

reactome_utility_module.pl --Reactome Utility Predicates



- This module has some useful predicates for interacting with the
- owl:rdf ontology file for Reactome
- @author Sam Neaves

reactome_string_id(+R:reactome string, -Id:reactome id) is **semidet**



Reactome_String is a string of the form: 'http://www.reactome.org/biopax/47/48887#Protein1'

reactome_name(?Rid:reactome_id, ?Name:atom) is semidet



Use with single quotes works in both directions.

reactome_name_fast(+Rid:reactome_id, -Name:atom) is **semidet**



Use with single quotes only works in direction of id to name.

protein_reactome_id_to_uniprot_id(?Reactome_Id:protein_id, ?UniprotId:uniprot_id) is semidet



Use with single quotes protein_reactome_Id_to_Uniprot_Id2('Protein7',UniprotId) works in both directions.

reactome_string_type(+Reactome_string:reactome_string, -ReactomeType:reactome_type) is semidet



Reactome_String is a string (atom) of the form: http://www.biopax.org/release/biopax-level3.owl#BiochemicalReaction

component type(+Component:reactome id, -Type:Reactome type) is semidet



This is works in one direction only and returns protein sets as proteins.

See also

- component type slow/2 and type c/2

component_type_slow(?Component:reactome_id, ?Type:reactome_type) is semidet



This works in both directions but can be slow. It returns protein sets as proteins.

See also

- component type/2 and type c/2

type_c(+Component:reactome_id, -Type:Reactome_type) is **semidet**



This three place predicate definition is simple checking of proteins, protein_sets and complexes, They work in one direction. The order of these three predicates is important for speed. The _c means calculated. Rdf files that have been preproceed will have a type/2 predicate to improve speed. Unlike component_type/2 and component_type/2 and component_type_slow/2 these return the correct type for protein sets

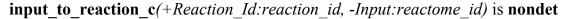
To be done

- add defs for small molecules, dna, rna and physical entities maybe at the momnent the parent_child and

all reactions(-Reactions:list) is det

sactions (itemetrons.vist) is acc

finds all the *Reactions* as a list in the ontology





_c is for calculated, some ontologies should be preprocessed for speed. On back tracking finds all inputs to a given reaction. Fails if reaction has no input or if reaction does not exist.

To be done

- What to do with 'Physical entities'? Should this be changed to only return, proteins, complexs and protein sets?

inputs_to_reaction_list(+Reaction_Id:reaction_id, -Inputs:list) is semidet



uses input_to_reaction/2 so is not very fast

output_to_reaction_c(+Reaction Id:reaction id, -Output Id:reactome id) is nondet



_c is for calculated, some ontologies should be preprocessed for speed. On back tracking finds all outputs for a given reaction. Fails if reaction has no output or if reaction does not exist @tbd What to do with 'Physical entities'? Should this be changed to only return, proteins, complexs and protein_sets?

outputs_to_reaction_list(+Reaction_Id:reaction_id, -Outputs:list) is semidet
 uses output to reaction c so is not very fast



component of complex(+Complex Id:complex id, -Component Id:reactome id) is **nondet**



Back tracks to find all components of a complex



complex_c(+Complex_Id:complex_id, -List_of_Children_Of_Complex:list) is semidet
 c is for calculated the ontology would normally be preprocessed to give a complex/3



component_of_protein_set(+Protein_Set_Id:protein_set_id, -Protein_Id:protein_id) is nondet



Is *Protein_Set_Id* a protein set? If so find members of that set on backtracking.



protein_set_c(+Protein_Set_Id:protein_set_id, -List_Of_Children_Of_Protein_Set:list) is
semidet



c means calcualted. The ontology would normally be preprocessed to provide protein set/3

child_component(+Parent:reactome_id, -Child:reactome_id) is **nondet**



Uses <u>complex/2</u> and <u>protein_set/2</u> which would be asserted on preprocessing, otherwise you would need <u>complex_c/2</u> and <u>protein_set_c/2</u> uses child and parent as a metaphor for membership of a protein set or componet of a protein complex

descendant component(+Parent:reactome id, -Child:reactome id) is **nondet**



On backtracking finds all descendants of a parent component. Uses parent child and descendant as a metaphor for the relations of set membership of a protein set or component of a protein complex. Uses type/2 for speed which should be preprocessed otherwise see type/2

descendant complex or set(+Parent:reactome id, -Descendant:reactome id) is **semidet**



Finds on backtracking all descendents of a component that will also have descendants as they are themselves complexes or protein sets.

child type protein(+Parent:reactome id, +Child:protein id) is **semidet**



Is it true that *Child* is a simple protein not a protien complex or set and is a *Child* of *Parent*. Uses $\underline{\text{type}/2}$ which would be preprocessed otherwise use $\underline{\text{type}}$ $\underline{\text{c}/2}$

all_children_proteins(+Parent:reactome id) is semidet



Is it true that a component such as a complex or protein set has 'children' all of which are proteins. i.e none of its children are also protein sets or complexes. $\underline{\text{type}/2}$ would be preprocessed otherwise see $\underline{\text{type}}$ c/2

See also

- <u>type c/2</u>