pythoncyc Documentation
Release

Mario Latendresse

July 28, 2016

Contents:

pythoncyc package

1.1 Submodules

1.2 pythoncyc.PGDB module

class `pythoncyc.PGDB.`**`PGDB`**(*orgid*)

    Please consult the the tutorial.html file under the doc directory for an introduction on how to use this class.

    `DNA_binding_sites_of_protein`(*tf*, *all_forms=None*)

        Description Given a transcription factor, return all of its DNA binding sites.

        Parms

            tf An instance of the class Proteins, a frame id or PFrame.

            all_forms Keyword, When True, then return the DNA binding sites of modified forms and subunits of tf as well.

        Return value A list of instances of the class DNA-Binding-Sites.

    `activation_p`(*reg_frame*)

        Description A predicate that determines if a given regulation frame is describing activation.

        Parms

            reg_frame An instance of class Regulation, a frame id or PFrame

        Return value A boolean value.

    `adjacent_genes_p`(*g1*, *g2*)

        Description Given two genes, this predicate will return True if they are on the same replicon, and adjacent to one another.

        Parms

            g1 An instance of class Genes, a frame id or PFrame.

            g2 An instance of class Genes, a frame id or PFrame.

        Return value A boolean value.

    `all_cofactors`()

        Description Return a list of all cofactors used in the current PGDB.

        Parms None.

Return value A list of cofactor frame ids.

all_direct_forms_of_protein(*protein*)

  Description Given a protein, this function will return all of the directly related proteins of its modified and unmodified forms, meaning all of their direct subunits and all of their direct containers.

  Parms

   protein An instance of the class Proteins, a frame id or PFrame.

  Return Value A list of instances of the class Proteins.

all_enzymes(*type=None*)

  Description Return all enzymes of a given type.

  Parms

   type Keyword, A type as taken from the parameter to fn enzyme. Defaults to 'chemical-change'.

  Return value A list of instances of class Proteins.

all_forms_of_protein(*protein*)

  Description Given a protein, this function will return all of the related proteins of its modified and unmodified forms, meaning all of their subunits and all of their containers. Unlike all_direct_forms_of_protein, this function is not limited to the direct containers only.

  Parms

   protein An instance of the class Proteins, a frame id or PFrame.

  Return value A list of instances of the class Proteins.

all_genetic_regulation_proteins(*allow_modified_forms=True*, *class_name=None*)

  Description Enumerates all proteins that are involved in genetic regulation of a particular given class. Optionally, just unmodified forms of the proteins are returned.

  Parms

   class_name Keyword, The class Regulation or a subclass. It defaults to Regulation-of-Transcription-Initiation.

   allow_modified_forms Keyword, A boolean value. If True, modified and unmodified forms of the protein are returned. If false, then only unmodified forms of the proteins are returned. The default value is True.

  Return value A list of protein frames that are involved in the specified form of regulation.

all_modulators()

  Description Enumerate all of the modulators, or direct regulators, in the current PGDB.

  Parms None.

  Return value A list of regulator frame ids.

all_operons()

  Description Enumerates all operons. In this case, an operon is defined as a list of overlapping instances of Transcription-Units.

  Parms None.

> **Return value** A list of lists of Transcription-Units, where all Transcription-Units in the list belong to the
> same operon.

all_pathways (*selector='all'*, *base=False*)

> **Description** Returns a list of pathway instance frames of a specified type.
>
> **Parms**
>
>> **selector** Selects whether all pathways, or just small-molecule metabolism base pathways. Can take
>> either 'all' or 'small-molecule'. Defaults to 'all'.
>>
>> **base** If this boolean parameter is True, only includes base pathways. Otherwise, all pathways, in-
>> cluding superpathways, will be returned.
>
> **Return value** A list of instances of class Pathways.

all_products_of_gene (*gene*)

> **Description** Collects all proteins (not necessarily enzymes) that are encoded by the given gene.
>
> **Parms**
>
>> **gene** An instance of class Genes, a frame id or PFrame.
>
> **Return value** A list of instances of class Proteins.

all_protein_complexes (*filter='all'*)

> **Description** Enumerates different types of protein complexes.
>
> **Parms**
>
>> **filter** Keyword, The type of protein complexes to return. The argument must be one of the following
>> values:
>>
>>> **'all'** Return all protein complexes.
>>>
>>> **'hetero'** Return all heteromultimers.
>>>
>>> **'homo'** Return all homomultimers.
>
> **Return value** A list of protein complex frame ids.

all_reactions (*type='metab-smm'*)

all_rxns (*type='metab-smm'*)

> **Description** Returns a set of reactions that fall within a particular category.
>
> **Parms**
>
>> **type** The type of reaction to return. Defaults to 'metab-smm'. The possible values are:
>>
>>> **'all'** All reactions.
>>>
>>> **'metab-pathways'** All reactions found within metabolic pathways. Includes reactions that are
>>> pathway holes. May include a handfull of reactions whose substrates are macromolecules, e.g.,
>>> ACP. Excludes transport reactions.
>>>
>>> **'metab-smm'** All reactions of small molecule metabolism, whether or not they are present in a
>>> pathway. Subsumes metab-pathways.
>>>
>>> **'metab-all'** All enzyme-catalyzed reactions. Subsumes metab-smm.
>>>
>>> **'enzyme'** All enzyme-catalyzed reactions (i.e., instances of either EC-Reactions class or
>>> Unclassified-Reactions class).
>>>
>>> **'transport'** All transport reactions.

'small-molecule' All reactions whose substrates are all small molecules, as opposed to macromolecules. Excludes transport reactions.

'protein-small-molecule-reaction' One of the substrates of the reaction is a macromolecule, and one of the substrates of the reaction is a small molecule.

'protein-reaction' All substrates of the reaction are proteins.

'trna-reaction' One of the substrates of the reaction is a tRNA.

'spontaneous' Spontaneous reactions.

'non-spontaneous' Non-spontaneous reactions that are likely to be enzyme catalyzed. Some reactions will be returned for type non-spontaneous that will not be returned by enzyme.

Return value A list of reaction frame ids.

all_sigma_factors()

Description Enumerate all RNA polymerase sigma factors.

Parms None.

Return value A list of all instances of the class Sigma-Factors.

all_substrates(*rxns*)

Description Returns all unique substrates used in the reactions specified by the parameter rxns.

Parms

    rxns A list of reaction PFrames or frame ids.

Return value A list of compound frame ids. There might be strings in the list, as the left and right slots of a reaction frame can contain strings.

all_transcription_factors(*allow_modified_forms=True*)

Description Enumerates all transcription factors, or just unmodified forms of transcription factors.

Parms

    allow_modified_forms Keyword, A boolean value. If True, modified and unmodified forms of the protein are returned. If false, then only unmodified forms of the proteins are returned. The default value is t.

Return value A list of protein frame ids that are transcription factors.

all_transported_chemicals(*from_compartment=None,   to_compartment=None,   primary_only=False*)

Description Enumerates all chemicals transported by transport reactions in the current PGDB.

Parms

    from_compartment Keyword, The compartment that the chemical is coming from (see Cellular Component Ontology).

    to_compartment Keyword, The compartment that the chemical is going to (see Cellular Component Ontology).

    primary_only Keyword, If True, filter out common transport compounds, such as protons and Na+.

Return value A list of compound frame ids.

all_transporters()

Description Enumerate all transport proteins.

Parms None.

Return value A list of instances of class Proteins.

all_transporters_across(*membranes=None*, *method=None*)

Description Returns a list of transport proteins that transport across one of the given membranes.

Parms

membranes Keyword, Either all or a list of instances of the class. Defaults to all CCO-MEMBRANE.

method Keyword, Either 'location' or 'reaction-compartments'. 'location' will check the locations slot, while 'reaction-compartments' will examine the compartments of reaction substrates. Default value is 'location'.

Return value A list of instances of class Proteins.

autocatalytic_reactions_of_enzyme(*protein*)

Description Returns a list of reaction frames, where the protein participates as a substrate of the reaction, and the reaction has no associated Enzymatic Reaction frame. This implies that the protein substrate of the reaction might autocatalyzing the reaction.

Parms

protein An instance frame of class Proteins, a frame id or PFrame.

Return value A list of instances of class Reactions.

base_components_of_protein(*p*, *exclude_small_molecules=None*)

Description Same as function monomers-of-protein, but also returns components of the protein that are RNAs or compounds, not just polypeptides.

Parms

p An instance of the class Proteins, a frame id or PFrame.

exclude_small_molecules Keyword, If nil, then small molecule components are also returned. Default value is True.

Return value Two values as a list. The first value is a list of the components, which can be instances of the following classes: Polypeptides, RNA, and Compounds. The second value is a list of the corresponding coefficients of the components in the first value.

binding_site_promoters(*tu*)

Description Returns the promoters of the given DNA binding site.

Parms

tu An instance of the class DNA-Binding-Sites, a frame id or PFrame.

Return value A list of instances of class Promoters.

binding_site_to_regulators(*bsite*)

Description Returns all of the transcription factors of the given binding site.

Parms

bsite An instance of class DNA-Binding-Sites, a frame id or PFrame.

Return value A list of instances of class Proteins.

binding_site_transcription_units(*promoter*)

Description Returns all transcription units of a given binding site.

Parms

promoter, a frame id or PFrame. An instance of class DNA-Binding-Sites or mRNA-Binding-Sites.

Return value A list of instances of class Transcription-Units.

binding_sites_affecting_gene(*gene*)

Description Returns all binding sites which are present in the same transcription units as the given gene.

Parms

gene An instance of the class Genes, a frame id or PFrame.

Return value A list of instances of class DNA-Binding-Sites.

chromosome_of_gene(*gene*)

Description Returns the replicon on which the gene is located. If the gene is located on a contig that is, in turn, part of a chromosome, then the contig is returned.

Parms

gene An instance of class Genes, a frame id or PFrame.

Return value An instance of class Genetic-Elements.

chromosome_of_object(*item*)

Description Given a frame object, the replicon where it is located is returned. If there is no associated replicon for the object, nil is returned. If the object is on more than one replicon, an error is thrown.

Parms

item, a frame id or PFrame An instance of class All-Genes, Transcription-Units, Promoters, Terminators, Misc-Features, or DNA-Binding-Sites.

Return value An instance of class Genetic-Elements.

chromosome_of_operon(*tu*)

Description Returns the replicon of the given transcription unit.

Parms

tu An instance of the class Transcription-Units, a frame id or PFrame.

Return value An instance of class Genetic-Elements.

cofactors_and_pgroups_of_enzrxn(*enzrxn*)

Description Returns the cofactors and prosthetic groups of an enzymatic reaction.

Parms

enzrxn An instance of the class Enzymatic-Reactions, a frame id or PFrame.

Return value A list of children of class Chemicals or strings, representing cofactors and/or prosthetic groups.

compartment_of_rxn(*rxn*, *default=None*)

Description Returns the compartment of the reaction for non-transport reactions.

Parms

rxn An instance of the class Reactions, that is, a frame id or PFrame.

default Keyword, The default compartment for reactions without any compartment annotations on their substrates. The default value is CCO-CYTOSOL.

Return value A child of the class CCO.

compartments_of_reaction(*rxn*, *sides=None*, *default_compartment=None*)

Description Returns the compartments associated with the given reaction.

Parms

rxn An instance of the class Reactions, a frame id or PFrame.

sides Keyword, The slots of the reaction to consider. The default value is (LEFT RIGHT).

default_compartment Keyword, The default compartment, as determined by the function (default-compartment), which currently is set to CCO-CYTOSOL.

Return value A list of children of the class CCO.

complex_p(*frame*)

Description A predicate that determines whether the given frame is a protein complex.

Parms

frame a frame id or PFrame.

Return value A boolean value.

compounds_of_pathway(*pwy*)

Description Return all substrates of all reactions that are within the given pathway.

Parms

pwy An instance of the class Pathways, a frame id or PFrame.

Return value A list of children of class Compounds, children of class Polymer-Segments, or strings.

containers_of(*protein*, *exclude_self=None*)

Description Return all complexes of which the given protein is a direct or indirect component.

Parms

protein An instance of the class Proteins, a frame id or PFrame.

exclude_self Keyword, If True, then protein will not be included in the return value.

Return value A list of instances of the class Proteins.

containing_chromosome(*site*)

Description Given a site (whether a DNA binding site, a promoter, a gene, or a terminator) along a transcription unit, return the correspodning regulon.

Parms

> **site, a frame id or PFrame.** An instance of class Transcription-Units, mRNA-Binding-Sites, DNA-Binding-Sites, Promoters, Genes, or Terminators.

Return value An instance of class Genetic-Elements.

containing_tus(*site*)

> **Description** Given a site (whether a DNA binding site, a promoter, a gene, or a terminator) along a transcription unit, return the correspodning transcription units.

> **Parms**

> > **site, a frame id or PFrame.** An instance of class Transcription-Units, mRNA-Binding-Sites, DNA-Binding-Sites, Promoters, Genes, or Terminators.

> **Return value** A list of instances of class Transcription-Units.

cotranscribed_genes(*gene*)

> **Description** Return all co-transcribed genes (i.e., genes which are a part of one or more of the same transcription units) of the given gene.

> **Parms**

> > **gene** An instance of class Genes, a frame id or PFrame

> **Return value** A list of instances of class Genes.

create_frame_objects(*frameids*)
Create all the required PFrames for the given frameids. If a PFrame already exist for a frameid on the PGDB, reuse that PFrame, otherwise create a PFrame. No data is transferred from Pathway Tools.

> **Parm** frameids, list of frame ids (strings)

> **Side-Effects** self is modified to contain new PFrames indexed on new frameids

> **Return** list of PFrames

deactivated_or_inhibited_by_compound(*cpds*, *mode=None*, *mechanisms=None*, *phys_relevant=None*, *slots=None*)

> **Description** Returns all pathways in which the given compound appears as a substrate.

> **Parms**

> > **cpds** An instance or list of instances of class Compounds, a frame id or PFrame.

> > **mode** Keyword, Represents the type of regulation. Can take on the values of "+", "-", or None.

> > **mechanisms** Keyword, Keywords from the mechanism slot of the corresponding sub-class of the class Regulation. If True, only regulation objects with mechanisms in this list will be explored for regulated objects.

> > **phys_relevant** Keyword, If True, then only return inhibitors that are associated with Regulation instances that have the physiologically-relevant? slot set to True.

> > **slots** Keyword, A list of enzymatic reaction slots.

> **Return value** A list of instances of class Enzymatic-Reactions.

direct_activators(*item*)

> **Description** Return all activators that are connected to an activated object by a single regulation object.

> **Parms**

> > **item** A frame id or PFrame.

> **Return value** A list of frames that activate item.

direct_inhibitors(*item*)

> **Description** Return all inhibitors that are connected to an inhibited object by a single regulation object.
>
> **Parms**
>
> > **item** A frame id or PFrame.
>
> **Return value** A list of frames that inhibit item.

direct_regulators(*item*, *filter_fn=None*)

> **Description** Return all regulators that are connected to a regulated object by a single regulation object.
>
> **Parms**
>
> > **item** A frame id or PFrame.
> >
> > **filter_fn** Keyword, A predicate used to filter the regulation objects used to find the regulators.
>
> **Return value** A list of frames that regulate item.

dna_binding_site_p(*gene*)

> **Description** A predicate that determines if the given frame is an instance of the class DNA-Binding-Sites.
>
> **Parms**
>
> > **gene** A frame id or PFrame.
>
> **Return value** A boolean value.

enzrxn_activators(*er*, *phys_relevant_only=None*)

> **Description** Returns the list of activators (generally small molecules) of the enzymatic reaction frame.
>
> Parms
>
> > **er** An instance of the class Enzymatic-Reactions, a frame id or PFrame.
> >
> > **phys_relevant_only** Keyword, If True, then only return activators that are associated with Regulation instances that have the physiologically-relevant? slot set to True.
>
> **Return value** A list of children of the class Chemicals.

enzrxn_inhibitors(*er*, *phys_relevant_only=None*)

> **Description** Returns the list of inhibitors (generally small molecules) of the enzymatic reaction frame.
>
> Parms
>
> > **er** An instance of the class Enzymatic-Reactions, a frame id or PFrame.
> >
> > **phys_relevant_only** Keyword, If True, then only return inhibitors that are associated with Regulation instances that have the physiologically-relevant? slot set to True.
>
> **Return value** A list of children of the class Chemicals.

enzyme_activity_name(*enzyme*, *reaction=None*)

> **Description** Computes the name of an enzyme in the context of a particular reaction. If the reaction is not provided, then we return the full enzyme name.
>
> **Parms**
>
> > **enzyme** An instance of the class Proteins, that is, a frame id or a PFrame.
> >
> > **reaction** Keyword, An instance of the class Reactions.

Return value A string.

enzyme_p (*protein*, *type=None*)

   Description Predicate that determines whether a specified protein is an enzyme or not.

   Parms

   protein An instance of the class Proteins, a frame id or PFrame.

   type Keyword, Can take on one of the following values to select more precisely what is meant by an "enzyme":

   'any' Any protein that catalyzes a reaction is considered an enzyme.

   'chemical-change' If the reactants and products of the catalyzed reactin differ, and not just by their cellular location, then the protein is considered an enzyme.

   'small-molecule' If the reactants of the catalyzed reaction differ and are small molecules, then the protein is considered an enzyme.

   'transport' If the protein catalyzes a transport reaction.

   'non-transport' If the protein only catalyzes non-transport reactions.

   Return value A boolean value.

enzymes_of_gene (*gene*)

   Description Collects all of the enzymes encoded by the given gene, including modified forms and complexes in which it is a sub-component.

   Parms

   gene An instance of class Genes, a frame id or PFrame.

   Return value A list of instances of class Proteins.

enzymes_of_pathway (*pwy*, *species=None*, *experimental_only=None*, *sorted=None*)

   Description Return all enzymes that are present in the given pathway.

   Parms

   pwy An instance of the class Pathways, a frame id or PFrame.

   species Keyword, A list of species, such that in a multi-organism PGDB such as MetaCyc, only proteins found in those organisms will be returned. This list can include valid org-ids, children of class Organisms, and strings. Please see the documentation for the species slot-unit for more information.

   experimental_only Keyword, When True, only return enzymes that have a non-computational evidence code associated with it.

   sorted Keyword, If True, the enzymes are sorted in the order in which the corresponding reaction occurrs in the sequence of the pathway.

   Return value A list of children of class Proteins or class Protein-RNA-Complexes.

enzymes_of_reaction (*rxn*, *species=None*, *experimental_only=None*, *local_only=None*)

   Description Return the enzymes that catalyze a given reaction.

   Parms

   rxn An instance of the class Reactions, a frame id or PFrame.

species Keyword, A list of species, such that in a multi-organism PGDB such as MetaCyc, only proteins found in those organisms will be returned. This list can include valid org-ids, children of class Organisms, and strings. Please see the documentation for the species slot-unit for more information. Default value is nil.

experimental_only

Keyword, When True, only return enzymes that have

a non-computational evidence code associated with it.

local_only Keyword, When True, only return enzymes that catalyze the specific form of the reaction, as opposed to enzymes that are known to catalyze a more general form (i.e., class) of the reaction.

Return value A list of children of class Proteins or class Protein-RNA-Complexes.

full_enzyme_name(*enzyme*, *use_frame_name=None*, *name=None*, *activity_names=None*)

Description Compute the full name of an enzyme as the concatenation of the common name of the protein followed by the common names of its enzymatic reactions. Note that two enzrxns for the same enzyme could have the same common name, so we avoid including the same name twice.

Parms

enzyme An instance of the class Proteins, that is, a frame id or a PFrame.

use_frame_name Keyword, If True, then the frameid of the enzyme instance is used in computing the enzyme name. Defaults to True.

name Keyword, A string that bypasses the function, and will be returned as the value of the function.

activity_names Keyword, A provided list of strings, that represent the names of the known catalytic activities of enzyme.

Return value A string.

gene_clusters(*genes*, *max_gap=None*)

Description Groups together genes based on whether each gene is a gene neighbor with other genes.

Parms

genes A list of instances of class Genes, a frame id or PFrame.

max_gap Keyword, An integer representing the number of genes any pair from genes can be from one another. Default value is 10.

Return value A list of lists, where the first element of each sub-list is a gene from genes, and the rest of the list are all of the gene neighbors of the first gene.

gene_p(*item*)

Description A predicate to determine if the given frame is a gene.

Parms

item a frame id or PFrame.

Return value A boolean value.

gene_transcription_units(*gene*)

Description Given a gene, return all of the transcription units which contain the gene.

Parms

gene An instance of class Genes, a frame id or PFrame

Return value A list of instances of class Transcription-Units.

genes_in_same_operon (*gene*)

Description Given a gene, return all other genes in the same operon.

Parms

gene An instance of class Genes, a frame id or PFrame.

Return value A list of instances of class Genes.

genes_of_pathway (*pwy*, *sorted=None*)

Description Return all genes coding for enzymes in the given pathway.

Parms

pwy An instance of the class Pathways, a frame id or PFrame.

sorted? Keyword, If True, the genes are sorted in the order in which the corresponding reaction occurrs in the sequence of the pathway.

Return value A list of instances of class Genes.

genes_of_protein (*protein*)

Description Given a protein, return the set of genes which encode all of the monomers of the protein.

Parms

protein An instance of the class Proteins, a frame id or PFrame.

Return value A list of instances of the class Genes.

genes_of_proteins (*protein*)

Description The same as genes_of_protein, except that it takes a list of proteins and returns a set of genes.

Parms

protein A list of instances of the class Proteins, a frame id or PFrame.

Return value A list of instances of the class Genes.

genes_of_reaction (*rxn*)

Description Return all genes that encode the enzymes of a given reaction.

Parms

rxn An instance of the class Reactions, a frame id or PFrame.

Return value A list of instances of class Genes.

genes_regulated_by_gene (*gene*)

Description Return all genes regulated by the given gene by means of a transcription factor.

Parms

gene An instance of the class Genes, a frame id or PFrame.

Return value A list of instances of class Genes.

genes_regulated_by_protein (*protein*)

Description Return all of the genes for which the given protein, or its modified form, acts as a regulator.

> **Parms**
>
>> **protein** An instance of the class Proteins, a frame id or PFrame.
>
> **Return value** A list of instances of the class Genes.

genes_regulating_gene(*gene*)

> **Description** Return all genes regulating the given gene by means of a transcription factor.
>
> **Parms**
>
>> **gene** An instance of the class Genes, a frame id or PFrame.
>
> **Return value** A list of instances of class Genes.

get_class_all_instances(*className*)

> Get all instances of the given class name for this PGDB. ClassName must be exactly as Pathway Tools expect the name of the class, no conversion is applied.
>
> **Parm** className, a symbol specified as a string (e.g., '|**Reactions**|')
>
> **Returns** list of frameids

get_class_all_subs(*classArg*)

> Get all subclasses of the given class name for this PGDB. If classArg is a string, it must be exactly as Pathway Tools expect the name of the class, no conversion is applied.
>
> **Parm** classArg, a symbol specified as a string (e.g., '|**Reactions**|') or as a PFrame.
>
> **Returns** list of frameids corresponding to the subclasses of the classArg.

get_class_data(*realClassName*, *getInstancesData=False*)

> Retrieve the class slots and their values, creating a PFrame for the class. Retrieve also the list of instances from Pathway Tools and create a PFrame for each instance. Store the list of PFrames in attribute 'instances' of the class object. If getInstancesData is True, get also all instances slots and their data.
>
> **Parms** realClassName, a string, the real name of the class to retrieve. getInstancesData, boolean, True => get the slots and data of all instances.
>
> **Returns** A PFrame representing the class with all its slot names as Python attributes.

get_frame_objects(*frameids*)

> For each frame id of the list frameids, retrieve the slots and their data. Reuse the PFrame of frameid if it already exist for this PGDB, otherwise create one and attach it to this PGDB.
>
> **Parm** frameids, list of frame ids (strings).
>
> **Return** list of PFrames, one for each frame id.

get_major_classes()

> Get from Pathway Tools the classes Reactions, Pathways, Genes, Compounds, Proteins, and all their instances with their data. This method is very time consuming has several ten of thousands of frames need to be transferred from Pathway Tools and the corresponding PFrames need to be created.

get_name_string(*item*, *rxn_eqn_as_name=None*, *rxn_common_name_as_name=None*, *direction=None*, *name_slot=None*, *strip_html=None*, *include_species_strain_name=None*, *italicize_species=None*, *short_name=None*, *species_initials=None*, *primary_class=None*)

> **Description** Given an object, compute the string name. The method used to compute the name varies per the object class.
>
> **Parms**
>
>> **item** A frame id or PFrame.

---

rxn_eqn_as_name Keyword, If True, then we use the reaction equation in string form as the name of the reaction. Defaults to True.

rxn_common_name_as_name Keyword, If True, then we use the reaction's common name as the name of the reaction.

direction Keyword, An argument of 'l2r' or 'r2l' can be given to specify the desired reaction orientiation when printed in reaction equation form. If this is not provided, then the reaction direction will be determined using pathway evidence.

name_slot Keyword, The specified slotunit frame name, as a symbol, will be used for extracting the name of the frame.

strip_html Keyword, Remove any HTML mark-up from the string form of the object name.

include_species_strain_name Keyword, Provide proper italicization for the organism strain name.

italicize_species Keyword, Provide proper italicization for the organism species name.

short_name Keyword, If the ABBREV-NAME slot is populated for the frame, then its value will be used.

species_initials Keyword, Print the name of the organism as initials.

primary_class Keyword, Specify explicitly the primary class of the given frame. This can be used to override the internal reasoning of this function, and you can give a suggestion to the function to treat the frame as another class.

Return value A string representing the name of the frame.

get_predecessors (*rxn*, *pwy*)

Description Return a list of all reactions that are direct predecessors (i.e., occurr earlier in the pathway) of the given reaction in the given pathway.

Parms

rxn An instance of the class Reactions, a frame id or PFrame.

pwy A child of the class Pathways.

Return value A list of instances of the class Reactions.

get_slot_value (*frameid*, *slotName*)
Return the single slot value of a frame object. Parms

frameid a string representing the unique identifier for a frame object.

slotName a string representing the slot of the frame object.

Example:

To get the substrates participating on the left of reaction RXN-9000: get_slot_values('RXN-9000', 'LEFT')

get_slot_values (*frameid*, *slotName*)
Return the slot values of a frame object. Parms

frameid a string representing the unique identifier for a frame object.

slotName a string representing the slot of the frame object.

Returns list of values of the given slot. Values can be frameids, booleans, strings, or numbers.

Example:

To get the substrates participating on the left of reaction RXN-9000:
  meta.get_slot_values('RXN-9000', 'LEFT')

where meta is a variable bound to a PGDB object.

get_successors(*rxn*, *pwy*)

  Description Return a list of all reactions that are direct successors (i.e., occurr later in the pathway) of the given reaction in the given pathway.

  Parms

    rxn An instance of the class Reactions, a frame id or PFrame.

    pwy A child of the class Pathways.

  Return value A list of instances of the class Reactions.

homomultimeric_containers_of(*protein*, *exclude_self=None*)

  Description This function is the same as the function containers-of, except that it only includes containers that are homomultimers.

  Parms

    protein An instance of the class Proteins, a frame id or PFrame.

    exclude_self Keyword, If True, then protein will not be included in the return value.

  Return value A list of instances of the class Proteins.

inhibition_p(*reg_frame*)

  Description A predicate that determines if a given regulation frame is describing inhibition.

  Parms

    reg_frame An instance of class Regulation, a frame id or PFrame

  Return value A boolean value.

is_a_class_name(*className*)

  Verify that className is a known class in Pathway Tools. Return the real name of that class because the className may have been transformed by fn class-name-p to generate the closest real class name.

  Parm className, a string, the class name to verify.

  Return a string, either className itself, or modified by replacing '_' to '-' or some case letters changed to match an existing class name in Pathway Tools.

is_an_instance_name(*frameid*)

  Similar to method is_a_class_name but for a frame that is not a class. If frameid is a real frame id of an object of this PGDB, returns frameid as is. If not, try to convert frameid to an exist frame id by transforming cases of letters and underscores to dashes in frameid.

  Parm frameid, a string.

  Returns a string representing an existing frame id in the PGDB.

leader_peptide_p(*protein*)

  Description A predicate that determines whether the given protein is a leader peptide.

  Parms

    protein An instance of the class Proteins, a frame id or PFrame.

Return value A boolean value.

modified_and_unmodified_forms(*protein*)

   Description Returns all of the modified and unmodified forms of the given protein.

   Parms

   protein An instance of the class Proteins, a frame id or PFrame.

   Return value A list of instances of the class Proteins.

modified_containers(*protein*)

   Description Returns all containers of a protein (including itself), and all modified forms of the containers.

   Parms

   protein An instance of the class Proteins, a frame id or PFrame.

   Return value A list of instances of the class Proteins.

modified_forms(*protein*, *exclude_self=None*, *all_variants=None*)

   Description Returns all modified forms of a protein.

   Parms

   protein An instance of the class Proteins, a frame id or PFrame.

   exclude_self Keyword, If True, then protein will not be included in the return value.

   all_variants Keyword, If True, and protein is a modified form, then we return all of the modified forms of the unmodified forms of protein.

   Return value A list of instances of the class Proteins.

monomers_of_protein(*p*, *coefficients=None*, *unmodify=None*)

   Description Returns the monomers of the given protein complex.

   Parms

   p An instance of the class Proteins, a frame id or PFrame.

   coefficients Keyword, If True, then the second return value of the function will be a list of monomer coefficients. Defaults to True.

   unmodify Keyword, If True, obtain the monomers of the unmodified form of p.

   Return value First value is a list of instances of the class Proteins. If coefficients? is True, then the second value is the corresponding coefficients of the monomers fromthe first return value.

neighboring_genes_p(*g1*, *g2*, *n=None*)

   Description Given two genes, this predicate determines if the two genes are "neighbors", or within a certain number of genes from one another along the replicon.

   Parms

   g1 An instance of class Genes, a frame id or PFrame.

   g2 An instance of class Genes, a frame id or PFrame.

n Keyword, An integer representing the number of genes g1 and g2 can be from one another. Default value is 10.

Return value A boolean value.

next_gene_on_replicon(*gene*)

Description Return the next gene on the replicon.

Parms

gene An instance of class Genes, a frame id or PFrame.

Return value Returns two values as a list. The first value is the next gene, or nil if there is not a next gene (i.e., the gene is at the end of a linear replicon). The second value is 'last' if the gene is the last gene on a linear replicon.

noncontiguous_pathway_p(*pwy*)

Description A predicate that determines if the pathway contains more than one connected component. See function pathway-components for more explanation.

Parms

pwy An instance of the class Pathways, a frame id or PFrame.

Return value A boolean value.

nonspecific_forms_of_rxn(*rxn*)

Description Return all of the generic forms of the given specific reaction. Not every reaction will necessarily have a generic form.

Parms

rxn An instance of the class Reactions, that is, a frame id or PFrame.

Return value A list of children of the class Reactions.

operon_of_gene(*gene*)

Description Given a gene, return a list of transcription units that form the operon containing the gene.

Parms

gene An instance of class Genes, a frame id or PFrame

Return value A list of instances of class Transcription-Units.

pathway_allows_enzrxn(*pwy*, *rxn*, *enzrxn*, *single_species=None*)

Description A predicate which returns a True value if the given pathway allows the given enzymatic reaction to catalyze the given reaction. Certain pathways have a list of enzymatic reactions that are known not to catalyze certain reactions. See the documentation of slot-unit enzyme-use for more information.

Parms

pwy An instance of the class Pathways, a frame id or PFrame.

rxn An instance of the class Reactions, a frame id or PFrame.

enzrxn An instance of the class Enzymatic-Reactions, a frame id or PFrame.

single_species Keyword, An instance of the class Organisms If set, then enzrxn has the further stricture that it must be an enzymatic reaction present in the organism specified by the value passed to single-species.

Return value A boolean value.

pathway_components (*pwy*, *rxn_list=None*, *pred_list=None*)

Description Returns all of the connected components of a pathway. A connected component of a pathway is a set of reactions in the pathway such that for all reactions R1 in the connected component, a predecessor relationship holds between R1 and some other reaction R2 in the connected component, and each connected component is of maximal size. Every pathway will have from 1 to N connected components, where N is the number of reactions in the pathway. Most pathways have one connected component, but not all.

Parms

pwy, a frame id or PFrame. An instance of the class Pathways, which is not a super-pathway (i.e., does not have any entries in its sub-pathways slot).

rxn_list Keyword, The list of reactions to use as the starting list of connected component clusters. Defaults to the content of slot reaction-list in pwy.

pred_list Keyword, The list of reaction predecessors to iterate from in order to cluster the reactions in rxn-list. Defaults to list in slot predecessors of pwy.

Return value Returns three values as a list: the connected components as a list of lists of the form ((r1 r2 r3) (r4 r5) (r6 r7 r8)) where each sub-list contains all reactions in one connected component, the number of connected components, and the length of the reaction list.

pathway_hole_p (*rxn*, *hole_if_any_gene_without_position=None*)

Description A predicate that determines if the current reaction is considered to be a 'pathway hole', or without an associated enzyme.

Parms

rxn An instance of the class Reactions, a frame id or PFrame.

hole_if_any_gene_without_position Keyword, If True, then genes without specified coordinates for the current organism's genome are not counted when determining the status of the reaction.

Return value A boolean value.

pathways_of_compound (*cpd*, *non_specific_too=None*, *modulators=None*, *phys_relevant=None*, *include_rxns=None*)

Description Returns all pathways in which the given compound appears as a substrate.

Parms

cpd An instance of class Compounds, a frame id or PFrame.

non-specific_too Keyword, If True, returns all generic reactions where cpd, or a parent of cpd, appears as a substrate.

modulators Keyword, If True, returns pathways where cpd appears as a regulator as well.

phys-relevant Keyword, If True, then only return inhibitors that are associated with Regulation instances that have the physiologically-relevant? slot set to True.

include-rxns Keyword, If True, then return a list of reaction-pathway pairs.

Return value A list of instances of class Pathways. If include-rxns? is True, then a list of lists, where each sub-list consists of an instance of class Reactions and an instance of class Pathways.

pathways_of_enzrxn (*enzrxn*, *include_super_pwys=None*)

Description Returns the list of pathways in which the given enzymatic reaction participates.

Parms

> **enzrxn** An instance of the class Enzymatic-Reactions, a frame id or PFrame.
>
> **include_super_pwys** Keyword, If True, then not only will the direct pathways in which enzrxn is associated in be returned, but also any enclosing super-pathways. If enzrxn is associated with a reaction that is directly associated with a super-pathway, then the function might return super-pathways even if this option is nil.

Return value A list of instances of class Pathways.

pathways_of_gene (*gene*, *include_super_pwys=None*)

> **Description** Returns the pathways of enzymes encoded by the given gene.
>
> **Parms**
>
> > **gene** An instance of class Genes, a frame id or PFrame.
> >
> > **include_super_pwys** Keyword, If True, then not only will the direct pathways in which gene encodes an enzyme be returned, but also any enclosing super-pathways. If gene is associated with a reaction that is directly associated with a super-pathway, then the function might return super-pathways even if this option is nil.
>
> **Return value** A list of instances of class Pathways.

phantom_gene_p (*gene*)

> **Description** A predicate that determines if the given gene is a phantom gene.
>
> **Parms**
>
> > **gene** An instance of the class Genes, a frame id or PFrame.
>
> **Return value** A boolean value.

polypeptide_or_homomultimer_p (*protein*)

> **Description** A predicate that determines if the given protein is a polypeptide or a homomultimer.
>
> Parms
>
> > **protein** An instance of the class Proteins, a frame id or PFrame.
>
> **Return value** A boolean value.

previous_gene_on_replicon (*gene*)

> **Description** Return the previous gene on the replicon.
>
> **Parms**
>
> > **gene** An instance of class Genes, a frame id or PFrame.
>
> **Return value** Returns two values as a list. The first value is the previous gene, or nil if there is not a previous gene (i.e., the gene is at the beginning of a linear replicon). The second value is 'first' if the gene is the first gene on a linear replicon.

promoter_binding_sites (*promoter*)

> **Description** Returns all of the binding sites associated with the given promoter, across multiple transcription units.
>
> **Parms**
>
> > **promoter** An instance of class Promoters, a frame id or PFrame.

Return value A list of instances of class DNA-Binding-Sites.

protein_coding_gene(*gene*)

Description A predicate that determines if the given gene encodes a protein (as opposed to an RNA).

Parms

gene An instance of the class Genes, a frame id or PFrame.

Return value A boolean value.

protein_in_compartment_p(*rxn*, *compartments*, *default_ok=None*, *pwy=None*, *loose=None*)

Description A predicate that checks if the given reaction is present in a list of cellular compartments.

Parms

rxn An instance of the class Reactions, a frame id or PFrame.

compartments A list of cellular compartments, as defined in the Cellular Components Ontology. See frame CCO.

default_ok Keyword, If True, then we return True if the reaction has no associated compartment information, or one of its associated locations is a super-class of one of the members of the compartments parameter.

pwy Keyword, a frame id or PFrame. If supplied, the search for associated enzymes of the parameter rxn is limited to the given child of Pathways.

loose Keyword, If True, then the compartments CCO-CYTOPLASM and CCO-CYTOSOL are treated as being the same compartment.

Return value A boolean value.

protein_or_rna_containers_of(*protein*, *exclude_self=None*)

Description This function is the same as the function containers-of, except that it only includes containers that are instances of either class Protein-Complexes, or class Protein-RNA-Complexes.

Parms

protein An instance of the class Proteins, a frame id or PFrame.

exclude_self Keyword, If True, then protein will not be included in the return value.

Return value A list of instances of the class Proteins.

protein_p(*frame*)

Description A predicate that determines whether the given frame is a protein.

Parms

frame a frame id or PFrame.

Return value A boolean value.

pseudo_gene_p(*gene*)

Description A predicate that determines if the given gene is a pseudo-gene.

Parms

gene An instance of the class Genes, a frame id or PFrame.

Return value A boolean value.

put_slot_value(*frameid*, *slotName*, *val*)
  Modify the slot value of a frame object with the given val. Val is a single value (e.g., not a list).

  Important: The modified frame is not updated for any PFrame object that might has been
  previously loaded from that PGDB. This operation should be used only for its effect on the PGDB in the running Pathway Tools application.

  Parms

  > **frameid** a string representing the unique identifier for a frame object.

  > **slotName** a string representing the slot of the frame object.

  > **val** a value to store in the slot.

  Side-Effects The slot of that frame is replaced with the new value

  Returns Nothing

  Example:

  > **To put the Gibbs free energy of reaction RXN-9000:** put_slot_value('RXN-9000',   'GIBBS-0', 7.52)

put_slot_values(*frameid*, *slotName*, *val*)
  Modify the slot values of a frame object with the given val. Val is typically a list of objects.

  Important: The modified frame is not updated for any PFrame object that might has been
  previously loaded from that PGDB. This operation should be used only for its effect on the PGDB in the running Pathway Tools application.

  Parms

  > **frameid** a string representing the unique identifier for a frame object.

  > **slotName** a string representing the slot of the frame object.

  > **val** a value to store in the slot, typically a list of values or objects.

  Side-Effects The slot of that frame is replaced with the new value.

  Returns Nothing

  Example:

  > **To put the substrates participating on the left of reaction RXN-9000:** put_slot_values('RXN-9000', 'LEFT', ['CPD-9459','CPD-9460'])

reaction_reactants_and_products(*rxn*, *direction=None*, *pwy=None*)

  Description Return the reactants and products of a reaction, based on a specified direction. The direction can be specified explicity or by giving a pathway as an argument. It is an error to both specify the pathway and the explicit direction. If neither an explicit direction or a pathway is given as an argument, then the direction is computationally inferred from available evidence within the PGDB.

  Parms

  > **rxn** An instance of the class Reactions, that is, a frame id or PFrame.

  > **direction** Keyword, Can take on the following values:

  > > **'L2R'** The reaction direction goes from 'left to right', as described in the Reactions instance.

  > > **'R2L'** The reaction direction goes from 'right to left'; the opposite of what is described in the Reactions instance.

  pwy Keyword, An instance of the class Pathways, a frame id or PFrame.

 Return value Returns multiple values as a list. The first value is a list of reactants as determined by the direction of the reaction, and the second value is a list of the products as determined by the direction of the reaction. Both lists have items that are children of class Compounds, children of class Polymer-Segments, or strings.

reaction_type(*rxn*)

 Description Returns a keyword describing the type of reaction.

 Parms

  rxn An instance of the class Reactions, a frame id or PFrame.

 Return value A string from the following list:

  'small-molecule' All substrates are small molecules, or small-molecule classes.

  'transport' A substrate is marked with different compartment annotations in the left and right slots.

  'protein-small-molecule-reaction' At least one substrate is a protein and at least one is a small molecule.

  'protein-reaction' All substrates are proteins.

  'trna-reaction' At least one substrate is a tRNA.

  'null-reaction' No substrates or reactants are specified.

  'other' None of the preceding cases apply.

reactions_of_compound(*cpd*, *non_specific_too=None*, *transport_only=None*, *compartment=None*, *enzymatic=None*)

 Description Return all reactions in which the given compound participates as a substrate.

 Parms

  cpd, a frame id or PFrame. A child of class Compounds.

  non_specific_too Keyword, If True, returns all generic reactions where cpd, or a parent of cpd, appears as a substrate.

  transport_only Keyword, If True, return only transport reactions.

  compartment Keyword, If True, return only reactions within the specified compartment.

  enzymatic Keyword, If True, return only enzymatic reactions.

 Return value A list of children of class Reactions.

reactions_of_enzyme(*protein*, *kb=None*, *include_specific_forms=None*)

 Description Return all of the reactions associated with a given protein via enzymatic reactions.

 Parms

  protein An instance of the class Proteins, a frame id or PFrame.

  kb Keyword, The KB object of the KB in which to find the associated reactions. Defaults to self.

  include_specific_forms Keyword, When True, specific forms of associated generic reactions are also returned. Default value is True.

 Return value A list of instances of the class Reactions.

reactions_of_gene(*gene*)

Description Returns all reactions catalyzed by enzymes encoded by the given gene.

Parms

> **gene** An instance of class Genes, a frame id or PFrame.

Return value A list of instances of class Reactions.

reactions_of_protein(*protein*, *check_protein_components=None*, *check_protein_containers=None*)

Description Returns all of the associated reactions that the given protein, or its components, catalyzes.

Parms

> **protein** An instance of the class Proteins, a frame id or PFrame.
>
> **check_protein_components?** Keyword, If True, check all components of this protein for catalyzed reactions. Defaults to True.
>
> **check_protein_containers?** Keyword, If True, check the containers and modified forms of the protein for catalyzed reactions.

Return value A list of instances of class Reactions.

reduce_modified_proteins(*prots*, *debind=None*)

Description Given a list of proteins, the function converts all of the proteins to their unmodified form, and then removes any duplicates from the subsequent list.

Parms

> **prots** A list of instances of the class Proteins, a frame id or PFrame.
>
> **debind** Keyword, When True, the proteins are further simplified by obtaining the unbound form of the protein, if it is bound to a small molecule.

Return Value A list of instances of the class Proteins.

regulation_frame_transcription_units(*reg_frame*)

Description Given a regulation object, return the transcription units when one of the regulated entities is a promoter or terminator of the transcription unit.

Parms

> **reg_frame** An instance of the class Regulation-of-Transcription, a frame id or PFrame.

Return value A list of instances of the class Transcription-Units.

regulator_of_type(*protein*, *class_name*)

Description A predicate that determines if the given protein is a regulator of the specified class.

Parms

> **protein** An instance frame of class Proteins, a frame id or PFrame.
>
> **class** A subclass of Regulation.

Return value A boolean value.

regulator_proteins_of_transcription_unit(*tu*)

Description Returns all transcription factors that regulate the given transcription unit.

Parms

> **tu** An instance of the class Transcription-Units, a frame id or PFrame.

Return value A list of instances of the class Proteins.

regulators_of_gene_transcription(*gene*, *by_function=None*)

Description Returns a list of proteins that are regulators of the given gene.

Parms

> **gene** An instance of the class Genes, a frame id or PFrame.
>
> **by_function** Keyword, If True, then return two values: a list of activator proteins and a list of inhibitor proteins.

Return value A list of instances of class Proteins. If by_function is True, then two values are returned. The first value is a list of activator proteins, and the second value is a list of inhibitor proteins.

regulators_of_operon_transcription(*operon_list*, *by_function=None*)

Description Returns a list of transcription factors of an operon.

Parms

> **operon_list** A list of instances of the class Transcription-Units, a frame id or PFrame.
>
> **by_function** Keyword, If True, then return two values: a list of activator proteins and a list of inhibitor proteins.

Return value A list of instances of class Proteins. If the modified form of the protein is the transcription factor, then that is the protein returned.

regulon_of_protein(*protein*)

Description Returns all transcription units regulated by any form of the given protein.

Parms

> **protein** An instance frame of class Proteins, a frame id or PFrame.

Return value A list of instances of the class Transcription-Units.

rna_coding_gene(*gene*)

Description A predicate that determines if the given gene encodes an RNA.

Parms

> **gene** An instance of the class Genes, a frame id or PFrame.

Return value A boolean value.

run_fba(*fileName*)
In PythonCyc there is a run_fba method defined globally in the pythoncyc module and there is this version which is run under a specific PGDB. Notice though that the FBA input file provided will decide which organism is used for running FBA and may override this PGDB.

Parms

> **fileName, a string which is the name of the FBA input file on the** running Pathway Tools machine.

Returns

> **A list with the following values:**
>
> > 1. True <=> success, the FBA completed without error (for growth, see 5)
> >
> > 2. List of error messages, if any

3. List of output messages generated by MetaFlux (FBA module) during parsing and execution

4. The solver (SCIP) status symbol

5. The flux of the objective biomass, non-zero if growth

6. Number of growth cases if FBA input file is a knockout run, or the number of active reactions if the FBA input file is solving a model

7. The list of reactions that were in the model after instantiation

8. The list of reactions that were active (non zero flux) with their fluxes

rxn_in_compartment_p (*rxn*, *compartments*, *default_ok=None*, *pwy=None*, *loose=None*)

> **Description** A predicate that checks if the given reaction is present in a list of cellular compartments.
>
> **Parms**
>
>> **rxn** An instance of the class Reactions, a frame id or PFrame.
>>
>> **compartments** A list of cellular compartments, as defined in the Cellular Components Ontology. See frame CCO.
>>
>> **default_ok** Keyword, If True, then we return True if the reaction has no associated compartment information, or one of its associated locations is a super-class of one of the members of the compartments parameter.
>>
>> **pwy** Keyword, a frame id or PFrame. If supplied, the search for associated enzymes of the parameter rxn is limited to the given child of Pathways.
>>
>> **loose** Keyword, boolean. If True, then the compartments CCO-CYTOPLASM and CCO-CYTOSOL are treated as being the same compartment.
>
> **Return value** A boolean value.

rxn_present_p (*rxn*)

> **Description** A predicate that determines if there is evidence for the occurrence of the given reaction in the current PGDB.
>
> **Parms**
>
>> **rxn** An instance of the class Reactions, that is, a frame id or PFrame.
>
> **Return value** A boolean value.

rxn_specific_form_of_rxn_p (*specific_rxn*, *generic_rxn*)

> **Description** A predicate that is True if the given generic reaction is a generalized form of the given specific reaction.
>
> **Parms**
>
>> **specific_rxn** A child of the class Reactions, that is, a frame id or PFrame.
>>
>> **generic_rxn** A child of the class Reactions, that is, a frame id or PFrame.
>
> **Return value** A boolean value.

rxn_w_isozymes_p (*rxn*)

> **Description** A predicate that tests if a given reaction has any associated isozymes (distinct proteins or protein classes that catalyze the same reaction).
>
> **Parms**
>
>> **rxn** An instance of the class Reactions, a frame id or PFrame.

      Return value A boolean value.

rxn_without_sequenced_enzyme_p(*rxn*, *complete=None*)

      Description A predicate that tests if a given reaction has genes with no associated sequence information.

      Parms

            rxn An instance of the class Reactions, that is, a frame id or PFrame.

            complete Keyword, if True, the predicate will return True when there is any associated gene without a sequence. If False, the predicate will return True when all associated genes are without a sequence.

      Return value A boolean value.

rxns_adjacent_in_pwy_p(*rxn1*, *rxn2*, *pwy*)

      Description A predicate to determine if two given reactions are adjacent to one another in the given pathway.

      Parms

            rxn1 An instance of the class Reactions, a frame id or PFrame.

            rxn2 An instance of the class Reactions, a frame id or PFrame.

            pwy An instance of the class Pathways, a frame id or PFrame.

      Return value A boolean value.

rxns_catalyzed_by_complex(*rxns=None*)

      Description Enumerates all reactions catalyzed by an enzyme that is a protein complex.

      Parms

            rxns Keyword, A list of instances of the class Reactions. Defaults to the result of (all-rxns :enzyme).

      Return value A list of instances of the class Reactions with a protein complex as an enzyme.

rxns_w_isozymes(*rxns=None*)

      Description Enumerate all reactions that have isozymes (distinct proteins or protein classes that catalyze the same reaction).

      Parms

            rxns Keyword, A list of instances of the class Reactions. Defaults to the result of (all-rxns :enzyme).

      Return value A list of A list of instances of the class Reactions with isozymes.

save_pgdb()

      Save a PGDB that has been modified in the running Pathway Tools server. The PGDB that will be saved is based on the orgid of this PGDB object.

sendPgdbFnCall(*fn*, **args*, ***kwargs*)

      Send a PGDB query to Pathway Tools based on function fn and arguments args and kwargs (keyword args) and return the result. If multiple values are returned by fn, the Pathway Tools Python server transforms them into a list.

sendPgdbFnCallBool(*fn*, **args*, ***kwargs*)

      Send a PGDB query to Pathway Tools that will return a Bool value. This method takes care of translating no value or an empty list to False.

sendPgdbFnCallList (*fn*, **args*, ***kwargs*)
Send a PGDB query to Pathway Tools that will return a List value. This method takes care of translating no value or False to an empty list.

sendPgdbQuery (*query*)
Send a query for a specific PGDB using its orgid. Use the macro with-organism for the Lisp Python server.

Parm query, a string. That string should be acceptable to the Lisp Python server.

Return the result (as a Python object) of the execution of the query in Pathway Tools.

small_molecule_cplxes_of_prot (*protein*)

Description Return all of the forms of the given protein that are complexes with small molecules.

Parms

protein An instance of the class Proteins, a frame id or PFrame.

Return value A list of instances of the class Proteins.

species_of_protein (*protein*)

Description Get the associated species for the given protein.

Parms

protein A list of instances of the class Proteins, a frame id or PFrame.

Return value An instance of the class Organisms, or a string.

specific_forms_of_rxn (*rxn*)

Description Return all of the specific forms of the given generic reaction. Not every reaction will necessarily have a specific form.

Parms

rxn A child of the class Reactions, that is, a frame id or PFrame.

Return value A list of instances of the class Reactions.

substrate_of_generic_rxn (*cpd*, *rxn*)

Description A predicate that determines if a parent of the given compound is a substrate of the given generic reaction.

Parms

cpd An instance of class Compounds, a frame id or PFrame.

rxn An instance of class Reactions, a frame id or PFrame.

Return value A boolean value.

substrates_of_pathway (*pwy*)

Description Bearing in mind the direction of all reactions within a pathway, this function returns the substrates of the reactions in four groups: a list of all reactant compounds (compounds occurring on the left side of some reaction in the given pathway), the list of proper reactants (the subset of reactants that are not also products), a list of all products, and a list of all proper products.

Parms

pwy An instance of the class Pathways, a frame id or PFrame.

Return value Four values as a list, each of which is a list of substrates. A substrate may be a child of class Compounds, a child of class Polymer-Segments, or a string.

substrates_of_reaction(*rxn*)

Description Return all of the reactants and products of a given reaction.

Parms

rxn An instance of the class Reactions, a frame id or PFrame.

Return value A list that may consist of children of class Compounds, children of class Polymer-Segments, or strings.

terminator_p(*gene*)

Description A predicate that determines if the given object is an instance of the class Terminators.

Parms

gene A frame id or PFrame.

Return value A boolean value.

terminators_affecting_gene(*gene*)

Description Find terminators in the same transcription unit and upstream of the given gene.

Parms

gene An instance of class Genes, a frame id or PFrame

Return value A list of instances of class Terminators.

tfs_bound_to_compound(*cpd*, *include_inactive=None*)

Description Returns a list of protein complexes that, when bound to the given compound, act as a transcription factor.

Parms

cpd An instance of class Compounds, a frame id or PFrame.

include_inactive Keyword, If True, then the inactive form of the protein is also checked. See the function transcription-factor? for more information.

Return value A list of instances of class Proteins.

top_containers(*protein*)

Description Return the top-most containers (i.e., they are not a component of any other protein complex) of the given protein.

Parms

protein An instance of the class Proteins, a frame id or PFrame.

Return value A list of instances of the class Proteins.

transcription_factor_active_forms(*tfs*)

Description For a given transcription factor, find all active forms (i.e, form of the protein that regulates) of the transcription factor.

Parms

tfs, a frame id or PFrame. An instance of the class Proteins.

Return value A list of instances of the class Proteins.

transcription_factor_ligands(*tfs*, *mode*)

Description For a single transcription factor or list of transcription factors, return all transcription factor ligands.

Parms

tfs, a frame id or PFrame or a list of these. An instance or a list of instances of the class Proteins. If tfs is not the active form, then the active form is determined automatically.

mode One of the following values: 'activator', 'inhibitor', or 'both'.

Return value A list of instances of the class Chemicals or strings.

transcription_factor_p(*protein*, *include_inactive=None*)

Description A predicate that determines if the given protein is a transcription factor, or a component of a transcription factor.

Parms

protein An instance frame of class Proteins, a frame id or PFrame.

include_inactive Keyword, If True, then the function checks to see if any of its components or containers is a transcription factor as well.

Return value A boolean value.

transcription_unit_activation_frames(*tu*)

Description Returns a list of regulation frames that activate the transcription unit.

Parms

tu An instance of the class Transcription-Units, a frame id or PFrame.

Return value A list of instances of the class Regulation.

transcription_unit_activators(*tu*)

Description Returns all activator proteins of the given transcription unit.

Parms

tu An instance of the class Transcription-Units, a frame id or PFrame.

Return value A list of instances of class Proteins.

transcription_unit_all_components(*tu*)

Description Returns all components (binding sites, promoters, genes, terminators) of the given transcription unit.

Parms

tu An instance of the class Transcription-Units, a frame id or PFrame.

Return value A list of instances of class Transcription-Units, mRNA-Binding-Sites, DNA-Binding-Sites, Promoters, Genes, or Terminators.

transcription_unit_binding_sites(*tu*)

Description Returns the binding sites of the given transcription unit.

Parms

tu An instance of the class Transcription-Units, a frame id or PFrame.

Return value A list of instances of class DNA-Binding-Sites.

transcription_unit_first_gene(*tu*)

    Description Returns the first gene of the given transcription unit.

    Parms

        tu An instance of the class Transcription-Units, a frame id or PFrame.

    Return value An instance of class Genes.

transcription_unit_genes(*tu*)

    Description Returns the genes of the given transcription unit.

    Parms

        tu An instance of the class Transcription-Units, a frame id or PFrame.

    Return value A list of instances of class Genes.

transcription_unit_inhibition_frames(*tu*)

    Description Returns a list of regulation frames that inhibit the transcription unit.

    Parms

        tu An instance of the class Transcription-Units, a frame id or PFrame.

    Return value A list of instances of the class Regulation.

transcription_unit_inhibitors(*tu*)

    Description Returns all inhibitor proteins of the given transcription unit.

    Parms

        tu An instance of the class Transcription-Units, a frame id or PFrame.

    Return value A list of instances of class Proteins.

transcription_unit_mrna_binding_sites(*tu*)

    Description Returns the mRNA binding sites of the given transcription unit.

    Parms

        tu An instance of the class Transcription-Units, a frame id or PFrame.

    Return value A list of instances of class mRNA-Binding-Sites.

transcription_unit_promoter(*tu*)

    Description Returns the promoter of the given transcription unit.

    Parms

        tu An instance of the class Transcription-Units, a frame id or PFrame.

    Return value An instance of class Promoters.

transcription_unit_regulation_frames(*tu*)

    Description Returns a list of regulation frames that regulate the transcription unit.

    Parms

        tu An instance of the class Transcription-Units, a frame id or PFrame.

    Return value A list of instances of the class Regulation.

transcription_unit_terminators(*operon*)

    Description Returns the terminators of the given transcription unit.

    Parms

        operon An instance of the class Transcription-Units, a frame id or PFrame.

    Return value A list of instances of class Terminators.

transcription_unit_transcription_factors(*tu*)

    Description Returns the binding sites of the given transcription unit.

    Parms

        tu An instance of the class Transcription-Units, a frame id or PFrame.

    Return value A list of instances of class DNA-Binding-Sites.

transcription_units_of_promoter(*promoter*)

    Description Returns all transcription units of a given promoter.

    Parms

        promoter An instance of class Promoters, a frame id or PFrame.

    Return value A list of instances of class Transcription-Units.

transcription_units_of_protein(*protein*)

    Description Return all of the transcription units for which the given protein, or its modified form, acts as a regulator.

    Parms

        protein An instance of the class Proteins, a frame id or PFrame.

    Return value A list of instances of the class Transcription-Units.

transported_chemicals(*rxn*, *side=None*, *primary_only=None*, *from_compartment=None*, *to_compartment=None*, *show_compartment=None*)

    Description Return the compounds in a transport reaction that change compartments.

    Parms

        rxn An instance of the class Reactions, a frame id or PFrame.

        side Keyword, The side of the reaction from which to return the transported compound.

        primary_only Keyword, If True, then filter out common exchangers (currently defined as (PROTON NA CPD-1)+). If True, and the only transported compounds are in this list, then the filter doesn't apply.

        from_compartment Keyword, A compartment (child of class CCO). If specified, then only return compounds transported from that compartment.

        to_compartment Keyword, A compartment (child of class CCO). If specified, then only return compounds transported to that compartment.

        show_compartment Keyword, A compartment (child of class CCO). If specified, and the compound is modified during transport, then only return the form of the compound as found in this compartment.

    Return value A list of children of class Compounds.

unmodified_form(*protein*)

Description Return the unmodified form of the given protein, which might be the same as the given protein.

Parms

protein An instance of the class Proteins, a frame id or PFrame.

Return value An instance of the class Proteins.

unmodified_gene_product(*gene*)

Description Returns the first element of the list returned by the function unmodified-gene-products. This is useful if you are sure that there are no alternative splice forms of your gene.

Parms

gene An instance of class Genes, a frame id or PFrame.

Return value An instance of either class Polypeptides or 'RNA.

unmodified_gene_products(*gene*)

Description Return all of the unmodified gene products (i.e. alternative splice forms) of the given gene.

Parms

gene An instance of class Genes, a frame id or PFrame.

Return value A list of instances of either class Polypeptides or 'RNA.

unmodified_or_unbound_form(*protein*)

Description Return the unmodified form or unbound (to a small molecule) form of the given protein, which might be the same as the given protein.

Parms

protein An instance of the class Proteins, a frame id or PFrame.

Return value An instance of the class Proteins.

variants_of_pathway(*pwy*)

Description Returns all variants of a pathway.

Parms

pwy An instance of the class Pathways, a frame id or PFrame.

Return value A list of instance of the class Pathways.

pythoncyc.PGDB.**convertArgToLisp**(*arg*, *inquote=False*)
    Convert the arg into an acceptable quoted object for Python server running on Pathway Tools. Note that any list is converted to a quoted Lisp list.

Parm arg, a PFrame, a string, a number, a boolean, None or an s-expr. inquote, a Boolean, True => this arg is inside an already quoted expression.

Return a string, the arg is transformed to be acceptable for the Python Lisp server.

pythoncyc.PGDB.**may_be_frameid**(*x*)
    This fn is useful to convert a string to a symbol or a list of strings to a list of symbols, to make sure it is interpreted as a frame id; but not to apply any conversion when the arg is not a string or a list of strings. All functions of PGDB.py apply this fn on the arguments that need a frame ids or PFrames.

Parm x: a Python object.

Side Effect Raise an error, if x is not None, String, PFrame, or a list of these.

Returns A symbol, list of symbols, or x unchanged.

pythoncyc.PGDB.**mkey**(*s*)
A simple function to convert a string into a Lisp keyword.

Parm s, any Python object

Returns if s is a string, it is suffixed by ':', otherwise s itself.

pythoncyc.PGDB.**prepareFnCall**(*fn*, **args*, ***kwargs*)
Prepare all arguments and keyword arguments for a function call to Pathway Tools. Parms

fn, a string, the name of the Lisp function to call. args, list of arguments kwargs, list of keyword arguments

Return a string which represents a Lisp fn call with args and keyword args.

1.3 pythoncyc.PTools module

This module handles basic operations for receiving and sending messages via a network socket to Pathway Tools.

No major class is defined in this file, but only toplevel functions and some simple classes for errors handling.

exception pythoncyc.PTools.**PToolsError**
Bases: exceptions.Exception

Error generated when Pathway Tools send an error due to its own Lisp execution.

exception pythoncyc.PTools.**PythonCycError**
Bases: exceptions.Exception

Error generated by one of the module of PythonCyc due to an incorrect use of its methods or functions.

pythoncyc.PTools.**recvAll**(*s*)
Receive the entire message sent by Pathway Tools on socket s. The message starts with a single character type, which is either 'A' or 'L'. The 'A' time is used without providing a length but can take longer to receive because it uses a timeout technique to read the entire message. The 'L' type assumes that the length of the message, in characters, is given on the next 10 characters as an integer. The message length is the number of characters after these 10 characters.

Parm s, an open network socket.

Return the message received on socket s as a string.

pythoncyc.PTools.**recvFixedLength**(*s*, *lengthMsg*)
Receive a fixed length message on socket s. Parm

lengthMsg, an integer, which is the length in characters of the message to receive.

Return the message received as a string.

pythoncyc.PTools.**recvTimeOut**(*socket*, *timeOut=2*)
Receive a message of unknown length on socket. While receiving a message, if no more characters are sent on socket after timeOut seconds, it is assumed that the message has ended. Therefore, it will always, whatever the lenght of the message, take at least timeOut seconds to execute this method. If no character is received after 60 seconds, this method returns with an empty message.

Parms socket, an open network socket. timeOut, number of seconds before timing out between fragments of the received

> message.

Return The received message, as a string, on socket.

pythoncyc.PTools.**sendAll**(*s*, *query*)

pythoncyc.PTools.**sendQueryToPTools**(*query*)
> Send a query to a running Pathway Tools application via a socket.

> **Parm** query, a string that the Python server in Pathway Tools can evaluate.

> **Returns** The result of the query, as a Python object, decoded by Json.

1.4 pythoncyc.PToolsFrame module

Please consult file tutorial.html under the doc directory for an introduction on how to use this class.

class pythoncyc.PToolsFrame.**PFrame**(*frameid*, *pgdb*, *getFrameData=False*, *isClass=False*)
> PFrame is a class to represent Pathway Tools' frames. A PFrame can represent a class frame (e.g., Reactions) as well as an instance frame (e.g., RXN-9000).

> The required parameters to create a PFrame are the frame id (a string) and a PGDB object. For example, assuming that 'meta' is bound to a PGDB object, the following create a PFrame to represent the reaction RXN-9000,

> > PFrame('RXN-9000', meta)

> By default, an instance PFrame (not a class PFrame) is created. To create a PGDB object, use class PGDB, or call method pythoncyc.so.

> A few methods were written to display frames using HTML in IPython. Naturally, this functionality is only defined when using IPython as a Python interpreter.

> Please consult file tutorial.html under the doc directory for an introduction on how to use this class.

> **get_frame_data**()
> > Retrieve the frame data from Pathway Tools, that is, all slots and their values for this PFrame are retrieved and stored locally. For a class, the instances are not retrieved by this method. Instead, use method get_class_data applied to a PGDB object.

> > **Return** the self PFrame, modified with the new slots and data.

> **get_frame_slot_value**(*slot*)
> > Retrieve the slot data for frame from Pathway Tools.

> > **Return** the self PFrame, modified with the new slot with data.

> **vectorize_dict**()

class pythoncyc.PToolsFrame.**Symbol**(*name*)
> This class can be used to represent Lisp symbols in PythonCyc. For example, Symbol('trp') represents the symbol trp. It is mainly used to ensure that some arguments of methods are interpreted as symbols and not as strings during the translation to send a request to Pathway Tools.

pythoncyc.PToolsFrame.**convertLispIdtoPythonId**(*s*)
> Convert string s such that it can be a valid Python identifier.

1.5 pythoncyc.config module

This module can be used to set various parameters for PythonCyc, in particular to set debug on or off, the host name and port number of the running Pathway Tools' Python server. By default, the Pathway Tools Python server is running locally on port 5008.

`pythoncyc.config.`**`set_debug_off`**`()`
> Turn off debug mode for PythonCyc. Turning off debugging should turn off all output tracings of the communications between PythonCyc and Pathway Tools.

`pythoncyc.config.`**`set_debug_on`**`()`
> Turn on debug mode for PythonCyc. Turning on debugging should turn on output tracings of the communications between PythonCyc and Pathway Tools.

`pythoncyc.config.`**`set_host_name`**`(`*hostname*`)`

`pythoncyc.config.`**`set_host_port`**`(`*hostport*`)`

1.6 Module contents

This module is part of PythonCyc, a Python interface module to Pathway Tools. This code has been tested with Python 2.6.

Pathway Tools (version 18.5 and up) must be running on some machine with at least the option '-python'. It is also recommended to start Pathway Tools with the option -lisp, so that the connection can be monitored:

./pathway-tools -lisp -python

The global functions defined in this init file can be called before any PGDB (an organism database in Pathway Tools) has been selected. In fact, two of these functions, select_organism and its synonym so, are needed to "select" a PGDB by creating a PGDB object. See class PGDB in PGDB.py for information about how to use a PGDB object.

Please consult the tutorial.html file, under directory doc, for more information about how to use PythonCyc.

`pythoncyc.`**`all_orgids`**`()`
> Returns all organism unique ids (orgids) available from the current running Pathway Tools.

`pythoncyc.`**`biovelo`**`(`*query*`)`
> Execute a BioVelo query and return the result.

> **Parameters** query: a string, which is a BioVelo query.

> **Returns** Whatever the BioVelo query computes.

> **Example** bv('[(p, reactions-of-pathway(p)): p<-ecoli^^pathways]')

`pythoncyc.`**`bv`**`(`*query*`)`
> A synonym of method biovelo.

`pythoncyc.`**`run_fba`**`(`*fileName*`)`
> The function run_fba does not need to have an organism selected before being used because the FBA input file provided as input can specify the organism.

> For the documentation of this function, see method run_fba in file PGDB.py.

`pythoncyc.`**`select_organism`**`(`*orgid*`)`
> **Select an organism PGDB based on its unique organism id.** orgid: string, the unique organism id in Pathway Tools (e.g., ecoli, meta).

`pythoncyc.`**`so`**(*orgid*)
   A synonym of method select_organism.

Indices and tables

- genindex
- modindex
- search

p