters **uniprot_id** – A UniProt identifier

```
>>> from bio2bel_expasy import Manager
>>> manager = Manager()
>>> protein = manager.get_protein_by_uniprot_id('Q6AZW2')
>>> protein.accession_number
'Q6AZW2'
```

get_prosite_by_id (*prosite_id: str*) → Optional[bio2bel_expasy.models.Prosite]

Get a ProSite having the given ProSite identifier.

Parameters *prosite_id* – A ProSite identifier

get_prosites_by_expasy_id (*expasy_id: str*) → Optional[List[bio2bel_expasy.models.Prosite]]

Get a list of ProSites associated with the enzyme corresponding to the given identifier.

Parameters *expasy_id* – An ExPASy identifier

get_enzymes_by_prosite_id (*prosite_id: str*) → Optional[List[bio2bel_expasy.models.Enzyme]]

Return a list of enzymes associated with the given ProSite ID.

Parameters *prosite_id* – ProSite identifier

get_proteins_by_expasy_id (*expasy_id: str*) → Optional[List[bio2bel_expasy.models.Protein]]

Return a list of UniProt entries as tuples (*accession_number*, *entry_name*) of the given enzyme_id.

Parameters *expasy_id* – An ExPASy identifier

get_enzymes_by_uniprot_id (*uniprot_id: str*) → Optional[List[bio2bel_expasy.models.Enzyme]]

Return a list of enzymes annotated to the protein with the given UniProt accession number.

Parameters *uniprot_id* – A UniProt identifier

Example:

```
>>> from bio2bel_expasy import Manager
>>> manager = Manager()
>>> manager.get_enzymes_by_uniprot_id('Q6AZW2')
>>> ...
```

enrich_proteins_with_enzyme_families (*graph: pybel.struct.graph.BELGraph*) → None

Enrich proteins in the BEL graph with IS_A relations to their enzyme classes.

1. Gets a list of UniProt proteins
2. Annotates `pybel.constants.IS_A` relations for all enzyme classes it finds

look_up_enzyme (*node: pybel.dsl.node_classes.BaseEntity*) → Optional[bio2bel_expasy.models.Enzyme]

Try to get an enzyme model from the given node.

enrich_enzyme_with_proteins (*graph: pybel.struct.graph.BELGraph, node: pybel.dsl.node_classes.BaseEntity*) → None

Enrich an enzyme with all of its member proteins.

enrich_enzyme_parents (*graph: pybel.struct.graph.BELGraph, node: pybel.dsl.node_classes.BaseEntity*) → None

Enrich an enzyme with its parents.

enrich_enzyme_children (*graph: pybel.struct.graph.BELGraph, node: pybel.dsl.node_classes.BaseEntity*) → None

Enrich an enzyme with all of its children.

enrich_enzymes (*graph: pybel.struct.graph.BELGraph*) → None

Add all children of entries.

enrich_enzymes_with_prosites (*graph: pybel.struct.graph.BELGraph*) → None

Enrich enzyme classes in the graph with ProSites.

SQLAlchemy models for Bio2BEL ExPASy.

class `bio2bel_expasy.models.Enzyme` (***kwargs*)
ExPASy's main entry.

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.

expasy_id
The ExPASy enzyme code.

description
The ExPASy enzyme description. May need context of parents.

level
Return what level (1, 2, 3, or 4) this enzyme is based on the number of dashes in its id.

to_json() → Mapping
Return the data from this model as a dictionary.

as_bel() → `pybel.dsl.node_classes.Protein`
Return a PyBEL node representing this enzyme.

class `bio2bel_expasy.models.Prosite` (***kwargs*)
Maps ec to prosite entries.

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.

prosite_id
ProSite Identifier

as_bel () → pybel.dsl.node_classes.Protein

Return a PyBEL node data dictionary representing this ProSite entry.

class bio2bel_expasy.models.**Protein** (**kwargs)

Maps enzyme to SwissProt or UniProt.

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.

accession_number

UniProt *accession number*

entry_name

UniProt *entry name*.

as_bel () → pybel.dsl.node_classes.Protein

Return a PyBEL node data dictionary representing this UniProt entry.

CHAPTER 5

Constants

Constants for Bio2BEL ExPASy.

```
bio2bel_expasy.constants.EXPASY_TREE_URL = 'ftp://ftp.expasy.org/databases/enzyme/enzclass'
```

The web location of the enzyme class tree document

```
bio2bel_expasy.constants.EXPASY_TREE_DATA_PATH = '/home/docs/.bio2bel/expasy/enzclass.txt'
```

The local cache location where the enzyme class tree document is stored

```
bio2bel_expasy.constants.EXPASY_DATABASE_URL = 'ftp://ftp.expasy.org/databases/enzyme/enzyme.dat'
```

The web location of the ENZYME database document

```
bio2bel_expasy.constants.EXPASY_DATA_PATH = '/home/docs/.bio2bel/expasy/enzyme.dat'
```

The local cache location where the ENZYME database document is stored

CHAPTER 6

Indices and tables

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

b

`bio2bel_expasy`, [??](#)
`bio2bel_expasy.constants`, [13](#)
`bio2bel_expasy.manager`, [7](#)
`bio2bel_expasy.models`, [11](#)

A

accession_number (*bio2bel_expasy.models.Protein attribute*), 12
 as_bel() (*bio2bel_expasy.models.Enzyme method*), 11
 as_bel() (*bio2bel_expasy.models.Prosite method*), 11
 as_bel() (*bio2bel_expasy.models.Protein method*), 12

B

bio2bel_expasy (*module*), 1
 bio2bel_expasy.constants (*module*), 13
 bio2bel_expasy.manager (*module*), 7
 bio2bel_expasy.models (*module*), 11

C

count_enzyme_prosites() (*bio2bel_expasy.manager.Manager method*), 7
 count_enzyme_proteins() (*bio2bel_expasy.manager.Manager method*), 7
 count_enzymes() (*bio2bel_expasy.manager.Manager method*), 7
 count_prosites() (*bio2bel_expasy.manager.Manager method*), 7
 count_proteins() (*bio2bel_expasy.manager.Manager method*), 7

D

description (*bio2bel_expasy.models.Enzyme attribute*), 11

E

enrich_enzyme_children() (*bio2bel_expasy.manager.Manager method*), 9
 enrich_enzyme_parents() (*bio2bel_expasy.manager.Manager method*), 9
 enrich_enzyme_with_proteins() (*bio2bel_expasy.manager.Manager method*), 9
 enrich_enzymes() (*bio2bel_expasy.manager.Manager method*), 9

enrich_enzymes_with_prosites() (*bio2bel_expasy.manager.Manager method*), 9
 enrich_proteins_with_enzyme_families() (*bio2bel_expasy.manager.Manager method*), 9
 entry_name (*bio2bel_expasy.models.Protein attribute*), 12
 Enzyme (*class in bio2bel_expasy.models*), 11
 EXPASY_DATA_PATH (*in module bio2bel_expasy.constants*), 13
 EXPASY_DATABASE_URL (*in module bio2bel_expasy.constants*), 13
 expasy_id (*bio2bel_expasy.models.Enzyme attribute*), 11
 EXPASY_TREE_DATA_PATH (*in module bio2bel_expasy.constants*), 13
 EXPASY_TREE_URL (*in module bio2bel_expasy.constants*), 13

G

get_children_by_expasy_id() (*bio2bel_expasy.manager.Manager method*), 8
 get_enzyme_by_id() (*bio2bel_expasy.manager.Manager method*), 8
 get_enzymes_by_prosite_id() (*bio2bel_expasy.manager.Manager method*), 9
 get_enzymes_by_uniprot_id() (*bio2bel_expasy.manager.Manager method*), 9
 get_or_create_enzyme() (*bio2bel_expasy.manager.Manager method*), 7
 get_or_create_prosite() (*bio2bel_expasy.manager.Manager method*), 7
 get_or_create_protein() (*bio2bel_expasy.manager.Manager method*), 7
 get_parent_by_expasy_id() (*bio2bel_expasy.manager.Manager method*), 8
 get_prosite_by_id() (*bio2bel_expasy.manager.Manager method*), 8
 get_prosites_by_expasy_id() (*bio2bel_expasy.manager.Manager method*), 9

`get_protein_by_uniprot_id()`
 (*bio2bel_expasy.manager.Manager* method), 8
`get_proteins_by_expasy_id()`
 (*bio2bel_expasy.manager.Manager* method), 9

I

`id_enzyme` (*bio2bel_expasy.manager.Manager* attribute), 7
`is_populated()` (*bio2bel_expasy.manager.Manager* method), 7

L

`level` (*bio2bel_expasy.models.Enzyme* attribute), 11
`look_up_enzyme()` (*bio2bel_expasy.manager.Manager* method), 9

M

`Manager` (class in *bio2bel_expasy.manager*), 7

N

`namespace_model` (*bio2bel_expasy.manager.Manager* attribute), 7

P

`populate()` (*bio2bel_expasy.manager.Manager* method), 8
`populate_database()`
 (*bio2bel_expasy.manager.Manager* method), 8
`populate_tree()` (*bio2bel_expasy.manager.Manager* method), 8
`Prosites` (class in *bio2bel_expasy.models*), 11
`prosite_id` (*bio2bel_expasy.models.Prosites* attribute), 11
`Protein` (class in *bio2bel_expasy.models*), 12

S

`summarize()` (*bio2bel_expasy.manager.Manager* method), 7

T

`to_json()` (*bio2bel_expasy.models.Enzyme* method),
11