
Bio2BEL InterPro Documentation

Release 0.2.1

Charles Tapley Hoyt

Apr 15, 2019

CONTENTS:

1	Manager	1
2	Models	3
3	Indices and tables	5
	Python Module Index	7

MANAGER

Manager for Bio2BEL InterPro.

```
class bio2bel_interpro.manager.Manager (*args, **kwargs)
    Protein-family and protein-domain memberships.

    pathway_model
        alias of bio2bel_interpro.models.Entry

    pathway_model_identifier_column
        The InterPro identifier

    protein_model
        alias of bio2bel_interpro.models.Protein

    namespace_model
        alias of bio2bel_interpro.models.Entry

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

    count_interpros () → int
        Count the number of InterPro entries in the database.

    list_interpros () → List[bio2bel_interpro.models.Entry]
        List the InterPro entries in the database.

    count_annotations () → int
        Count the number of protein-interpro associations.

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

    list_proteins () → List[bio2bel_interpro.models.Protein]
        List the proteins in the database.

    count_go_terms () → int
        Count the GO terms in the database.

    summarize () → Mapping[str, int]
        Summarize the database.

    get_type_by_name (name: str) → Optional[bio2bel_interpro.models.Type]
        Get an InterPro entry type by its name if it exists.

    get_interpro_by_interpro_id (interpro_id: str) → Optional[bio2bel_interpro.models.Entry]
        Get a InterPro entry by its identifier if it exists.

    get_go_by_go_identifier (go_id: str) → Optional[bio2bel_interpro.models.GoTerm]
        Get a GO term by its identifier if it exists.
```

get_or_create_interpro (*interpro_id: str, **kwargs*) → *bio2bel_interpro.models.Entry*
 Get an InterPro entry by its identifier if it exists, or create one.

get_or_create_go_term (*go_id: str, name=None*) → *bio2bel_interpro.models.GoTerm*
 Get a GO term by its identifier if it exists, or create one.

populate (*entries_url: Optional[str] = None, tree_url: Optional[str] = None, go_mapping_path: Optional[str] = None, populate_proteins: bool = False, proteins_url: Optional[str] = None*) → *None*
 Populate the database.

Parameters

- **entries_url** (*Optional[str]*) –
- **tree_url** (*Optional[str]*) –
- **go_mapping_path** (*Optional[str]*) –
- **proteins_url** (*Optional[str]*) –

get_interpro_by_name (*name: str*) → *Optional[bio2bel_interpro.models.Entry]*
 Get an InterPro family by name, if exists.

enrich_proteins (*graph: pybel.struct.graph.BELGraph*)
 Find UniProt entries and annotates their InterPro entries.

enrich_interpros (*graph: pybel.struct.graph.BELGraph*)
 Find InterPro entries and annotates their proteins.

to_bel () → *pybel.struct.graph.BELGraph*
 Get the InterPro hierarchy and annotations as BEL.

MODELS

SQLAlchemy database models for Bio2BEL InterPro.

class bio2bel_interpro.models.**Type** (**kwargs)
InterPro Entry Type.

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.

name
The InterPro entry type

class bio2bel_interpro.models.**Protein** (**kwargs)
Represents proteins that are annotated to InterPro families.

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.

uniprot_id
UniProt identifier

as_bel () → pybel.dsl.node_classes.Protein
Return this protein as a PyBEL node.

class bio2bel_interpro.models.**GoTerm** (**kwargs)
Represents a GO term.

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.

go_id
Gene Ontology identifier

name
Label

class bio2bel_interpro.models.**Entry** (**kwargs)
Represents families, domains, etc. in InterPro.

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.

interpro_id

The InterPro identifier

name

The InterPro entry name

as_bel () → `pybel.dsl.node_classes.Protein`

Return this InterPro entry as a PyBEL node.

class `bio2bel_interpro.models.Annotation` (***kwargs*)

Mapping of InterPro to protein.

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.

start

Starting position on reference sequence of annotation

end

Ending position on reference sequence of annotation

INDICES AND TABLES

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

PYTHON MODULE INDEX

b

`bio2bel_interpro.manager`, 1
`bio2bel_interpro.models`, 3

A

Annotation (class in *bio2bel_interpro.models*), 4
 as_bel() (*bio2bel_interpro.models.Entry* method), 4
 as_bel() (*bio2bel_interpro.models.Protein* method), 3

B

bio2bel_interpro.manager (module), 1
 bio2bel_interpro.models (module), 3

C

count_annotations() (*bio2bel_interpro.manager.Manager* method), 1
 count_go_terms() (*bio2bel_interpro.manager.Manager* method), 1
 count_interpros() (*bio2bel_interpro.manager.Manager* method), 1
 count_proteins() (*bio2bel_interpro.manager.Manager* method), 1

E

end (*bio2bel_interpro.models.Annotation* attribute), 4
 enrich_interpros() (*bio2bel_interpro.manager.Manager* method), 2
 enrich_proteins() (*bio2bel_interpro.manager.Manager* method), 2
 Entry (class in *bio2bel_interpro.models*), 3

G

get_go_by_go_identifier() (*bio2bel_interpro.manager.Manager* method), 1
 get_interpro_by_interpro_id() (*bio2bel_interpro.manager.Manager* method), 1
 get_interpro_by_name() (*bio2bel_interpro.manager.Manager* method), 2

get_or_create_go_term() (*bio2bel_interpro.manager.Manager* method), 2
 get_or_create_interpro() (*bio2bel_interpro.manager.Manager* method), 1
 get_type_by_name() (*bio2bel_interpro.manager.Manager* method), 1
 go_id (*bio2bel_interpro.models.GoTerm* attribute), 3
 GoTerm (class in *bio2bel_interpro.models*), 3

interpro_id (*bio2bel_interpro.models.Entry* attribute), 4
 is_populated() (*bio2bel_interpro.manager.Manager* method), 1

L

list_interpros() (*bio2bel_interpro.manager.Manager* method), 1
 list_proteins() (*bio2bel_interpro.manager.Manager* method), 1

M

Manager (class in *bio2bel_interpro.manager*), 1

N

name (*bio2bel_interpro.models.Entry* attribute), 4
 name (*bio2bel_interpro.models.GoTerm* attribute), 3
 name (*bio2bel_interpro.models.Type* attribute), 3
 namespace_model (*bio2bel_interpro.manager.Manager* attribute), 1

P

pathway_model (*bio2bel_interpro.manager.Manager* attribute), 1
 pathway_model_identifier_column (*bio2bel_interpro.manager.Manager* attribute), 1
 populate() (*bio2bel_interpro.manager.Manager* method), 2

`Protein` (*class in bio2bel_interpro.models*), [3](#)
`protein_model` (*bio2bel_interpro.manager.Manager*
attribute), [1](#)

S

`start` (*bio2bel_interpro.models.Annotation attribute*), [4](#)
`summarize()` (*bio2bel_interpro.manager.Manager*
method), [1](#)

T

`to_bel()` (*bio2bel_interpro.manager.Manager*
method), [2](#)
`Type` (*class in bio2bel_interpro.models*), [3](#)

U

`uniprot_id` (*bio2bel_interpro.models.Protein at-*
tribute), [3](#)