R software for network analysis

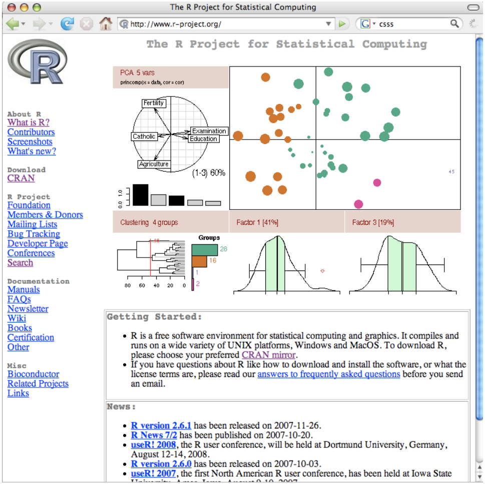
Dave Hunter

Penn State Dept. of Statistics
Joint with Mark and Carter and many others

MURI networks grant meeting, November 18, 2008

The R project and statnet

www.r-project.org



The R Project for Statistical Computing

About R
What is R?
Contributors
Screenshots
What's new?

Download
CRAN

R Project
Foundation
Members & Donors
Mailing Lists
Bug Tracking
Developer Page
Conferences
Search

Documentation
Manuals
FAQs
Newsletter
Wiki
Books
Certification
Other

Misc
Bioconductor
Related Projects
Links

Getting Started:

- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To download R, please choose your preferred CRAN mirror.
- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News:

- R version 2.6.1 has been released on 2007-11-26.
- R News 7/2 has been published on 2007-10-20.
- useR! 2008, the R user conference, will be held at Dortmund University, Germany, August 12-14, 2008.
- R version 2.6.0 has been released on 2007-10-03.
- useR! 2007, the first North American R user conference, has been held at Iowa State

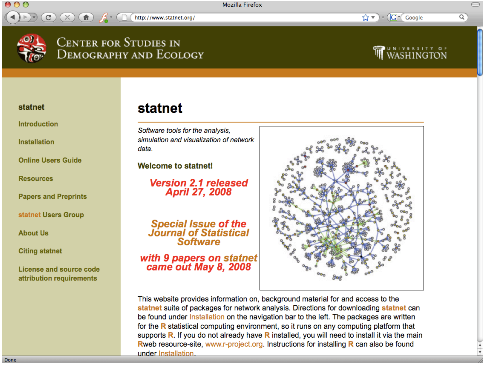
R: A statistical environment

- Open-source
- Extendible; ∼1500 user-created packages exist
- Mostly written in C
- Well-maintained (if not always well-documented) by a core group of the world's top computational statisticians

statnet

`www.statnet.org`

Thanks Carter!



statnet

Introduction

Installation

Online Users Guide

Resources

Papers and Preprints

statnet Users Group

About Us

Citing statnet

License and source code attribution requirements

statnet

Software tools for the analysis, simulation and visualization of network data.

Welcome to statnet!

Version 2.1 released April 27, 2008

Special Issue of the Journal of Statistical Software

with 9 papers on statnet came out May 8, 2008

This website provides information on, background material for and access to the statnet suite of packages for network analysis. Directions for downloading statnet can be found under Installation on the navigation bar to the left. The packages are written for the R statistical computing environment, so it runs on any computing platform that supports R. If you do not already have R installed, you will need to install it via the main Rweb resource-site, www.r-project.org. Instructions for installing R can also be found under Installation.

Done

statnet: A suite of R packages for network analysis

Includes packages:

- sna
- network
- ergm
- latentnet
- degreenet
- . . .

Special issue (v 24) of *J. Stat. Soft.* devoted to statnet

A simple R package exploiting the R/C interface

```
% ls -RF1 pakij/
DESCRIPTION*
R/
src/

pakij//R:
my_runif.R
zzz.R*

pakij//src:
my_runif.c*
```

```
% R CMD INSTALL pakij
```

A simple R package exploiting the R/C interface

```
% ls -RF1 pakij/
DESCRIPTION*
R/
src/

pakij//R:
my_runif.R
zzz.R*

pakij//src:
my_runif.c*


% R CMD INSTALL pakij
```

pakij/R/my_runif.R

```
my_runif <- function(n) {
  .C("hey_diddle_diddle",
     n=as.integer(n),
     foo=double(n)
     )$foo
}
```

A simple R package exploiting the R/C interface

```
% ls -RF1 pakij/
DESCRIPTION*
R/
src/

pakij//R:
my_runif.R
zzz.R*

pakij//src:
my_runif.c*


% R CMD INSTALL pakij
```

pakij/R/my_runif.R

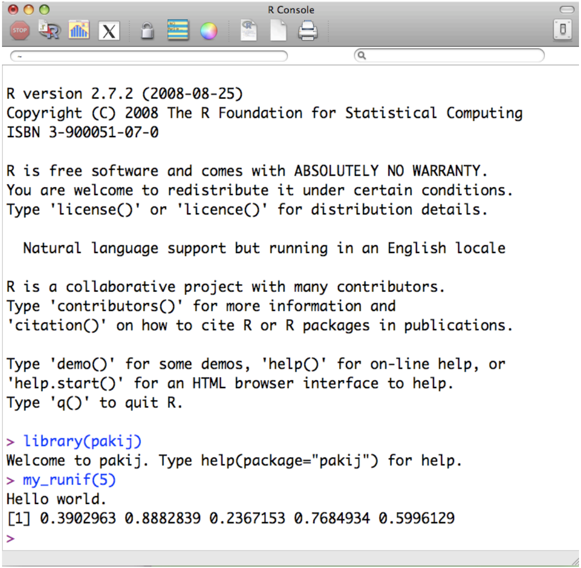
```
my_runif <- function(n) {
  .C("hey_diddle_diddle",
     n=as.integer(n),
     foo=double(n)
     )$foo
}
```

pakij/src/my_runif.c

```
#include <R.h>

void hey_diddle_diddle (int *n, double *answer) {
  Rprintf("Hello world.\n");
  GetRNGstate();  /* R: enable uniform RNG */
  for (int i=0; i<*n; i++)
    answer[i] = unif_rand();
  PutRNGstate();  /* Disable RNG before return */
}
```

Using the pakij package



```
R version 2.7.2 (2008-08-25)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(pakij)
Welcome to pakij. Type help(package="pakij") for help.
> my_runif(5)
Hello world.
[1] 0.3902963 0.8882839 0.2367153 0.7684934 0.5996129
>
```

The R API

- Entry points in the R executable callable from C code
- Example: Carter exploits this in the `network` package.

In network/R:

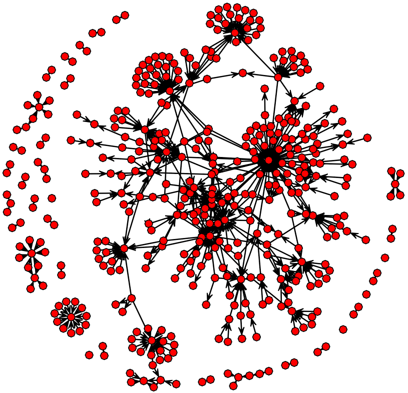
```
add.edge<-function(x, tail, head, names.eval=NULL,
                   vals.eval=NULL, edge.check=FALSE, ...){
  #Check to be sure we were called with a network
  if(!is.network(x))
    stop("add.edge requires an argument of class network.")
  #Do the deed
  invisible(.Call("addEdge_R",x,tail,head,names.eval,vals.eval,
                  edge.check, PACKAGE="network"))
}
```

In network/src:

```
SEXP addEdge_R(SEXP x, SEXP tail, SEXP head, SEXP namesEval,
               SEXP valsEval, SEXP edgeCheck) {
   ...
```

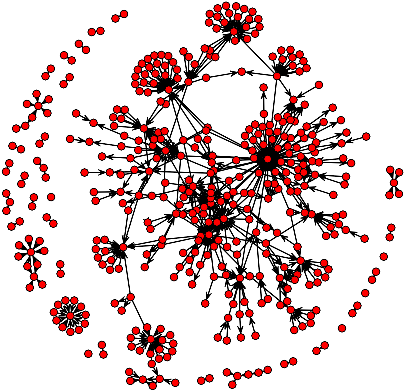
An E. Coli network (Salgado et al., 2001) in R

```
> plot(ecoli1)
```



An E. Coli network (Salgado et al., 2001) in R

```
> plot(ecoli1)
> class(ecoli1)
[1] "network"
```

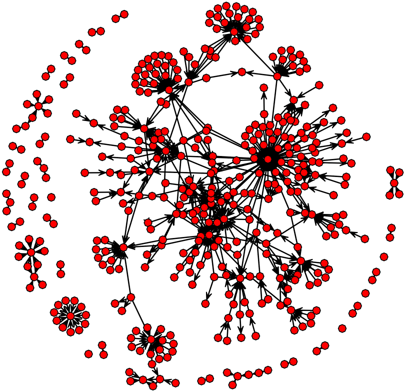


An E. Coli network (Salgado et al., 2001) in R

```
> plot(ecoli1)
> class(ecoli1)
[1] "network"
> summary(ecoli1)
Network attributes:
  vertices: 423
   directed : TRUE
   hyper : FALSE
   loops : FALSE
   multiple : FALSE
   bipartite : FALSE
   total edges= 519
   density = 0.002907465

 Vertex attributes:

self:
FALSE  TRUE
  364    59
```

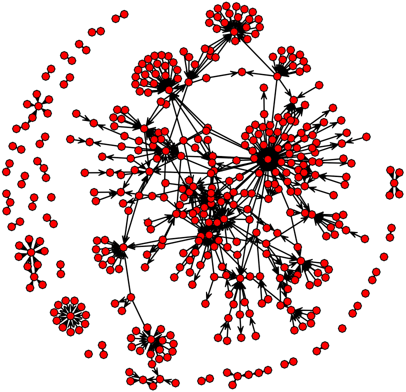


An E. Coli network (Salgado et al., 2001) in R

```
> plot(ecoli1)
> class(ecoli1)
[1] "network"
> summary(ecoli1)
Network attributes:
  vertices: 423
   directed : TRUE
   hyper : FALSE
   loops : FALSE
   multiple : FALSE
   bipartite : FALSE
   total edges= 519
   density = 0.002907465

 Vertex attributes:

self:
FALSE  TRUE
  364    59
> summary(ecoli1 ~ edges + odegree(0:5))
   edges odegree0 odegree1 odegree2 odegree3 odegree4 odegree5
     519       81      229       71       25       13        3
```



A simple ERGM using the ergm package in R

The `ecoli1` network has 519 edges, 423 nodes (thus 423×422 possible directed edges). Take $g(y) = \#$ edges in y:

$$\begin{aligned} P(Y=y) &\propto \exp\{\theta \times \#\text{ edges in } y\} \\ &= \left(\frac{p}{1-p}\right)^{(\#\text{ edges in } y)} \end{aligned}$$

A simple ERGM using the ergm package in R

The `ecoli1` network has 519 edges, 423 nodes (thus 423×422 possible directed edges). Take $g(y) = \#$ edges in y:

$$\begin{aligned} P(Y = y) &\propto \exp\{\theta \times \#\text{ edges in } y\} \\ &= \left(\frac{p}{1-p}\right)^{(\#\text{ edges in } y)} \end{aligned}$$

```
> model1 <- ergm(ecoli1 ~ edges)
> model1$coef
    edges
-5.837562
```

This means the MLE is $\hat{\theta} = -5.84$. It is exact because $g(y)$ in this example ensures independent Y_{ij}, so MPLE=MLE!

A simple ERGM using the ergm package in R

The `ecoli1` network has 519 edges, 423 nodes (thus 423×422 possible directed edges). Take $g(y) = \#$ edges in y:

$$\begin{aligned} P(Y=y) &\propto \exp\{\theta \times \#\text{ edges in } y\} \\ &= \left(\frac{p}{1-p}\right)^{(\#\text{ edges in } y)} \end{aligned}$$

```
> model1 <- ergm(ecoli1 ~ edges)
> model1$coef
    edges
-5.837562
```

This means the MLE is $\hat{\theta} = -5.84$. It is exact because $g(y)$ in this example ensures independent Y_{ij}, so MPLE=MLE!

Observe:

```
> log(519 / (423 * 422 - 519))
[1] -5.837562
```

Another simple ERGM

Let's consider an edge to have one of four categories, depending on whether or not each of its endpoints is self-regulating:

```
> model2 <- ergm(ecoli1 ~ nodemix("self"))
> summary(model2)
Pseudolikelihood Results:
                     Estimate Std. Error MCMC s.e. p-value
mix.self.FALSE.FALSE -6.61974    0.07543        NA  <1e-04
mix.self.TRUE.FALSE  -7.48923    0.28874        NA  <1e-04
mix.self.FALSE.TRUE  -4.22686    0.05730        NA  <1e-04
mix.self.TRUE.TRUE   -5.04049    0.21389        NA  <1e-04
```

This choice of $g(y)$ also ensures independent Y_{ij}, so the MLE and the MPLE are the same. Thus, the MLE is found analytically and there is no MCMC standard error.