Pathway Tools Programming

• Pathway Tools based on Lisp
  • Lisp queries
• Interfaces exist for other languages
  • Perl (PerlCyc)
  • Java (JavaCyc)
  • Python
Lisp queries

- Start Pathway Tools with `-lisp` option
- Type commands at the prompt
- Possible to load scripts from files
Basic Lisp

• Programs and data are lists
• Lists are represented in parenthesis
  • ( 0 1 2 3 4 5 )
• Lisp uses prefix notation
  • ( + 1 2 )
Generic Frame Protocol

• What is a frame?

FRAME: At1g01010

TYPE: Gene
COMMON-NAME: NAC
MEMBER-OF: ...
DESCRIPTION: DB-LINKS:

LINKS TO OTHER FRAMES

SLOTS

SLOT VALUES

FRAME
Pathway Tools functions

Basic GFP functions

- http://pathwaytools.org/gfp.html

Pathway Tools functions

- http://www.pathwaytools.net/ptools-fns.html
Commonly Used GFP Functions

GFP (Generic Frame Protocol) is the API used to access Pathway Tools databases that are stored in the Ocelot object-oriented database system. Many additional GFP operations are described in the full GFP specification.

Note that for all functions we omit the kb argument that allows the programmer to operate on a KB other than the currently selected KB. Function select-organism can be used to change the currently selected KB to the KB for a different organism.

Retrieval Functions

Functions described in this section retrieve data from a KB, but do not modify a KB.

<table>
<thead>
<tr>
<th>Function:</th>
<th>get-slot-values(frame slot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns:</td>
<td>A list of all values of Slot of Frame.</td>
</tr>
<tr>
<td>Arguments:</td>
<td>Frame: A frame id or object. Slot: A slot name (symbol).</td>
</tr>
<tr>
<td>Example:</td>
<td>(get-slot-values 'trp 'synonyms)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function:</th>
<th>get-slot-value(frame slot)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns:</td>
<td>The first value of Slot of Frame.</td>
</tr>
<tr>
<td>Arguments:</td>
<td>Frame: A frame id or object. Slot: A slot name (symbol).</td>
</tr>
<tr>
<td>Example:</td>
<td>(get-slot-value 'trp 'common-name)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function:</th>
<th>get-class-all-instances(class)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns:</td>
<td>A list of all frames that are direct or indirect instances of Class.</td>
</tr>
<tr>
<td>Arguments:</td>
<td>Class: A class frame.</td>
</tr>
<tr>
<td>Example:</td>
<td>(get-class-all-instances '</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function:</th>
<th>instance-all-instance-of-p(instance class)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns:</td>
<td>T if Instance is a direct or an indirect child of Class.</td>
</tr>
<tr>
<td>Arguments:</td>
<td>Class: A class frame.</td>
</tr>
<tr>
<td>Instance:</td>
<td>An instance frame.</td>
</tr>
<tr>
<td>Example:</td>
<td>(instance-all-instance-of-p 'P238 : Pumps)</td>
</tr>
</tbody>
</table>
### Reaction Operations

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| all-rxns (optional (type :smm))              | Function: all-rxns (optional (type :smm))  
  Returns: All in the current organism. The Type arg determines exactly which reactions are returned.  
  Arguments: Type can be one of: :all = All reactions. :smm = All reactions whose substrates are all small molecules, plus all reactions that are members of pathways of small-molecule metabolism. Note that :smm will often return more reactions than does :small-molecule because some pathways of small-molecule metabolism contain reactions involving macromolecules, e.g., ACP. :small-molecule = All reactions whose substrates are all small molecules, as opposed to macromolecules. :enzyme = All enzyme-catalyzed reactions (instances of classes EC-Reactions or Unclassified-Reactions). :transport = All transport reactions. :dna = All DNA Binding Reactions. |
| genes-of-reaction (rxn)                      | Function: genes-of-reaction (rxn)  
  Returns: All genes that code for enzymes (or subunits of enzyme complexes) that catalyze the reaction Rxn. If multiple enzymes catalyze Rxn, genes encoding all of the enzymes are returned.  
  Arguments: Rxn: A reaction frame. |
| substrates-of-reaction (rxn)                 | Function: substrates-of-reaction (rxn)  
  Returns: All substrates of the reaction Rxn, meaning the union of the reactants and products of Rxn.  
  Arguments: Rxn: A reaction frame. |
| enzymes-of-reaction (rxn)                    | Function: enzymes-of-reaction (rxn)  
  Returns: All enzymes that catalyze the reaction Rxn.  
  Arguments: Rxn: A reaction frame. |
| reaction-reactants-and-products (rxn &key pwy direction) | Function: reaction-reactants-and-products (rxn &key pwy direction)  
  Returns: the reactants of Rxn, and the products of Rxn. The reactants and products are those determined according to either the direction of Rxn in Pwy, or the direction specified by the Direction arg (which should be either :L2R or :R2L). In other words, either the Pwy or the Direction arg should be specified, but not both.  
  Arguments: Rxn: A reaction frame.  
  Pwy: A pathway frame.  
  Direction: :L2R (left to right) or :R2L (right to left) |
| reaction-type? (rxn type)                    | Function: reaction-type? (rxn type)  
  Returns: |
|"
(select-organism :org-id 'LYCO)
(setq pwys (all-pathways))
(setq first (car pwys))
(get-slot-value first 'COMMON-NAME)
PerlCyc limitations

• Only one PerlCyc script can be running at a time (socket connection)
• Works only on UNIX
• Optional parameters to functions are not supported
• Certain functions may not be available
• It's just a thin layer - frames are not implemented in Perl
Installing and running PerlCyc

• Requirements:
  – UNIX installation of Pathway Tools
    • does not work on Windows
  – Perl 5.6 or later

• Installation
  – Download PerlCyc
    • http://github.com/solgenomics/perlcyc
  – Follow installation instructions
  – Start pathway-tools using the `-api` option
  – Write and run PerlCyc perl scripts
Perl interface to Pathway Tools — Read more
http://solgenomics.net

comment on deprecation added.

lukasmueller authored a year ago

- bin 3 years ago added a couple of dumping scripts [rbuels]
- lib  a year ago comment on deprecation added. [lukasmueller]
- t  4 years ago added check and skip for pathwaytools presence in tests [rbuels]
- .gitignore  3 years ago Update MANIFEST, MANIFEST.SKIP and add a .gitignore [loto]
- Build.PL  2 years ago Perl dependency should be 5.10.0 instead of 5.10 [loto]
- Changes  4 years ago added an INSTALL file with installation instructions [rbuels]
- INSTALL  4 years ago added makefile generation step in INSTALL [rbuels]
- MANIFEST  3 years ago Update MANIFEST, MANIFEST.SKIP and add a .gitignore [leto]
- MANIFEST.SKIP  3 years ago Update MANIFEST, MANIFEST.SKIP and add a .gitignore [leto]
PerlCyc API

• Implements
  – Generic Frame Protocol (GFP) functions
  – Pathway Tools functions

• Function name conventions:
  – Replace dashes with underlines
  – Replace question marks with ‘_p’
Data type equivalents

• Note: Perl is not strongly typed!
  – Lisp list = Perl array
  – Lisp string = Perl scalar
  – Lisp symbol = Perl scalar
  – Lisp boolean ("NIL" | "t") = Perl scalar (NIL|t)
  – Lisp integer, etc = Perl scalar
Getting information

- all_pathways
- all_rxns
- all_genes
Frame functions

- `add_slot_value`
- `get_slot_value`
- `get_slot_values`
- `put_slot_value`
- `replace_slot_values`
- `remove_slot_value`
- `get_frame_slot`
Using the PerlCyc object

use strict;
use perlcyc;

my $c = perlcyc->new("ARA");

;;;; Lisp equivalent
;; (select-organism :org-id 'ARA)
Using the PerlCyc object

use strict;
use perlcyc;

my $c = perlcyc->new("ARA");

my @pathways = $c->all_pathways();

foreach my $p (@pathways) {
    my $name = $c->get_slot_value($p, "COMMON-NAME");
}
Using the PerlCyc object

use strict;
use perlcyc;

my $c = perlcyc->new("ECO");

my @pathways = $c->all_pathways();
my $desc = "blabla...";
foreach my $p (@pathways) {
    my $name = $c->get_slot_value($p, "COMMON-NAME");
    if ($name =~ /glycolysis/i) {
        $c->put_slot_value($name,"DESCRIPTION",$desc);
    }
}
$c->save_kb();
Version 2.0

- Version 2.0 will be available from CPAN
- Supports optional parameters

Future
- Support for multiple connections planned
- http://github.com/solgenomics/perlcyc
- http://github.com/solgenomics/javacyc