gRaphical modelling software in R – status (and future?)

www.r-project.org/gR

Claus Dethlefsen,
Aalborg Hospital, Aarhus University Hospital

Søren Højsgaard
Aarhus University

Steffen Lauritzen
University of Oxford

The gR-initiative - background

- Graphical models in various forms have been around for 25+ years.
- Various pieces of software (commercial and non-commercial) have been developed.

Typical for these programs:

- Independent stand-alone programs.
- Developed and maintained by a small group.
- Code is not open source.
- Each package has its own script language and GUI.
- Packages usually only run on a single platform.

The gR-initiative - history

- The gR-initiative: Integrate graphical model software into general, multi-platform extendable software environment – R.

- Background:A kick-off meeting in Vienna, 2002
- A graphical model session at DSC2003 in Vienna
- A small grant from the Danish Natural Science Research Council.
- Described in ”gRaphical Models in R: A new initiative within the R project”, (S.L. Lauritzen, *R News* **2**(3), 39)

Time to look at how far we have made it (briefly):

- Several gRaphical modelling R-packages (on CRAN)
- A common platform for graphical modelling software (on CRAN)
- The pace has gone down...

Outline

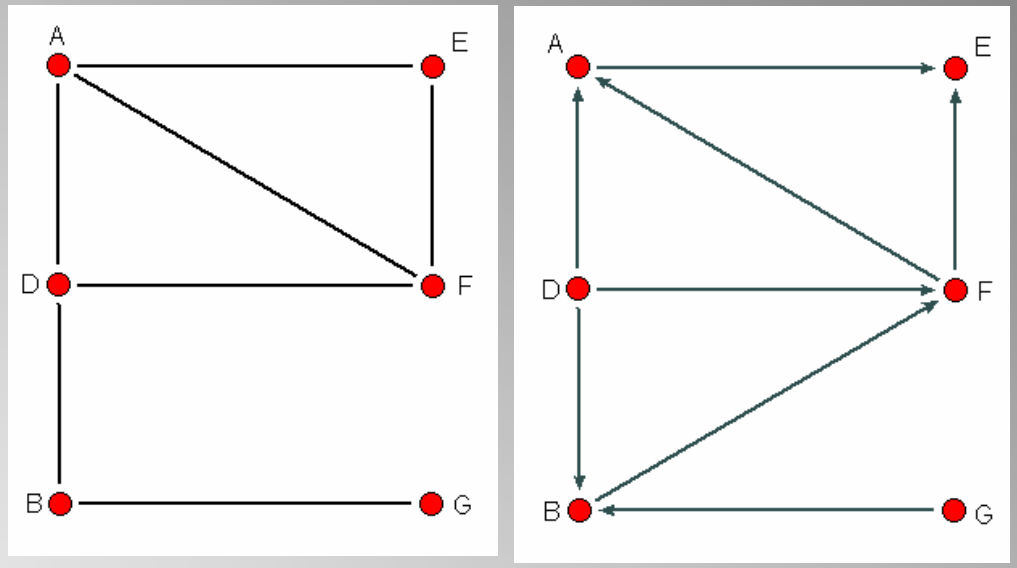
- Graphical models and existing software
- gR-initiative, organisation and status
- Core packages: gRbase, dynamicGraph, giRaph
- Examples of ”end-user-packages”
  - BRUGS
  - mimR
- Winding up – and future work

Graphical models in short

- **Graph**
  Sets of nodes and edges (directed undirected, both...)

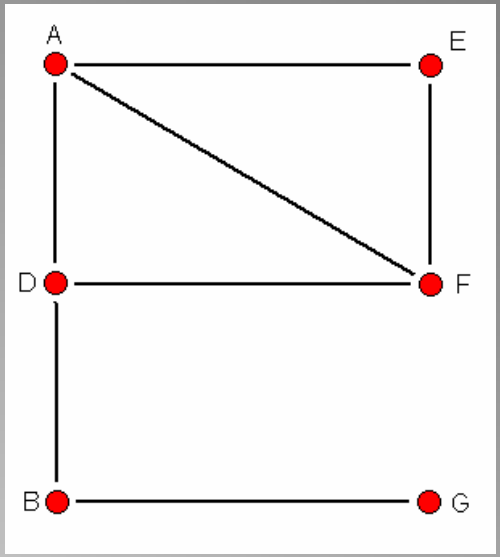
- **Graphical modelling**
  Multivariate data, exploit conditional independence properties for eg. modelling, computations, interpretation and display.

- **Examples**
  Bayesian networks, log-linear models, mixed interaction models, covariance selection models, block-recursive graphical models, BUGS models...



Example: log-linear models

- 6-dim contingency table
- Write cell probabilities as
  p(abdefg) =
  $\psi_1(aef)\ \psi_2(adf)\ \psi_3(bd)\psi_4(bg)$
- Factorisation implies conditional independencies (depicted in graph)
- Graph shows model is decomposable, implies closed form MLE



Existing Graphical Model Software – Rough characteristics

Examples: CoCo, Digram, MIM, TETRAD, Bugs

- Independent stand-alone programs.
- Developed and maintained by a small group.
- Code may not be open source.
- Each package has its own script language and GUI.
- Packages usually only run on a single platform.

Consequently,

- New packages start from ’scratch’.

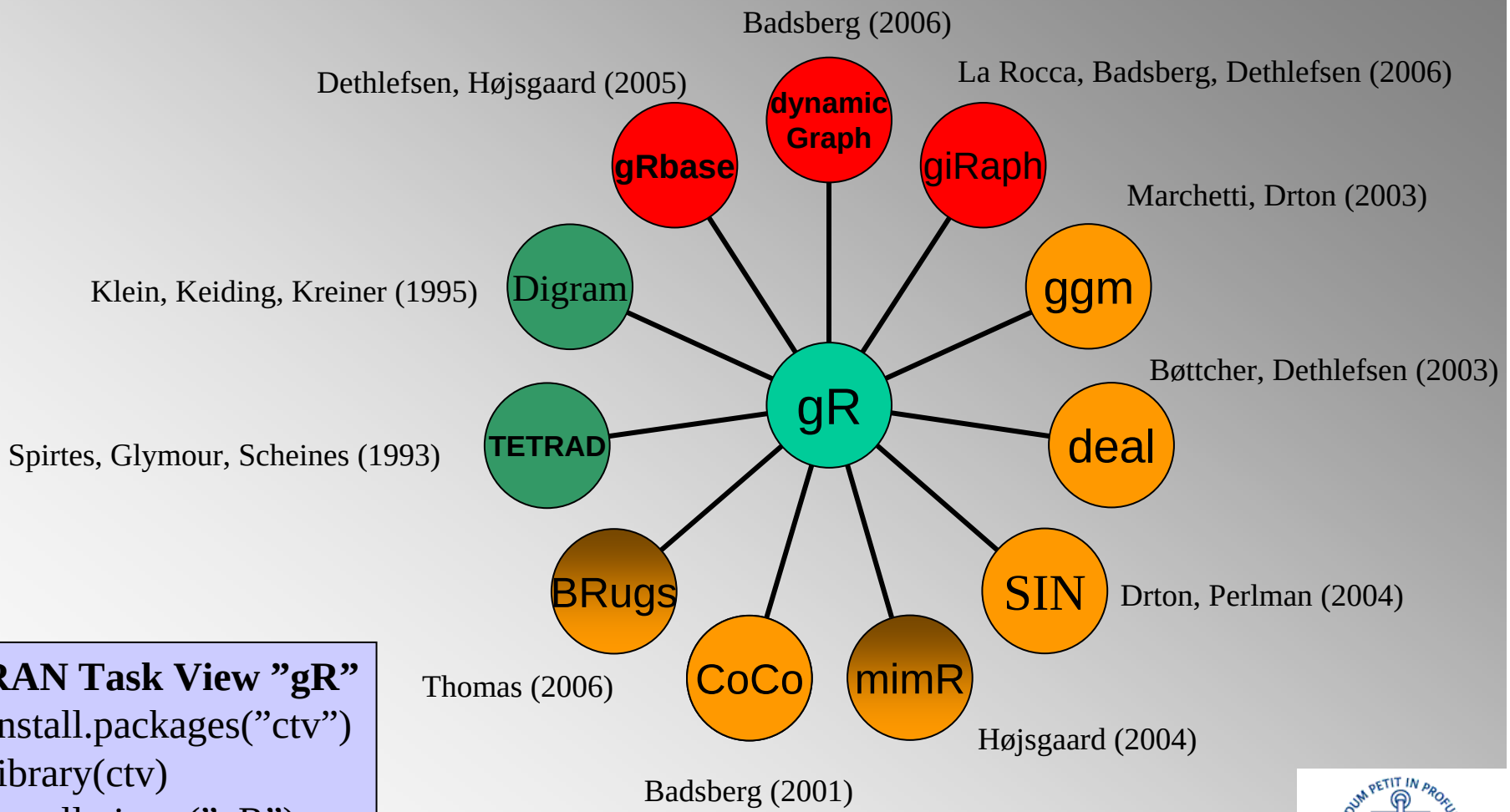
The gR initiative – aims

- Ambitious goal: Make graphical modelling possible in R - a general, multi-platform extendable software environment.
- Moreover, ease the effort of creating future packages.
  - Provide a library of efficient algorithms for computation with graphs.
  - Provide a graphical user interface easily adapted to future packages.
- Ease the use of different types of graphical models
  - Set standards for user interface and representation of data and models.
  - Develop graphical model packages for end users.
- Modest goal: Make existing graphical modelling software available to use from within R

The gR initiative - organisation

- A friendly anarchy of developers.
- gR core group: Develops core packages and sets standards.
- gR developers: Developers of packages for specific purposes. Uses gR ”core products” and other R packages.
- gR end-users: Analyses data with the developed packages using a unified user interface.

gR status



CRAN Task View "gR"
> install.packages("ctv")
> library(ctv)
> install.views("gR")

gR core packages

- **gRbase:** Defines data structure and model structure; sets standards for how to combine (data, model) with inference engines.
- **dynamicGraph:** implementation of an interactive graphical user interface for manipulating graphs, using tcl/tk.
- **giRaph:** representation of graphs and computation with graphs.

gRbase: gmData class

- gmData (graphical meta data). A common class for representing data. No matter the actual representation of data, the important characteristics are contained in a gmData object.

- Data are not always needed (some model properties do not depend on data, e.g. dimension).

gRbase: gmData class example

```
> data(HairEyeColor)  # A Table


> as.gmData(HairEyeColor)
 varNames shortNames varTypes nLevels
1     Hair          H Discrete       4
2      Eye          E Discrete       4
3      Sex          S Discrete       2
To see the values of the factors use the 'valueLabels' function
To see the data use the 'observations' function
```

gRbase: gModel class and fitting engines

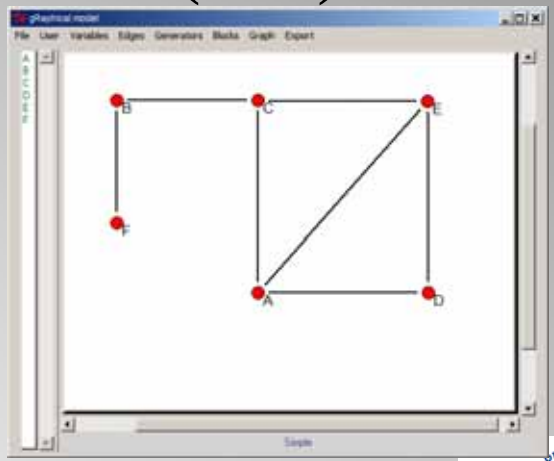
- Contains information on generators of the model (model formula) and metadata.
- Specific types of models inherit from this class.

- Example with hierarchical log-linear models (hllm)

```
data(reinis)
reinis <- as.gmData(reinis)

m1 <- hllm(~.. , reinis)
fit(m1,engine="loglm")

m2 <- hllm(~A*C*E+B*C+B*F+A*D*E, reinis)
dynamic.Graph(m2)
```



dynamicGraph

- Package writers can use pre-defined, customizable functions for
  - Displaying vertices and edges (different types)
  - Adding functionality in context menus (right-clicking) and main menus
  - Defining ”views” of a model (different types of graphs) that are linked together
  - Defining several ”models” for comparison
- End-users get a homogeneous user interface

giRaph

- Provides functionality for representation of (quite general) mathematical graphs
- Possible to work with graphs independent of their representations
- Change between representations
- Interface with dynamicGraph

giRaph

- generalGraph

```
a-b-c, a-b, b->c, b->c, a<>a
```

- generalGraph

	a	*b*	*c*
1	1	1	1
2	1	1	0
3	0	1	2
⋮			

- multiGraph

```
…
b <- {}
   -- {a}
   -> {c,c}
c <- {b,b}
   -- {}
   -> {}
```

- simpleGraph

	a	*b*	*c*
a	0	1	0
b	1	0	1
c	0	0	0

- incidenceList
- incidenceMatrix
- adjacencyList
- adjacencyMatrix

giRaph example

```
> IL <- new("incidenceList", E=list(u(1,2), d(1,3), u(3), d(2,5)),V=1:5)

> G <- new("anyGraph",incidenceList=IL)

> G <- G + u(3,1) # insert undirected edge from 3 to 1

> Gs <- as(G, ”simpleGraph”)
…
An object of class "incidenceList"
V={X1,X2,X3,X4,X5}
E={
X1--X2
X1--X3
X2->X5
}
> adjacancyMatrix(Gs)
> incidenceMatrix(Gs)
```

	X1	X2	X3	X4	X5
X1	0	1	1	0	0
X2	1	0	0	0	1
X3	1	0	0	0	0
X4	0	0	0	0	0
X5	0	0	0	0	0

	X1	X2	X3	X4	X5
[1,]	1	1	0	0	0
[2,]	1	0	1	0	0
[3,]	0	1	0	0	2

Docs on gRcore packages

- gRbase:
  Dethlefsen & SH (2005) A Common Platform for Graphical Models in R: The gRbase Package. J.Stat.Soft, 14, 2006
- dynamicGraph:
  Unpublished, status unknown
- giRaph:
  Unpublished, BUT manuscript on its way.

Specific package 1: BRUGS

- Bayesian approach,
- Joint model specified as product of conditionals,
- All parameters are explicit

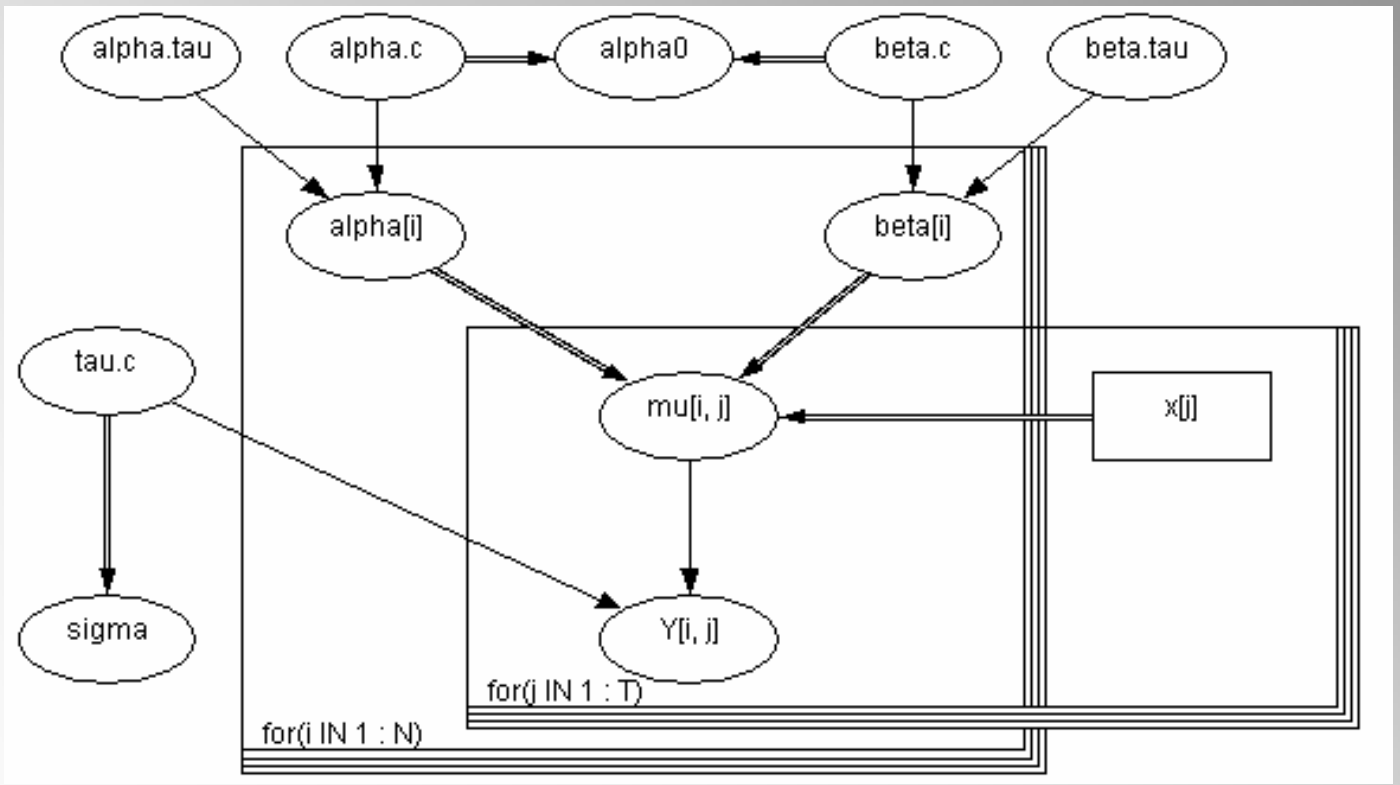
Example:

- Weekly measurements of body weight of 30 rats for 5 weeks.
- Plot suggest downward curvature

RATS model

- Random regression model

$Y_{ij} \sim \text{Normal}(\alpha_i + \beta_i(x_j - x_{bar}), \tau_c)$ $\alpha_i \sim \text{Normal}(\alpha_c, \tau_\alpha)$ $\beta_i \sim \text{Normal}(\beta_c, \tau_\beta)$



Model in S/R-like language

```
model {
  for( i in 1 : N ) {
    for( j in 1 : T ) {
      Y[i , j] ~ dnorm(mu[i , j],tau.c)
      mu[i , j] <- alpha[i] + beta[i] * (x[j] - xbar)
    }
    alpha[i] ~ dnorm(alpha.c,alpha.tau)
    beta[i] ~ dnorm(beta.c,beta.tau)
  }
  tau.c ~ dgamma(0.001,0.001)
  sigma <- 1 / sqrt(tau.c)
  alpha.c ~ dnorm(0.0,1.0E-6)
  alpha.tau ~ dgamma(0.001,0.001)
  beta.c ~ dnorm(0.0,1.0E-6)
  beta.tau ~ dgamma(0.001,0.001)
  alpha0 <- alpha.c - xbar * beta.c
}
```

```
## some usual steps (like clicking in WinBUGS):
modelCheck("ratsmodel.txt")          # check model file
modelData("ratsdata.txt")            # read data file
modelCompile(numChains=2)            # compile model with 2 chains
modelInits(rep("ratsinits.txt", 2))  # read init data file
modelUpdate(1000)                    # burn in
samplesSet(c("alpha0", "alpha"))     # alpha0 and alpha monitored
modelUpdate(1000)                    # 1000 more iterations ....
samplesStats("*")                    # the summarized results

## some plots
samplesHistory("*", mfrow = c(4, 2)) # plot the chain,
samplesDensity("alpha")              # plot the densities,
samplesBgr("alpha[1:6]")             # plot the bgr statistics, and
samplesAutoC("alpha[1:6]", 1)        # plot autocorrelations of chain
samplesDensity("alpha")              # plot the densities,
```

Output directly into R

```
> samplesStats("*")
           mean    sd MC_error val2.5pc median val97.5pc start sample
alpha[1]  239.9 2.684  0.06498   234.80  240.0     245.1  1001   2000
alpha[2]  247.8 2.733  0.05468   242.50  247.9     253.1  1001   2000
alpha[3]  252.4 2.653  0.05837   247.10  252.4     257.4  1001   2000
alpha[4]  232.6 2.641  0.06602   227.40  232.6     237.7  1001   2000
alpha[5]  231.5 2.696  0.05603   226.30  231.5     236.9  1001   2000
alpha[6]  249.7 2.689  0.05530   244.40  249.7     254.8  1001   2000
……..
alpha[30] 241.4 2.687  0.05222   236.00  241.4     246.8  1001   2000
alpha0    106.5 3.657  0.10590    99.19  106.5     113.9  1001   2000
```

Specific package 2: mimR

- Frequentist approach
- Joint specification of model (mixed interaction models, e.g. log-linear models and covariance selection models.)
- All parameters are implicit

Example: Mathematics marks data

- Mathmark data: 88 students marks on (a)lgebra, a(n)alysis, (m)echanics, (v)ectors and (s)tatistics.

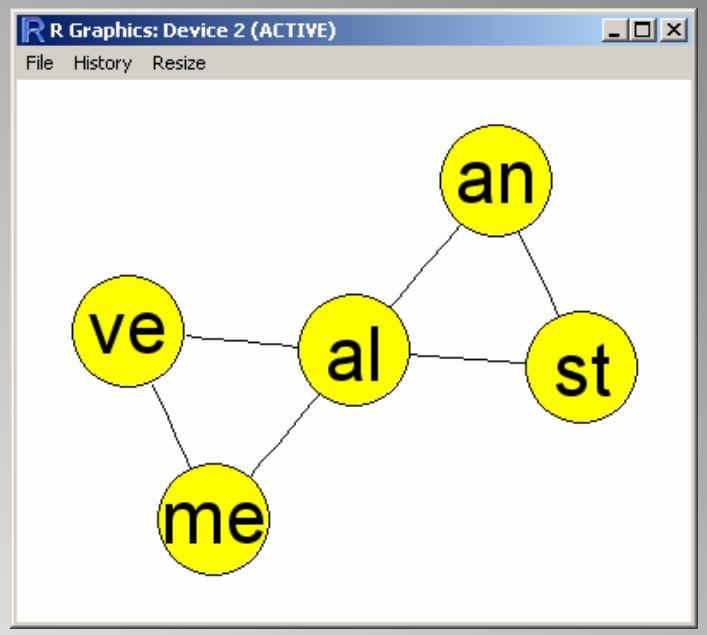
- Concentration matrix: Inverse covariance matrix.
- A zero concentration implies conditional independence

Concentrations ($\times 1000$):

	mechanics	vectors	algebra	analysis	statistics
mechanics	5.24	−2.44	−2.74	0.01	−0.14
vectors	−2.44	10.43	−4.71	−0.79	−0.17
algebra	−2.74	−4.71	26.95	−7.05	−4.70
analysis	0.01	−0.79	−7.05	9.88	−2.02
statistics	−0.14	−0.17	−4.70	−2.02	6.45

Mathmark model

```
> data(math)
> math <- as.gmData(math)
> m1 <- mim("..", data=math)
> m2 <- stepwise(m1)
> m2
> Formula: //me:ve:al + al:an:st
> -2logL: 3391.021 DF: 4
> display(m2)
> round(1000*solve(fitted(m2)$quadratic),2)
      me    ve    al    an    st
me  5.30 -2.47 -2.91  0.00  0.00
ve -2.47 10.46 -5.67  0.00  0.00
al -2.91 -5.67 28.82 -7.64 -4.99
an  0.00  0.00 -7.64  9.93 -2.06
st  0.00  0.00 -4.99 -2.06  6.51
```



Winding up – future work

- Graphical modelling packages available from within R and on CRAN. Good!
- Packages do not use gRbase-'architecture' (except for mimR). Bad ! (?)
- Biggest virtue is probably that R is used as a common basis for getting data into and results out from the packages with. Good !
- Momentum has gone down. Bad !
- Contributions welcome…

What have we learned?

- Things have evolved around us. The Rgraphviz and graph packages are now available (on all platforms)
- Hence the dynamicGraph (and giRaph?) packages might be ’redundant’ in the future.
- We should perhaps have focused more on the gRbase package itself – and on getting that part integrated with the graphical modelling packages.

People involved in gR

gR Core

- Claus Dethlefsen, Aalborg Hospital, Aarhus University Hospital
- Søren Højsgaard, Aarhus University
- Jens Henrik Badsberg, Statens Serum Institut
- Luca La Rocca, University of Modena and Reggio Emilia

Other gR people

- Susanne G. Bøttcher, Aalborg University
- Peter Dalgaard, University of Copenhagen
- David Edwards, Novo Nordisk A/S
- Poul Svante Eriksen, Aalborg University
- Peter Green, University of Bristol
- Anders Rhod Gregersen, Aalborg University
- Svend Kreiner, University of Copenhagen
- Steffen Lilholt Lauritzen, University of Oxford
- Giovanni Marchetti, University of Florence
- Duncan Murdoch, University of Western Ontario
- Andrew Thomas, University of Helsinki

The Danish activities of the gR initiative was supported by the Danish Natural Science Research Council.

References

- Badsberg, J.H. (2006). dynamicGraph: Interactive graphical tool for manipulating graphs. http://cran.r-project.org
- Badsberg, J.H. (2001). A guide to CoCo. Journal of statistical software. http://www.math.aau.dk/gr/material/CoCo/
- Bøttcher, S.G. and Dethlefsen, C. (2003). deal: A Package for Learning Bayesian Networks. Journal of Statistical Software. http://cran.r-project.org
- Dethlefsen, C. and Højsgaard, S (2005)
- Edwards, D. (2000). Introduction to Graphical Modelling. Springer Verlag. http://www.hypergraph.dk/
- Højsgaard, S. (2003). mimR -- A Package for Graphical Modelling in R. Proceedings of the 3rd international workshop on distributed statistical computing. http://cran.r-project.org
- Kreiner, S. (1989). User's guide to DIGRAM - a program for discrete graphical modelling, *Technical Report 89–10*, Statistical Research Unit, University of Copenhagen. http://www.biostat.mcw.edu/software/digram.html
- Marchetti, G. and Drton, M. (2003). ggm: an R package for Gaussian graphical models. http://cran.r-project.org
- Spirtes, P., Glymour, C. and Scheines, R.: 1993, *Causation, Prediction and Search*, Springer-Verlag, New York. Reprinted by MIT Press. http://www.phil.cmu.edu/projects/tetrad/
- Thomas, A. (1994). BUGS: a statistical modelling package. RTA/BCS Modular Languages Newsletter. http://www.mrc-bsu.cam.ac.uk/bugs/

Something different …

Where to go in June:

- Biometric Society, Nordic and Baltic Regions joint conference in Denmark, June 6-8, 2007
- More info at http://www.nbbc07.agrsci.org/
- Topics
  - Statistics in Agriculture and Veterinary Science
  - Bioinformatics and Genetics
  - Clinical Trials and Drug Development
  - Epidemiology and Statistics in Health Care
  - Statistics in Forestry, Wildlife and the Environment
  - Advances in Theory and Computational Methods
  - Other topics

Also different…

- SASweave – see http://www.cs.uiowa.edu/~rlenth/SASweave/
- Not unlike Sweave
- Allows for SAS, R, and Latex in same document
- Available on Window/Linux platforms