Search and Rescue: logic and visualisation of biochemical networks

Nicos Angelopoulos[1], Paul Shannon[2] and Lodewyk Wessels[1]

n.angelopoulos@nki.nl

[1] Netherlands Cancer Institute, Amsterdam, Netherlands
[2] Fred Hutchinson Cancer Research Center, Seattle, USA

overview

- position paper
- demonstrates

  reactions as relations and search

  strengths of R interface

logic programming for biology

relations based knowledge representation
selection as search
database integration
interactive operation
scripting

but,...

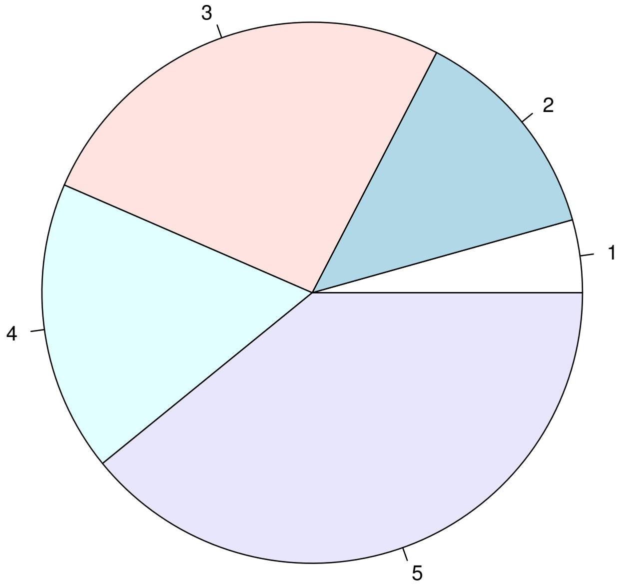
no visualisation
no statistics
no user-contributed code culture

r..eal

Interface to the R statistical software,

- visualisation
- statistics
- tons of user-contributed code

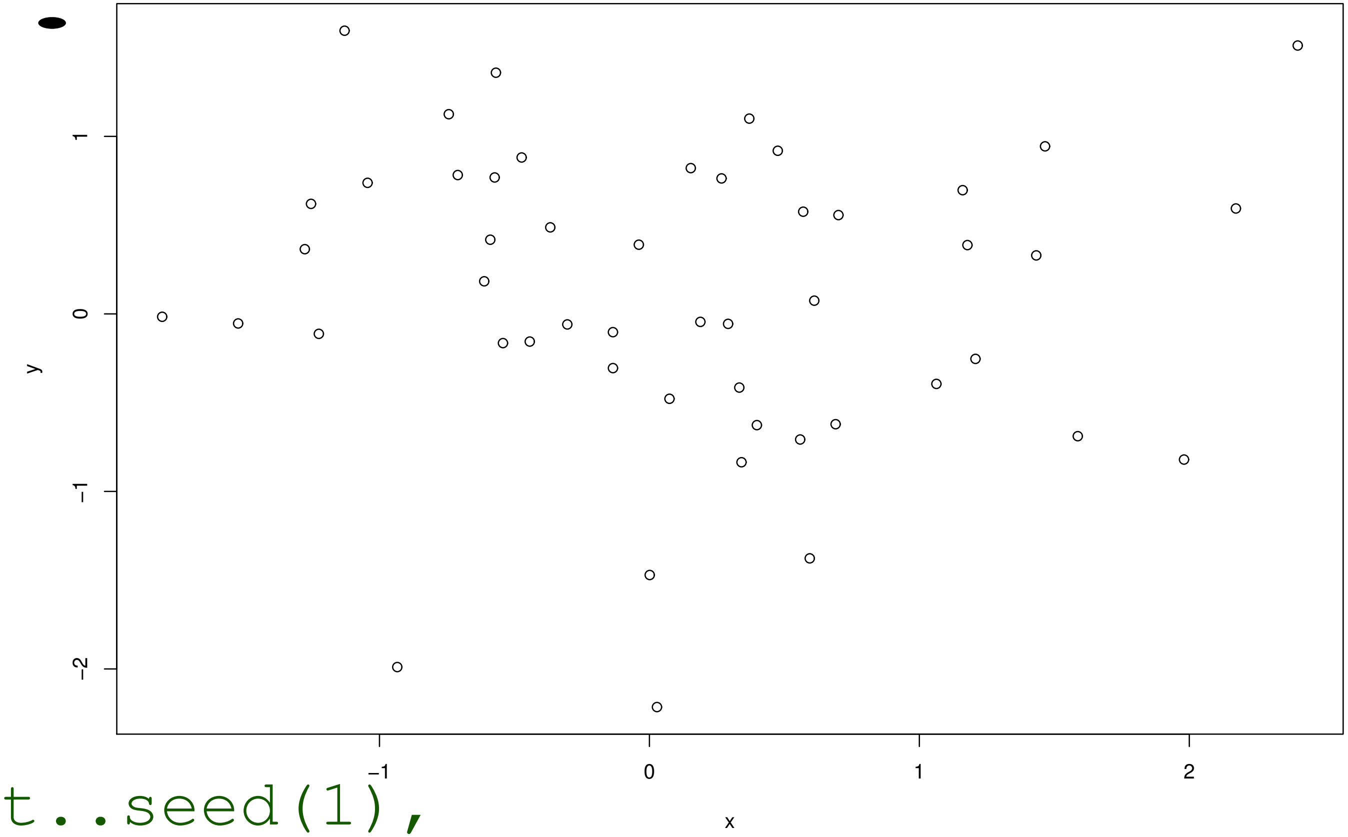
r..eal example



```
ensure_loaded( library(real) ).
cars <- [1,3,6,4,9].
<- plot( cars ).

<- plot( [1,3,6,4,9] ) .
```

r..eal example II



```
    <- set..seed(1),
    y <- rnorm(50), x <- rnorm(y),
    <- x11(width=5,height=3.5)
    <- plot(x,y),
    X <- x.
X = [0.39810588036706807, -0.6120263932507712,
```

sources of biological knowledge

deluge of data generated due to high throughput technologies
PPI protein-protein interactions

STRING

$5,214,234$ proteins

$224,346,017$ interactions

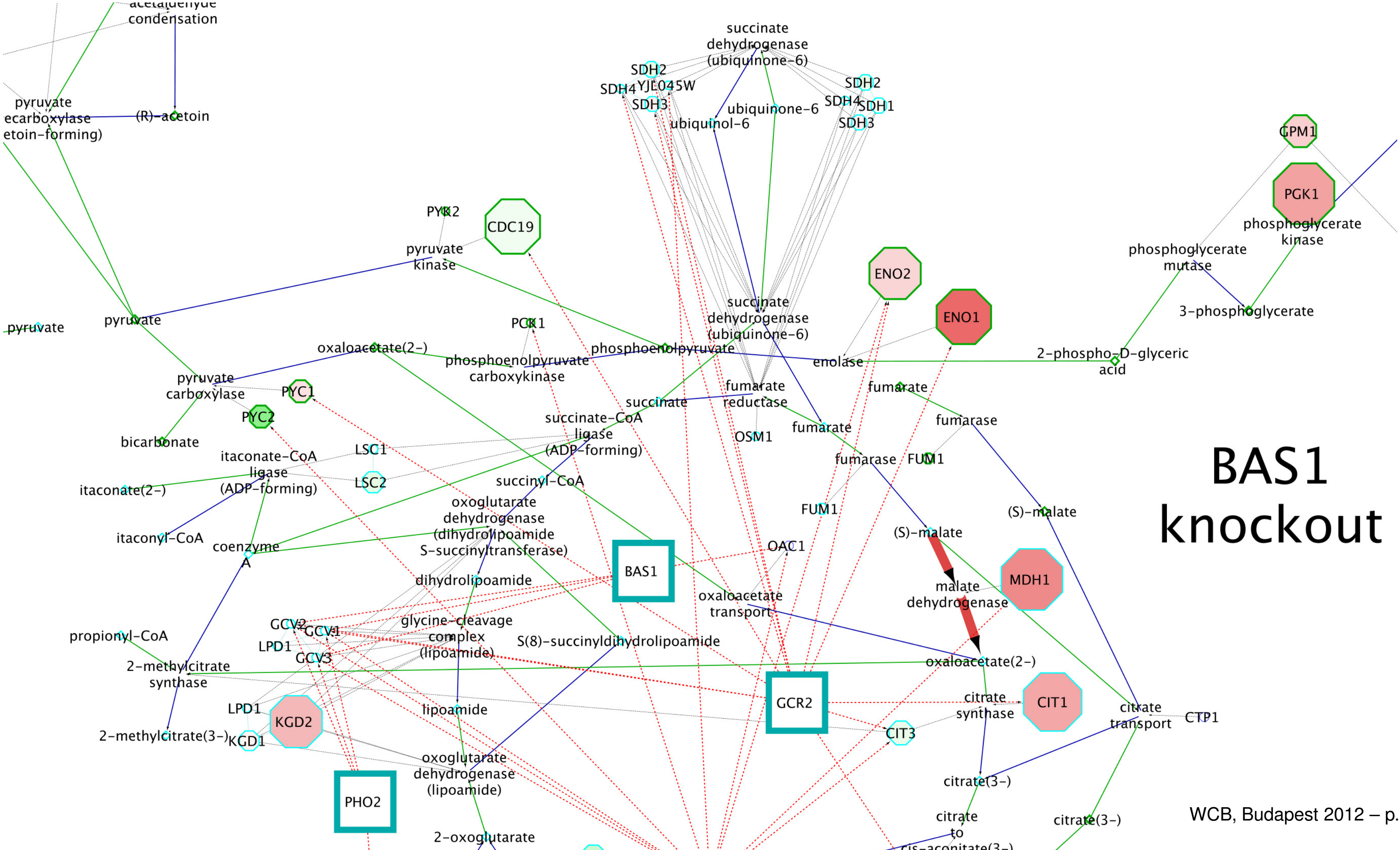
1133 organism

HPRD

$39,194$ interactions

homo sapiens

metabolic TFs in yeast



BAS1
knockout

representation

```
interaction( From, To, Types, References ).

activation( From, To, Organism, Pathway ).
inhibition( From, To, Organism, Pathway ).
phosphorylation( From, To, Organism, Pathway )
ubiquination( From, To, Organism, Pathway ).
```

Dijkstra’s algo w/ max depth and targets

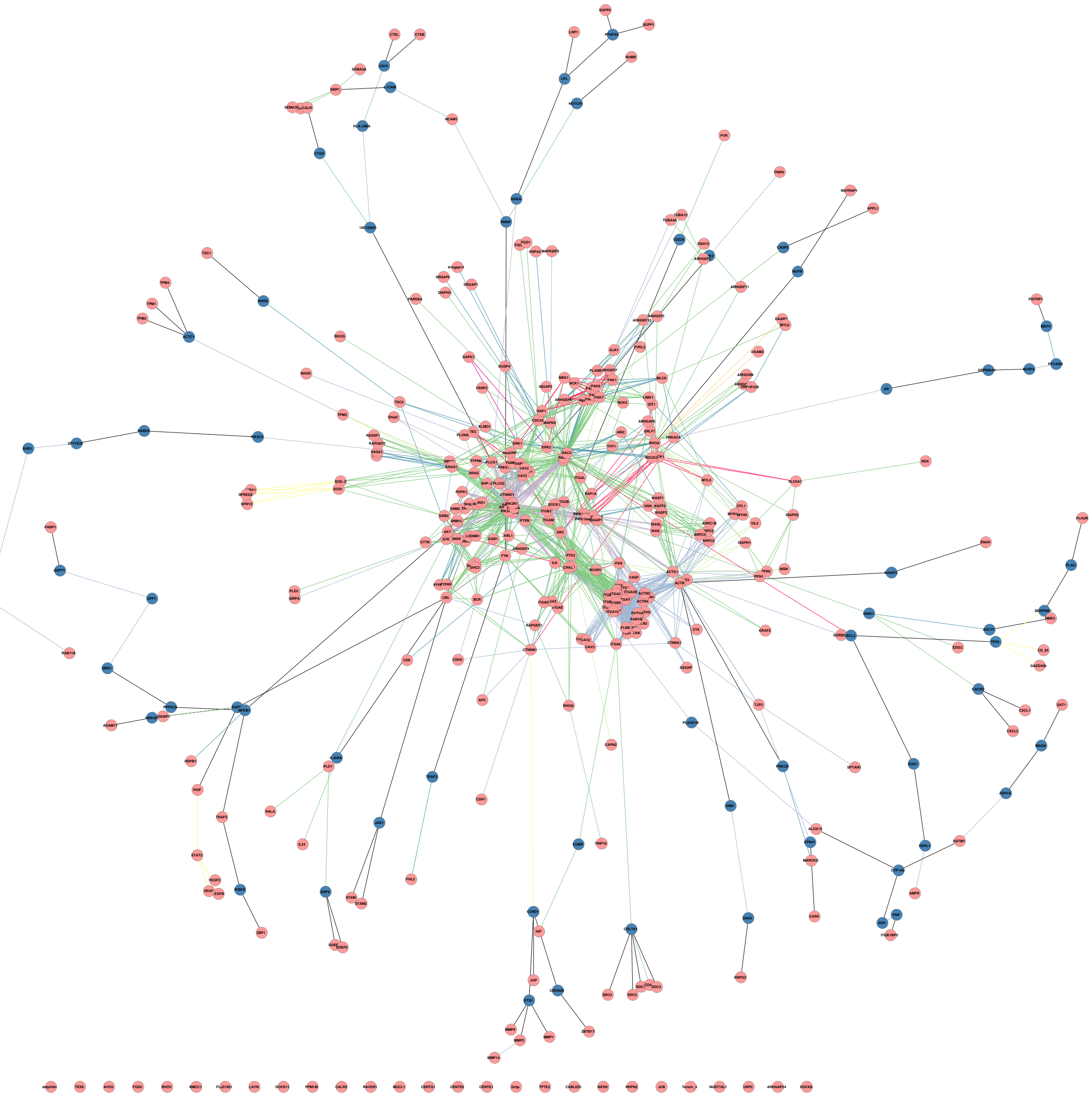
```
connect( Sel, D, Paths ) :-
     findall( path(0,S,[]), member(S,Sel), Emptys ),
     list_to_ord_set( Sel, OrdSel ),
     connect_paths( Emptys, D, OrdSel, Paths, [] ).

connect_paths( [], _D, _Sel, Paths, Paths ).
connect_paths( [path(Ds,S,Route)|T], D, Sel, Paths, TP ) :-
     findall( X, edge(S,X), Xs ),
     Ds1 is Ds + 1,
     add_connecting_edges( Xs, Ds1, S, Route, T, D, Sel, Rem, Paths,
     connect_paths( Rem, D, Sel, ContPaths, TP ).
```

part II

```
add_connecting_edges( [], _Ds, _S, _Route, Rem, _D, _Sel, Rem, Paths,
add_connecting_edges( [X|Xs], Ds, S, Route, T, D, Sel, Rem, Paths, TP
     ( memberchk(X,Route) ->
          TRem = Rem,
          MidPaths = Paths
          ;
          ( ord_memberchk(X,Sel) ->
               Rem = TRem,
               Paths = [[X,S|Route]|MidPaths]
               ;
               MidPaths = Paths,
               ( Ds > D ->
                    Rem = TRem
                    ;
                    Rem = [path(Ds,X,[S|Route])|TRem]
               )
          )
     ),
     add_connecting_edges( Xs, Ds, S, Route, T, D, Sel, TRem, MidPath
```

adhesome library



(R)Bioconductor

Bioconductor graphs visualisation software.

RBioconductor R bi-directional interface to Bioconductor.

rcy r..eal based routines for displaying Prolog graphs in Bioconductor

r..eal availability

SWI and Yap from sources.
Yap windows binary

`http://bioinformatics.nki.nl/~nicos/sware/real`

also on
`git://www.swi-prolog.org/home/`
`pl/git/packages/real.git`

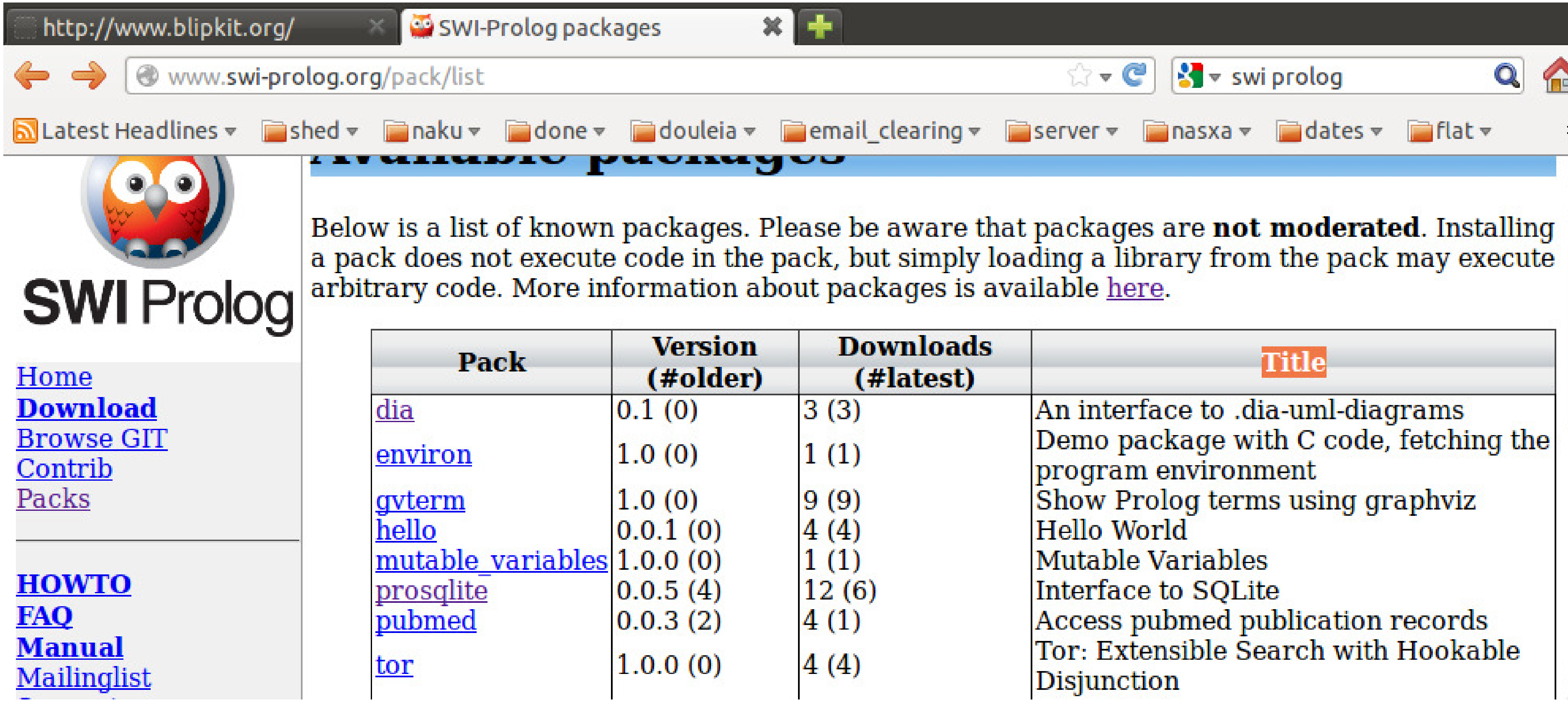
piece-meal prolog bioinformatics

r..eal	Swi/Yap <-> R interface
pubmed	access pumed citation records
prosqlite	Swi/Yap <-> SQLite interface
rcy	graph visualisation
depth search	depth limited reachability

versus the more holistic

blip : `http://www.blipkit.org/`

SWI packs



Below is a list of known packages. Please be aware that packages are **not moderated**. Installing a pack does not execute code in the pack, but simply loading a library from the pack may execute arbitrary code. More information about packages is available here.

Pack	**Version (#older)**	**Downloads (#latest)**	**Title**
dia	0.1 (0)	3 (3)	An interface to .dia-uml-diagrams
environ	1.0 (0)	1 (1)	Demo package with C code, fetching the program environment
gvterm	1.0 (0)	9 (9)	Show Prolog terms using graphviz
hello	0.0.1 (0)	4 (4)	Hello World
mutable_variables	1.0.0 (0)	1 (1)	Mutable Variables
prosqlite	0.0.5 (4)	12 (6)	Interface to SQLite
pubmed	0.0.3 (2)	4 (1)	Access pubmed publication records
tor	1.0.0 (0)	4 (4)	Tor: Extensible Search with Hookable Disjunction

bottom-line

relations provide an intuitive interface to biological data which the lp engine can powerfully exploit

Future work.

- publish rcy and the search routine
- access large datasets via prosqlite