

---

# **Bio2BEL Entrez Documentation**

***Release 0.3.0-dev***

**Charles Tapley Hoyt**

**Mar 11, 2020**



**CONTENTS:**

|          |                            |           |
|----------|----------------------------|-----------|
| <b>1</b> | <b>Manager</b>             | <b>3</b>  |
| <b>2</b> | <b>Models</b>              | <b>7</b>  |
| <b>3</b> | <b>Constants</b>           | <b>9</b>  |
| <b>4</b> | <b>Indices and tables</b>  | <b>11</b> |
|          | <b>Python Module Index</b> | <b>13</b> |
|          | <b>Index</b>               | <b>15</b> |



A Bio2BEL package for Entrez Gene and HomoloGene.



## MANAGER

Manager for Bio2BEL Entrez.

```
class bio2bel_entrez.manager.Manager (*args, **kwargs)
    Genes and orthologies.

    namespace_model
        alias of bio2bel_entrez.models.Gene

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

    get_or_create_species (taxonomy_id: str, **kwargs) → bio2bel_entrez.models.Species
        Get or create a Species model.

        Parameters taxonomy_id – NCBI taxonomy identifier

    get_gene_by_entrez_id (entrez_id: str) → Optional[bio2bel_entrez.models.Gene]
        Get a gene with the given Entrez Gene identifier, if it exists.

        Parameters entrez_id – Entrez Gene identifier

    get_genes_by_name (name: str) → List[bio2bel_entrez.models.Gene]
        Get a list of genes with the given name (case insensitive).

        Parameters name – A gene name

    get_gene_by_rgd_name (name: str) → Optional[bio2bel_entrez.models.Gene]
        Get a gene by its RGD name.

        Parameters name – RGD gene symbol

    get_gene_by_mgi_name (name: str) → Optional[bio2bel_entrez.models.Gene]
        Get a gene by its MGI name.

        Parameters name – MGI gene symbol

    get_gene_by_hgnc_name (name: str) → Optional[bio2bel_entrez.models.Gene]
        Get a gene by its HGNC gene symbol.

    get_or_create_gene (entrez_id: str, **kwargs) → bio2bel_entrez.models.Gene
        Get or create a Gene model.

        Parameters entrez_id – Entrez Gene identifier

    get_or_create_homologene (homologene_id: str, **kwargs) →
        bio2bel_entrez.models.Homologene
        Get or create a HomoloGene model.

        Parameters homologene_id – HomoloGene Gene identifier
```

**populate\_homologene** (*url=None, cache=True, force\_download=False, tax\_id\_filter=None*) → None  
 Populate the database.

**Parameters**

- **url** (*Optional[str]*) – Homologene data url
- **cache** (*bool*) – If true, the data is downloaded to the file system, else it is loaded from the internet
- **force\_download** (*bool*) – If true, overwrites a previously cached file
- **tax\_id\_filter** (*Optional[iter[str]]*) – Species to keep

**populate\_gene\_info** (*url: Optional[str] = None, cache: bool = True, force\_download: bool = False, interval: Optional[int] = None, tax\_id\_filter: Iterable[str] = None*)  
 Populate the database.

**Parameters**

- **url** – A custom url to download
- **interval** – The number of records to commit at a time
- **cache** – If true, the data is downloaded to the file system, else it is loaded from the internet
- **force\_download** – If true, overwrites a previously cached file
- **tax\_id\_filter** – Species to keep

**populate** (*gene\_info\_url: Optional[str] = None, interval: Optional[int] = None, tax\_id\_filter: Iterable[str] = ('9606', '10090', '10116', '7227', '4932', '6239', '7955', '9913', '9615'), homologene\_url: Optional[str] = None*)  
 Populate the database.

**Parameters**

- **gene\_info\_url** – A custom url to download
- **interval** – The number of records to commit at a time
- **tax\_id\_filter** – Species to keep. Defaults to 9606 (human), 10090 (mouse), 10116 (rat), 7227 (fly), and 4932 (yeast). Explicitly set to None to get all taxonomies.
- **homologene\_url** – A custom url to download

**lookup\_node** (*node: pybel.dsl.node\_classes.BaseEntity*) → *Optional[bio2bel\_entrez.models.Gene]*  
 Look up a gene from a PyBEL data dictionary.

**iter\_genes** (*graph: pybel.struct.graph.BELGraph, use\_tqdm: bool = False*) → *Iterable[Tuple[pybel.dsl.node\_classes.BaseEntity, bio2bel\_entrez.models.Gene]]*  
 Iterate over genes in the graph that can be mapped to an Entrez gene.

**normalize\_genes** (*graph: pybel.struct.graph.BELGraph, use\_tqdm: bool = False*) → None  
 Add identifiers to all Entrez genes.

**enrich\_genes\_with\_homologenes** (*graph: pybel.struct.graph.BELGraph*) → None  
 Enrich the nodes in a graph with their HomoloGene parents.

**enrich\_equivalences** (*graph: pybel.struct.graph.BELGraph*) → None  
 Add equivalent node information.

**enrich\_orthologies** (*graph: pybel.struct.graph.BELGraph*) → None  
 Add ortholog relationships to graph.

**add\_homologene\_namespace\_to\_graph** (*graph*: *pybel.struct.graph.BELGraph*) → *pybel.manager.models.Namespace*  
 Add the homologene namespace to the graph.

**count\_genes** () → int  
 Count the genes in the database.

**count\_homologenes** () → int  
 Count the HomoloGenes in the database.

**count\_species** () → int  
 Count the species in the database.

**list\_species** () → List[bio2bel\_entrez.models.Species]  
 List all species in the database.

**list\_homologenes** () → List[bio2bel\_entrez.models.Homologene]  
 List all HomoloGenes in the database.

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

**list\_genes** (*limit*: *Optional[int]* = *None*, *offset*: *Optional[int]* = *None*) → List[bio2bel\_entrez.models.Gene]  
 List genes in the database.



## MODELS

SQLAlchemy models for Bio2BEL Entrez.

```
class bio2bel_entrez.models.Species (**kwargs)
```

Represents a Species.

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.

**taxonomy\_id**

NCBI Taxonomy Identifier

```
class bio2bel_entrez.models.Homologene (**kwargs)
```

Represents a HomoloGene Group.

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.

**as\_bel** (*func: Optional[str] = None*) → `pybel.dsl.node_classes.CentralDogma`

Make a PyBEL DSL object from this HomoloGene.

```
class bio2bel_entrez.models.Gene (**kwargs)
```

Represents a gene.

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.

**entrez\_id**

NCBI Entrez Gene Identifier

**name**

Entrez Gene Symbol

**description**

Gene Description

**type\_of\_gene**

Type of Gene

**property bel\_encoding**

Return the BEL encoding.

**as\_bel** (*func=None*) → pybel.dsl.node\_classes.CentralDogma

Make a PyBEL DSL object from this gene.

**property is\_transcribed**

Return if this gene can be transcribed to an RNA.

**property is\_translated**

Return if this gene can be translated to a protein.

**to\_json** () → Mapping[str, int]

Return this Gene as a JSON dictionary.

**class** bio2bel\_entrez.models.Xref (\*\**kwargs*)

Represents a database cross reference.

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.

**database**

Database name

**value**

Database entry name

## CONSTANTS

Constants for Bio2BEL Entrez.

```
bio2bel_entrez.constants.GENE_INFO_COLUMNS = ['#tax_id', 'GeneID', 'Symbol', 'dbXrefs', 'd  
Columns fro gene_info.gz that are used
```

```
bio2bel_entrez.constants.CONSORTIUM_SPECIES_MAPPING = {'FLYBASE': '7227', 'HGNC': '9606',  
All namespace codes (in lowercase) that can map to ncbigene
```



## INDICES AND TABLES

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



## PYTHON MODULE INDEX

### b

`bio2bel_entrez`, ??  
`bio2bel_entrez.constants`, 9  
`bio2bel_entrez.manager`, 3  
`bio2bel_entrez.models`, 7



## A

`add_homologene_namespace_to_graph()`  
(*bio2bel\_entrez.manager.Manager* method), 4  
`as_bel()` (*bio2bel\_entrez.models.Gene* method), 8  
`as_bel()` (*bio2bel\_entrez.models.Homologene*  
method), 7

## B

`bel_encoding()` (*bio2bel\_entrez.models.Gene* prop-  
erty), 7  
`bio2bel_entrez` (module), 1  
`bio2bel_entrez.constants` (module), 9  
`bio2bel_entrez.manager` (module), 3  
`bio2bel_entrez.models` (module), 7

## C

`CONSORTIUM_SPECIES_MAPPING` (in module  
*bio2bel\_entrez.constants*), 9  
`count_genes()` (*bio2bel\_entrez.manager.Manager*  
method), 5  
`count_homologenes()`  
(*bio2bel\_entrez.manager.Manager* method), 5  
`count_species()` (*bio2bel\_entrez.manager.Manager*  
method), 5

## D

`database` (*bio2bel\_entrez.models.Xref* attribute), 8  
`description` (*bio2bel\_entrez.models.Gene* attribute),  
7

## E

`enrich_equivalences()`  
(*bio2bel\_entrez.manager.Manager* method), 4  
`enrich_genes_with_homologenes()`  
(*bio2bel\_entrez.manager.Manager* method), 4  
`enrich_orthologies()`  
(*bio2bel\_entrez.manager.Manager* method), 4  
`entrez_id` (*bio2bel\_entrez.models.Gene* attribute), 7

## G

`Gene` (class in *bio2bel\_entrez.models*), 7

`GENE_INFO_COLUMNS` (in module  
*bio2bel\_entrez.constants*), 9  
`get_gene_by_entrez_id()`  
(*bio2bel\_entrez.manager.Manager* method), 3  
`get_gene_by_hgnc_name()`  
(*bio2bel\_entrez.manager.Manager* method), 3  
`get_gene_by_mgi_name()`  
(*bio2bel\_entrez.manager.Manager* method), 3  
`get_gene_by_rgd_name()`  
(*bio2bel\_entrez.manager.Manager* method), 3  
`get_genes_by_name()`  
(*bio2bel\_entrez.manager.Manager* method), 3  
`get_or_create_gene()`  
(*bio2bel\_entrez.manager.Manager* method), 3  
`get_or_create_homologene()`  
(*bio2bel\_entrez.manager.Manager* method), 3  
`get_or_create_species()`  
(*bio2bel\_entrez.manager.Manager* method), 3

## H

`Homologene` (class in *bio2bel\_entrez.models*), 7

## I

`is_populated()` (*bio2bel\_entrez.manager.Manager*  
method), 3  
`is_transcribed()` (*bio2bel\_entrez.models.Gene*  
property), 8  
`is_translated()` (*bio2bel\_entrez.models.Gene*  
property), 8  
`iter_genes()` (*bio2bel\_entrez.manager.Manager*  
method), 4

## L

`list_genes()` (*bio2bel\_entrez.manager.Manager*  
method), 5  
`list_homologenes()`  
(*bio2bel\_entrez.manager.Manager* method), 5  
`list_species()` (*bio2bel\_entrez.manager.Manager*  
method), 5  
`lookup_node()` (*bio2bel\_entrez.manager.Manager*  
method), 4

## M

Manager (*class in bio2bel\_entrez.manager*), 3

## N

name (*bio2bel\_entrez.models.Gene attribute*), 7

namespace\_model (*bio2bel\_entrez.manager.Manager attribute*), 3

normalize\_genes ()  
(*bio2bel\_entrez.manager.Manager method*), 4

## P

populate ()  
(*bio2bel\_entrez.manager.Manager method*), 4

populate\_gene\_info ()  
(*bio2bel\_entrez.manager.Manager method*), 4

populate\_homologene ()  
(*bio2bel\_entrez.manager.Manager method*), 3

## S

Species (*class in bio2bel\_entrez.models*), 7

summarize ()  
(*bio2bel\_entrez.manager.Manager method*), 5

## T

taxonomy\_id (*bio2bel\_entrez.models.Species attribute*), 7

to\_json () (*bio2bel\_entrez.models.Gene method*), 8

type\_of\_gene (*bio2bel\_entrez.models.Gene attribute*), 7

## V

value (*bio2bel\_entrez.models.Xref attribute*), 8

## X

Xref (*class in bio2bel\_entrez.models*), 8