r..eal : Integrative statistics with R

Nicos Angelopoulos, Vítor Santos Costa, João Azevedo, Jan Wielemaker, Rui Camacho and Lodewyk Wessels

n.angelopoulos@nki.nl

Netherlands Cancer Institute, Amsterdam, Netherlands

overview

- availability
  design philosophy
  syntax
  likely application areas
- basic examples
  page-ranking Aleph
  search and visualisation of biochemical networks

what is r..eal

interface to the R statistical software

- library that integrates the *R* language to LP systems
- language for incoporating functional statistics in LP

what is R

+ statistical programming language (*S*)
  functional style
  strong presence in niche areas
  user contributed packages
  single implementation
  major satellite projects (Bioconductor)
- unclear semantics
  conglomerate of features

availability

SWI-Prolog

```
?- pack_install(real).
```

% binaries for

i368-win32, x64-win64, i386-linux and x86_64-linux

source compilation via package manager and manually

Yap

included in binary distributions

latest source can be dropped in development sources

design

minimality
: interactions through a small number of predicates

R flavour
: it should feel as if we are writing R code

Prolog flavour
: based on Prolog terms

access predicates

R uses <- as one of its 2 assignment operators

r..eal defines predicates <-/1 and <- 2

also as operators

```
op(950,fx,<-)
op(950,yfx,<-)
```

r..eal interactions

Prolog is the top-level

start *R* as a shared OS library (.so, .dll, ...)

C-interface to pass data-structures between *R* and Prolog

operations

- copy data across
- apply *R* functions
- pass results back to Prolog

simple example

pass some Prolog data to an *R* variable (x)

```
?- x <- [1,2,3].
true.
```

pass the result of an *R* function to a Prolog variable (Y)

```
?- Y <- mean(x)
Y = 2.0.
```

call modes

```
+Rvar    <-  +PLvalue            x <- [1,2,3]
-PLvar   <-  +Rvar                 X <- x
-PLvar   <-  +Rexpr              X <- mean(x)
+RExpr1  <-  +Rexpr1   length(y) <- mean(x)
```

PLvar		unbound variable
	PLvalue	(RHS) atomic or list
Rexpr	Rexpr	non-list term structure
Rvar		atomic
	Rvar	atomic known to R

data traffic

Prolog		R
integer	<->	integer
float	<->	double
atom	<->	char
char	->	char
true/false	<->	logical

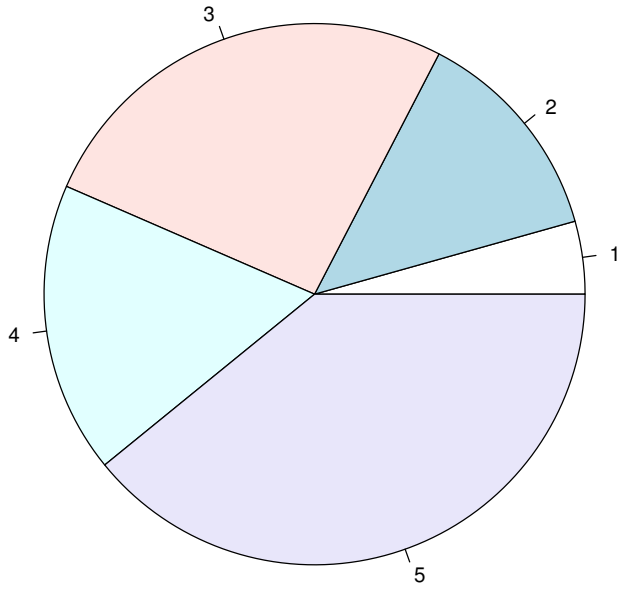
R expressions syntax

```
as..integer(c(1,2)) => as.integer(c(1,2))
devoff(.)  => dev.off()

a^[2] => a[2]
a^[*,*,2] => a[„2]

a$val => a$val
a@val => a@val

source(+"String") => source("String")
source(+'Atom') => source("Atom")
'Expr', -'Expr', -"Expr" => Expr
```

hidden variables

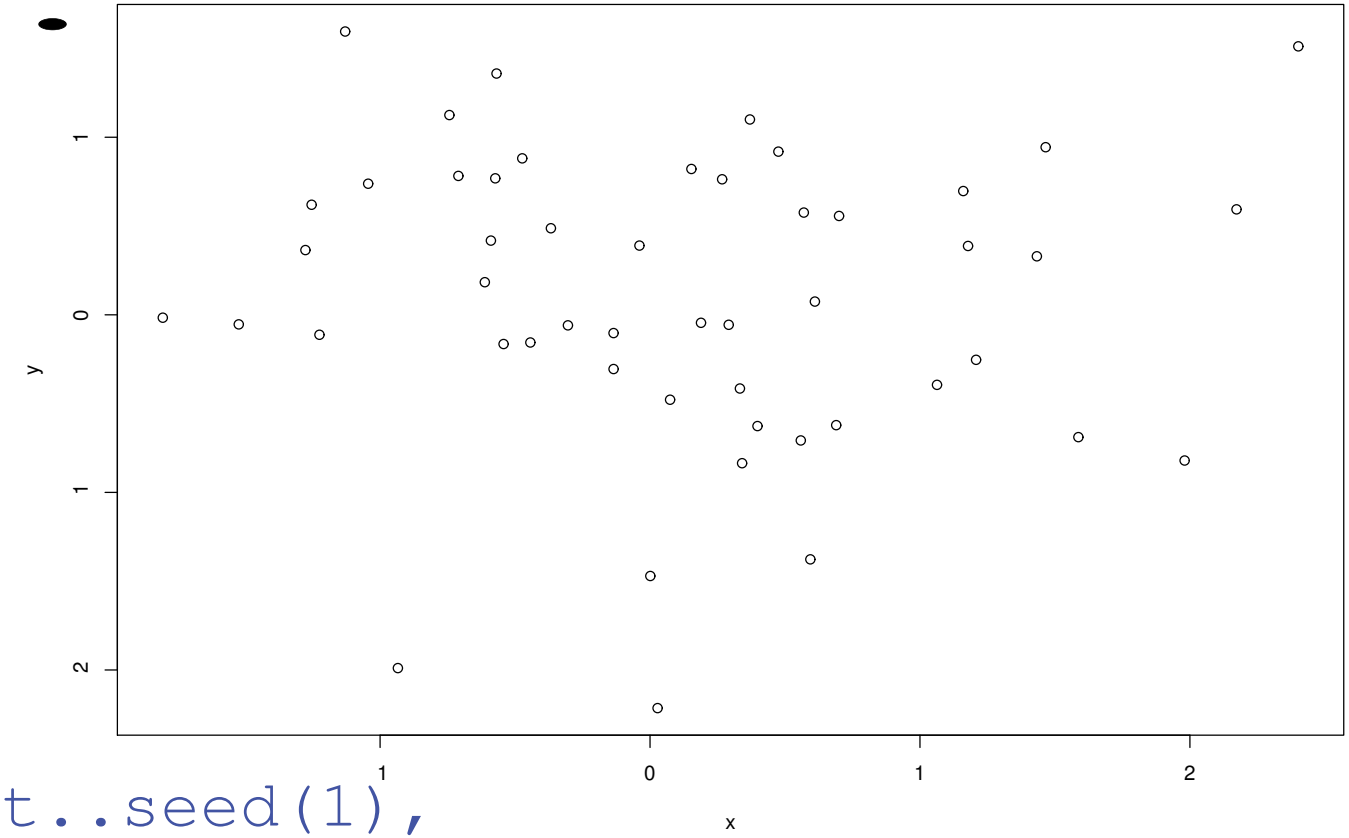
```
?- findall(I, between(1,50000,I), Is),
time( A <- mean(Is) ).

% 181 inferences,0.002 CPU in 0.002 seconds
(100% CPU,75597 Lips)
Is = [1, 2, 3, 4, 5, 6, 7, 8, 9|...],
A = 25000.5.
```

r..eal example




```
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,
-0.61202639325077 12,
```

logic programming for biology

relational knowledge representation
logical inference
database integration
interactive operation
scripting
selection as search

R

statistics
visualisation

user-contributed code culture

sources of biological knowledge

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

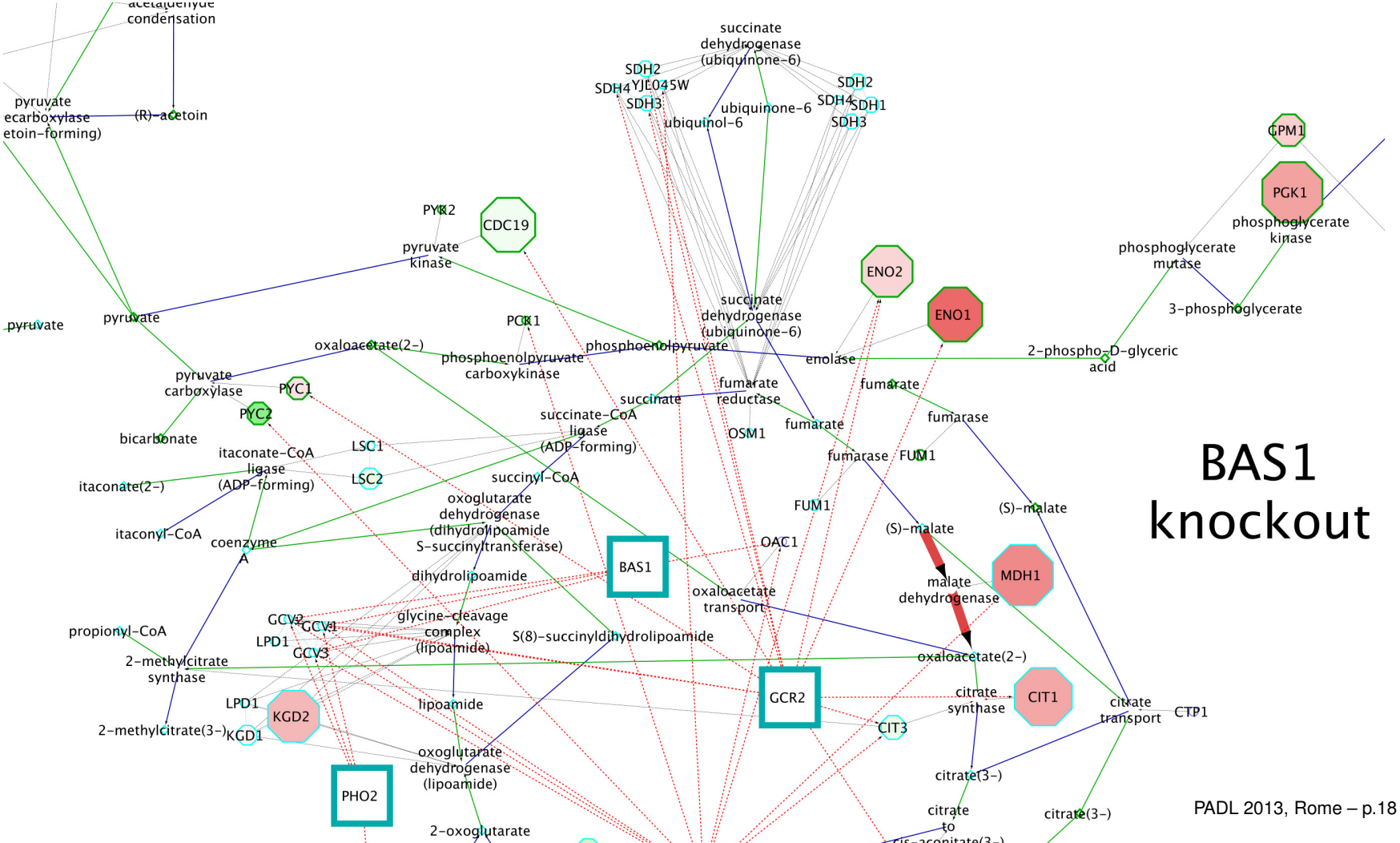
STRING

- $5,214,234$ proteins
- $224,346,017$ interactions
- 1133 organism

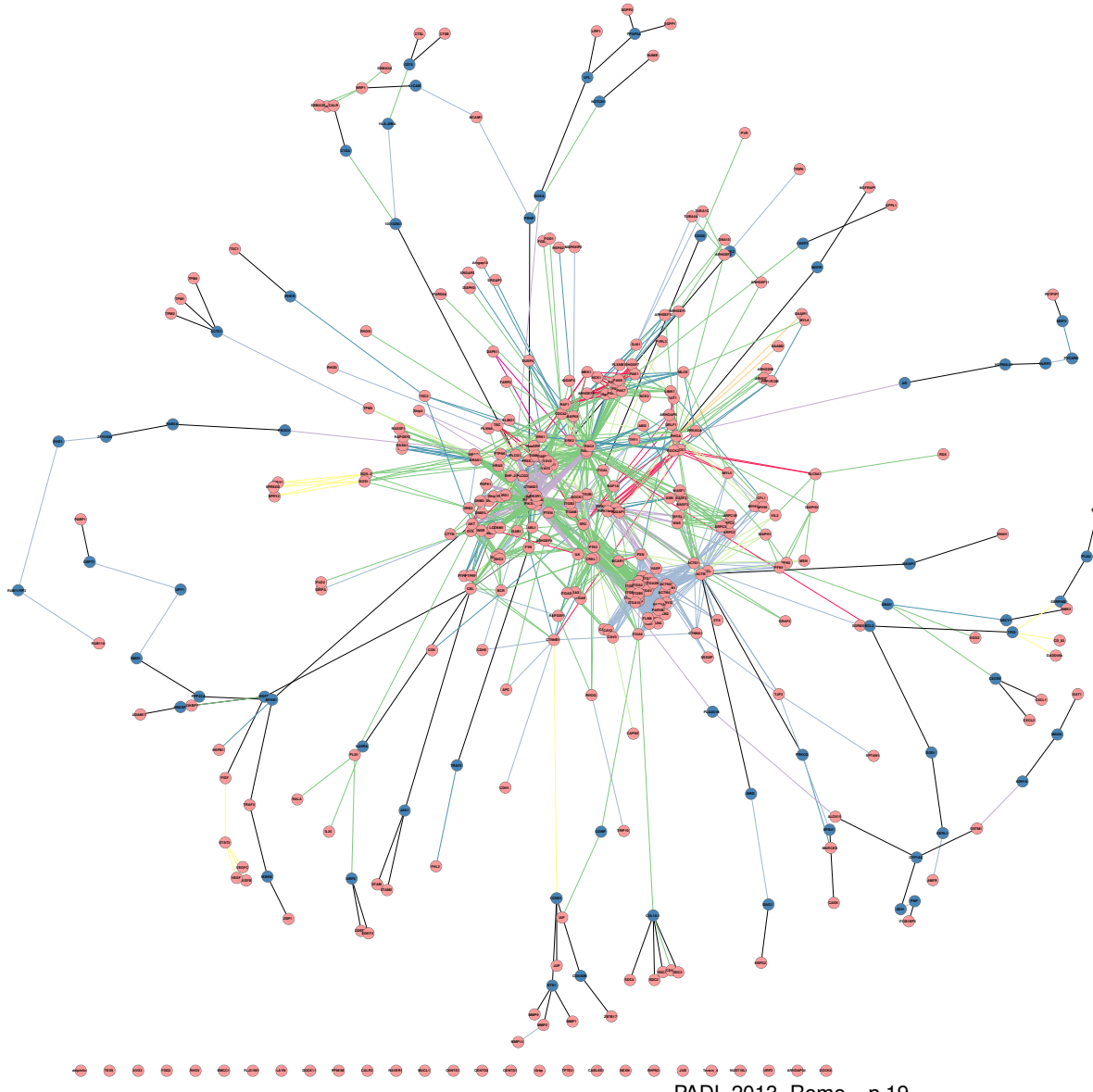
HPRD

- $39,194$ interactions
- homo sapiens

metabolic TFs in yeast



adhesome library



(R)Cytoscape

Cytoscape graphs visualisation software.

RCytoscape *R* bi-directional interface to Cytoscape.

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

r..eal information

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

also on

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

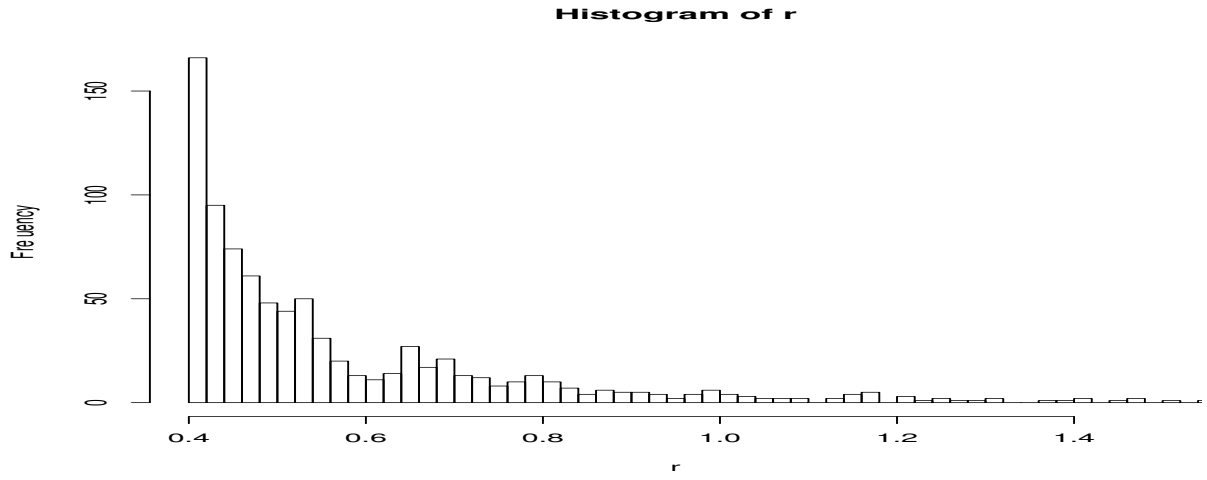
piece-meal prolog bioinformatics

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/`

Aleph




```
pagerank(File,nav(Name,Arity,Value)) :-
      parse(File,Graph),
      g <- graph(Graph),
      r <- page..rank(g),
      Scores <- r$vector,
      max_element(Scores, Name, Arity, Value).
```

bottom-line

r..eal is an

- intuitive
- efficient
- tight

integration to *R*

Future work.

- thread-wrapper
- port to more Prologs
- more applications