Network inference with

qpgraph

recent developments and future challenges

Robert Castelo
Pompeu Fabra University
Barcelona, Spain
(robert.castelo@upf.edu)

joint work with

Alberto Roverato
University of Bologna
Bologna, Italy

European Bioconductor Developer's Meeting
EMBL Heidelberg 2010

Purpose

to infer molecular regulatory networks
from microarray data using
q-order **p**artial correlation **graphs**
"qp-graphs"

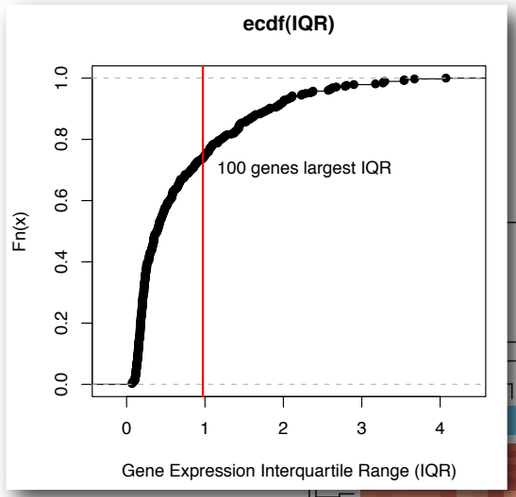
(entered the BioC release cycle on april 2009)

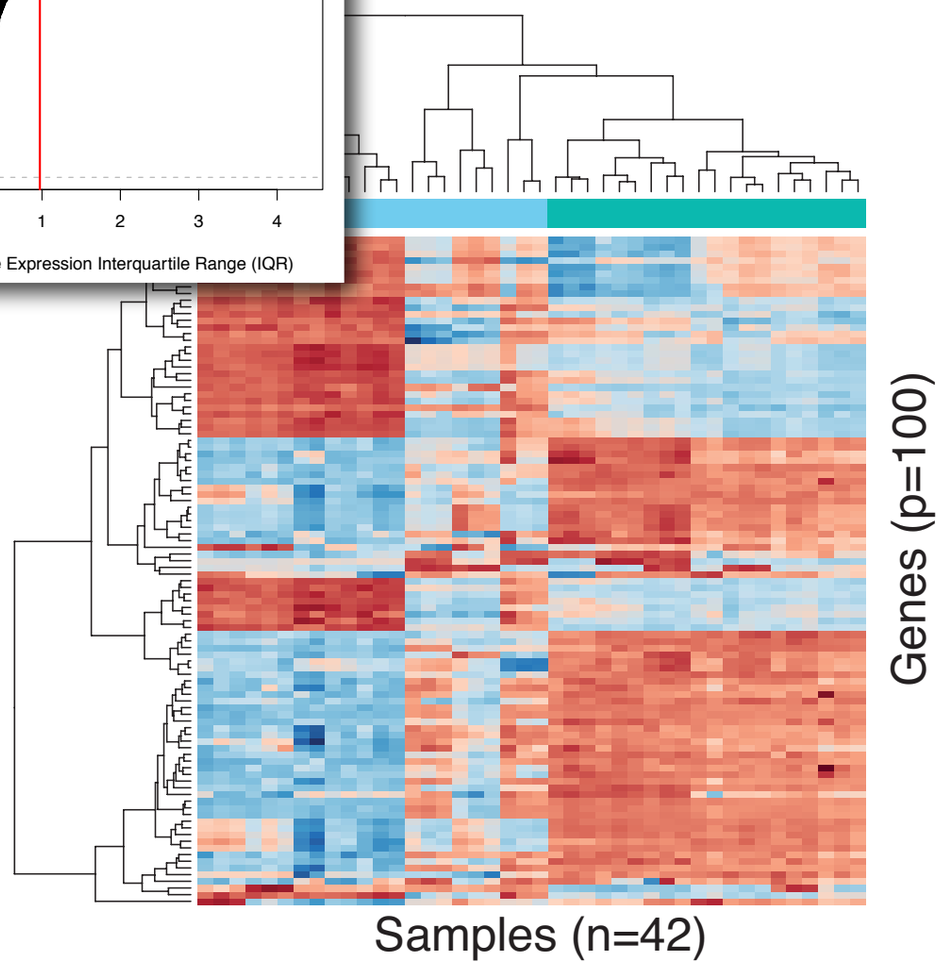
R. Castelo and A. Roverato. A robust procedure for Gaussian graphical model search from microarray data with p larger than n. *Journal of Machine Learning Research*, 7 (Dec):2621-2650, 2006.

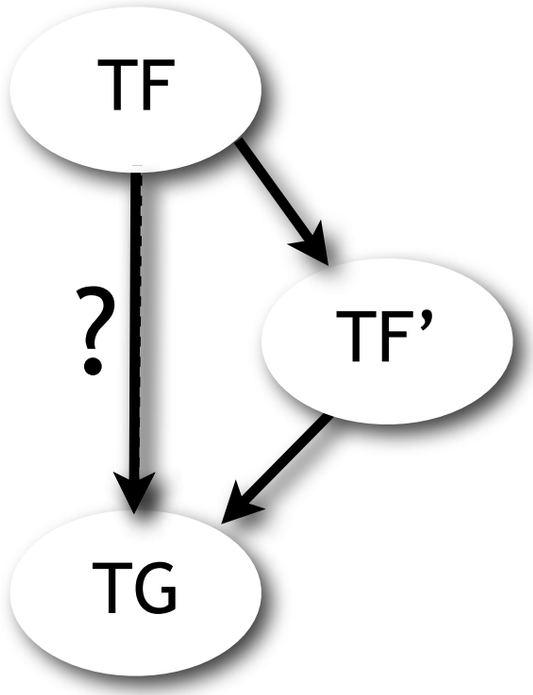
R. Castelo and A. Roverato. Reverse engineering molecular regulatory networks from microarray data with qp-graphs, *Journal of Computational Biology*, 16(2):213-227, 2009.

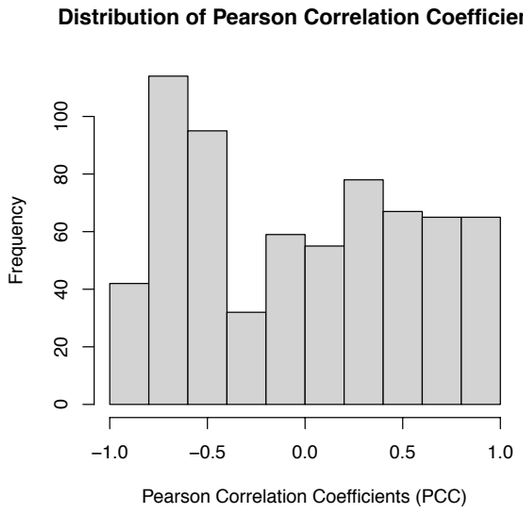
http://functionalgenomics.upf.edu/qpgraph

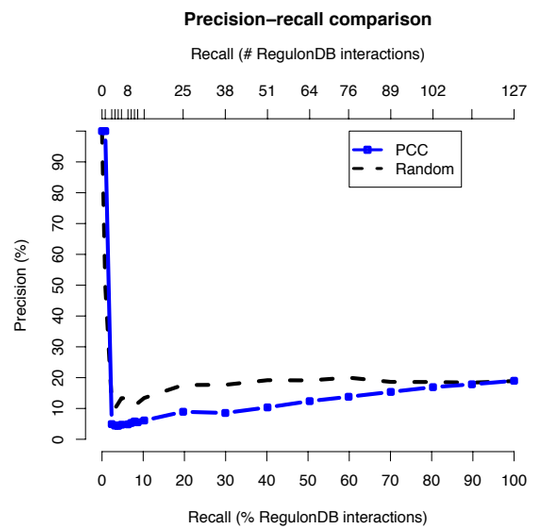
Purpose







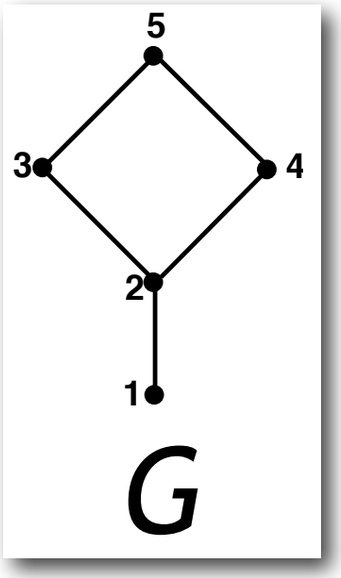




distinguish between direct and indirect interactions in the context of a microarray data set where p >> n

Partial correlations

$$V = \{1, 2, \ldots, p\} \quad X_V \sim N(\mu, \Sigma)$$



$$\Sigma^{-1} = \begin{pmatrix} \kappa_{11} & \kappa_{12} & 0 & 0 & 0 \\ \kappa_{21} & \kappa_{22} & \kappa_{23} & \kappa_{24} & 0 \\ 0 & \kappa_{32} & \kappa_{33} & 0 & \kappa_{35} \\ 0 & \kappa_{42} & 0 & \kappa_{44} & \kappa_{45} \\ 0 & 0 & \kappa_{53} & \kappa_{54} & \kappa_{55} \end{pmatrix}$$

Gaussian graphical model

$$R = V \backslash \{i, j\}$$

$$\rho_{ij.R} = \frac{-\kappa_{ij}}{\sqrt{\kappa_{ii}\kappa_{jj}}}$$

full-order partial correlation

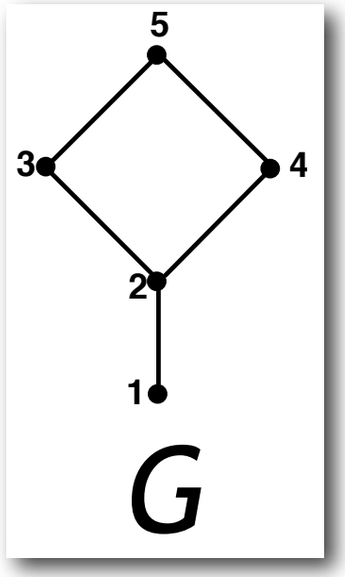
$$\rho_{ij.R} = 0 \iff \kappa_{ij} = 0 \iff X_i \perp\!\!\!\perp X_j | R$$

q-order partial correlations

$$Q \subseteq R = V \backslash \{i, j\}, \quad |Q| = q, \quad q < (n-2)$$

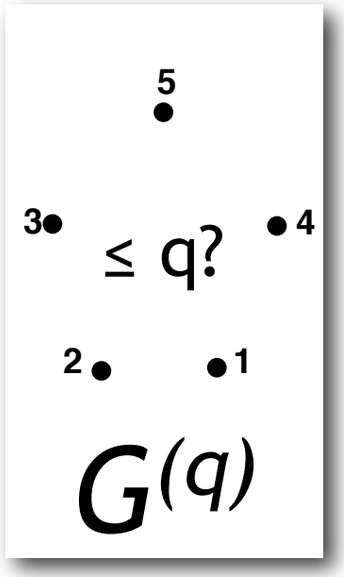
which will allow us to test $H_0 : \rho_{ij.Q} = 0$ with standard techniques where $\rho_{ij.Q} = 0 \iff X_i \perp\!\!\!\perp X_j | X_Q$ and we expect then to identify the missing edges of the graph G using $\rho_{ij.Q}$

q-order partial correlation graphs



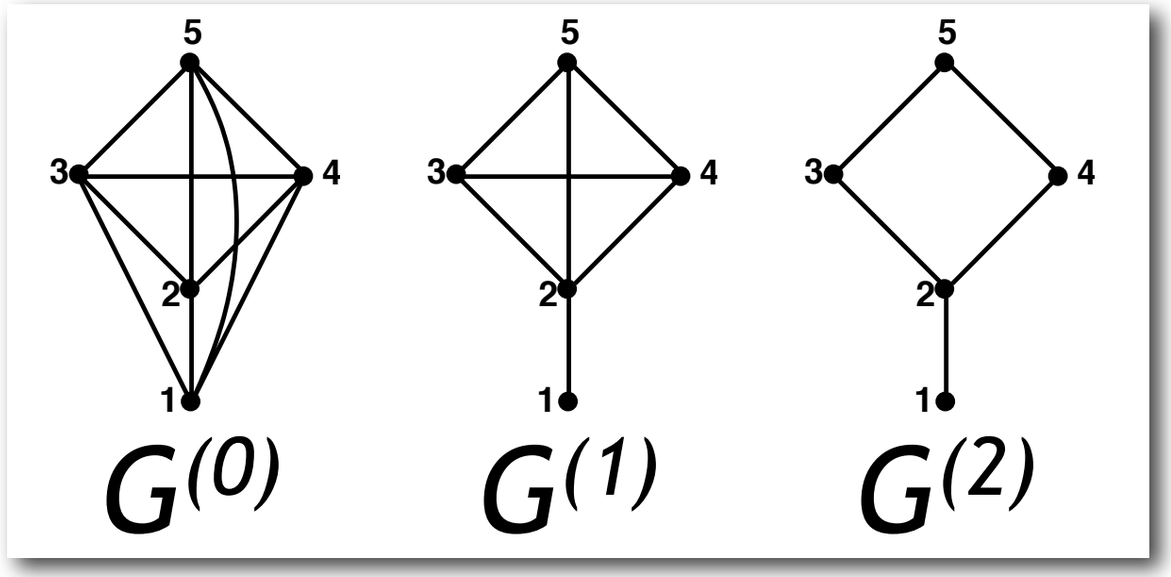
G

associated with the multivariate distribution $P(X_V)$ of dimension p

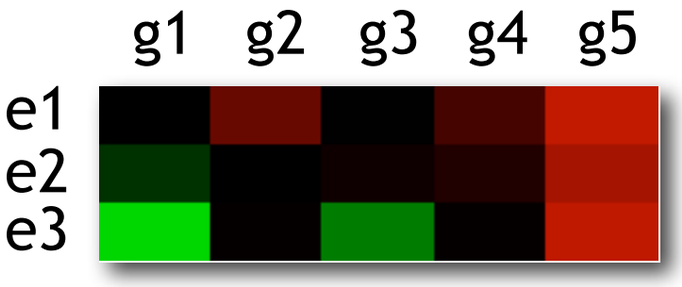


$G^{(q)}$

associated with all marginal multivariate distributions $P_Q(X_V)$ of dimension q+2



$G^{(0)}$ $G^{(1)}$ $G^{(2)}$



p=5 > n=3 but q=2 < n=3 !!

Learning with the non-rejection rate, implemented in `qpNrr()`:

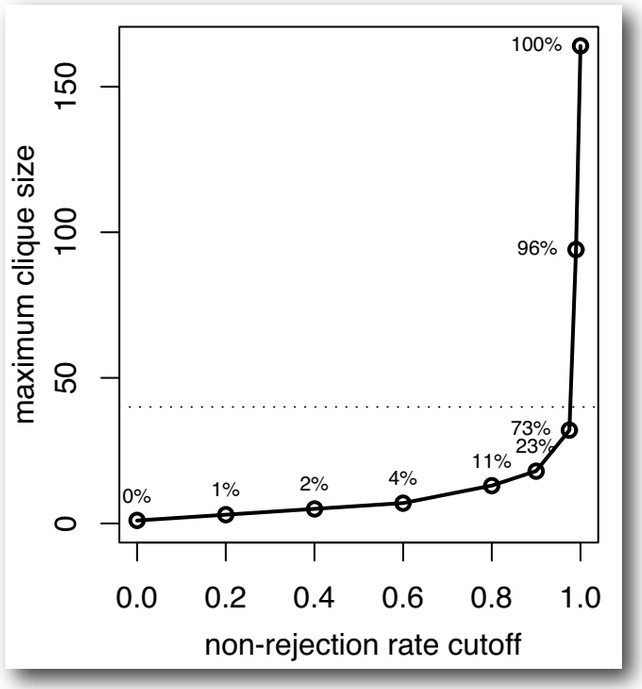
$$\mathrm{NRR}(i,j|q,D) := \beta_{ij}^q(1-\pi_{ij}^q) + (1-\alpha)\pi_{ij}^q$$

mean value Type-II errors $\beta_{ij.Q}$

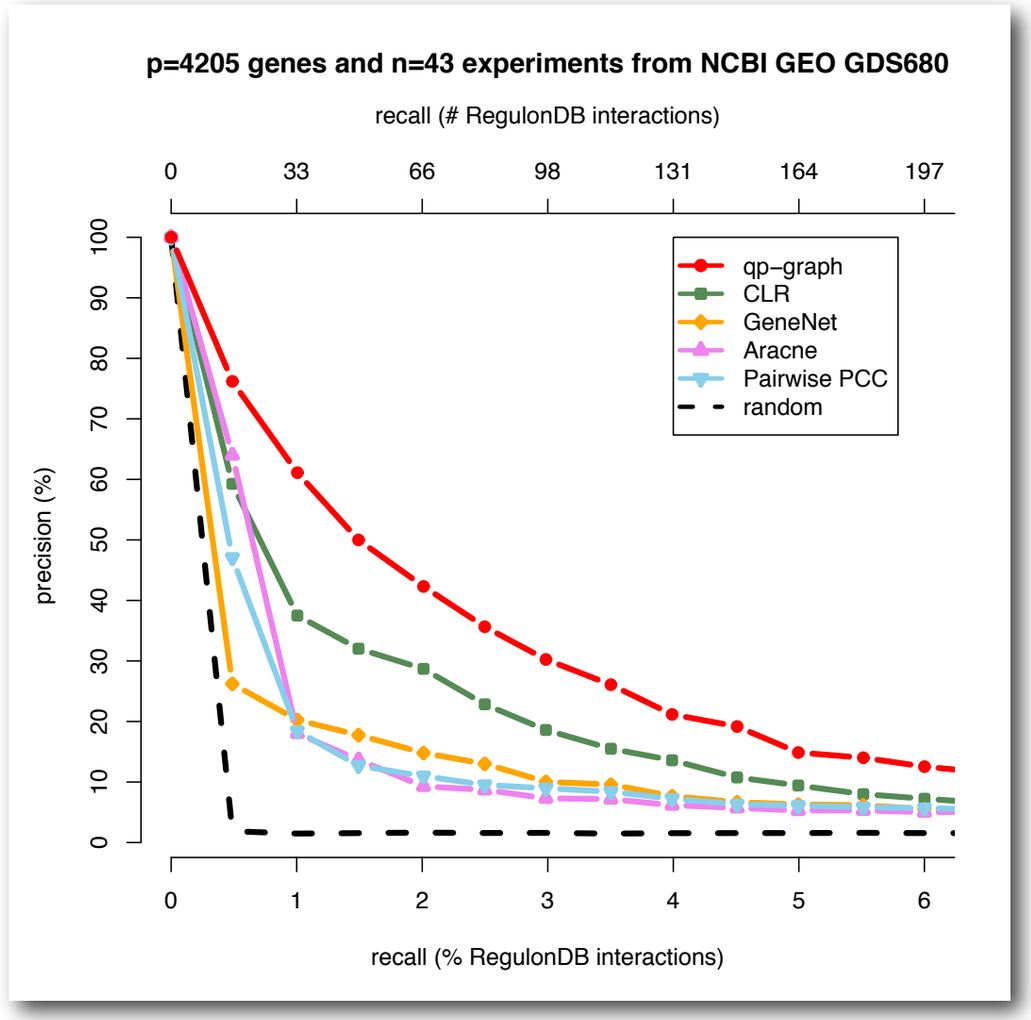
Pr. Type-I error

fraction of subsets Q that separate (i,j) in G

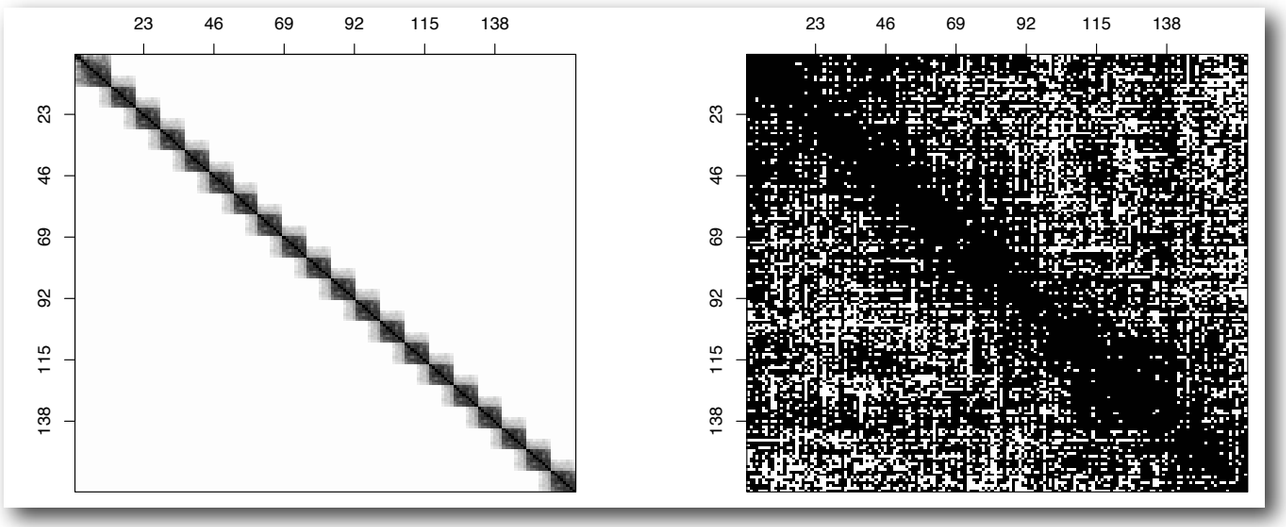
The non-rejection rate



machine learning approach
using stringent cutoffs



statistical model-based approach
using conservative cutoffs



Σ^{-1} $\hat{S}^{-1}$

Computational aspects

- non-rejection rate: Monte Carlo method performing ~100 hypothesis tests for each of the $p(p-1)/2$ pairs of genes involving sum, multiplication and inversion of matrices of size q (needs sample covariance $p \times p$ matrix)

- all matrices are symmetric, some dense, some sparse

- required graph calculations like clique enumeration or maximum clique size are provided via the cliquer library (Ostergard, 2002):

`http://users.tkk.fi/pat/cliquer.html`

Recent developments

- efficient storage and manipulation of matrices via `Matrix`
- parallel support via `snow` and argument `clusterSize=n`:

```
qpNrr(X, q=10, clusterSize=8)
```

- wall-time estimation for production clusters:

```
qpNrr(X, q=10, clusterSize=8, estimateTime=TRUE)
```

- cluster computation progress monitoring:

```
qpNrr(X, q=10, clusterSize=8, verbose=TRUE)
```

Matrix incomplete functionality

- arithmetic operations on a symmetric matrix

```
> m <- new("dspMatrix", Dim=c(500L,500L), x=rnorm(500+choose(500,2)))
> m2 <- m + m
> class(m2)
[1] "dgeMatrix"
attr(,"package")
[1] "Matrix"
> print(object.size(m), units="Mb")
1 Mb
> print(object.size(m2), units="Mb")
1.9 Mb
```

- operations involving adjacency matrices (lgc, sym, sparse)

```
> A <- Diagonal(500, TRUE)
> m[A] <- 0
Error in .local(x, i, j, ..., value) :
  not-yet-implemented 'Matrix[<-' method
```

- `cov()` does not return yet a dspMatrix ⇒ `qpCov()`

Future challenges

- integration with R & BioC visualization and analysis tools for networks

- encapsulate resulting network and its parameters into a proper object class (graphSet? sbml?)