Uncovering gene regulatory relationships using `networkBMA`

Ka Yee Yeung, Chris Fraley, Adrian E. Raftery
Departments of Microbiology (KYY) and Statistics (CF and AER)
University of Washington

This document illustrates the use of the `networkBMA` R package (Fraley et al. 2012) to uncover regulatory relationships in yeast (*Saccharomyces cerevisiae*) from microarray data measuring time-dependent gene-expression levels in 95 genotyped yeast segregants subjected to a drug (rapamycin) perturbation.

1 Data

The expression data for this vignette is provided in the `networkBMA` package in the `vignette` database, which consists of three R objects:

`timeSeries`: A 582 by 102 data frame in which the first two columns are factors identifying the replicate and time (in minutes) after drug perturbation, and the remaining 100 columns are the expression measurements for a subset of 100 genes from the yeast-rapamycin experiment described in Yeung et al. (2011). There are $582/6 = 97$ replicates (the 95 segregants plus two parental strains of the segregants), each with measurements at 6 time points. The complete time series data is available from Array Express (Parkinson et al. 2007) with accession number E-MTAB-412 (http://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-412).

`reg.known`: A 18 by 3 data frame giving known regulatory relationships among this subset of 100 genes. The first two columns give the regulator and target gene, respectively, while the third encodes the source of the regulatory information (`'YPD'` for Yeast Proteome Database (Hodges et al. 1999) and `'SCPD'` for The Promoter Database of *Saccharomyces cerevisiae* (Zhu and Zhang 1999). In this example, we constrained `reg.known` to high-confidence experimental results obtained from biochemical (non-high-throughput) experiments.

`reg.prob`: A 100 by 100 matrix, giving probability estimates for regulatory relationships in which the (i, j) entry gives the estimated probability that gene i regulates gene j. These were computed using the supervised framework integrating multiple data sources of Lo et al. (2012).

`referencePairs`: A 2-column data frame giving 287 regulator-gene pairs among the selected set of 100 genes reported from the literature. In this yeast example, the reference network was extracted from the documented evidence from the `YEASTRACT` database (Teixeira et al. 2006), which includes curated regulatory relationships from the literature inferred from high-throughput experiments.

`brem.data`: An 85 by 111 subset of the data used for network inference in yeast (Brem et al. 2002, Brem and Kruglyak 2005). The rows correspond to genes and the columns to experiments. Provided courtesy of Rachel Brem.

```
> library(networkBMA)
> data(vignette)
> dim(timeSeries)

[1] 582 102

> dim(reg.prob)

[1] 100 100

> dim(brem.data)

[1]  85 111

> reg.known
```

	Regulator	TargetGene	source
1	YDR216W	YKR009C	YPD
2	YER040W	YPL111W	YPD
3	YER040W	YKL015W	YPD
4	YER040W	YOR348C	YPD
5	YJR094C	YDR523C	YPD
6	YKL062W	YMR169C	YPD
7	YKL062W	YPL061W	YPD
8	YKL062W	YAL062W	YPD
9	YKL062W	YIL155C	YPD
10	YKL062W	YFL014W	YPD
11	YKL062W	YCR021C	YPD
12	YKL062W	YDR258C	YPD
13	YKL062W	YJR094C	YPD
14	YKL062W	YER150W	YPD
15	YKL062W	YNL194C	YPD
16	YBL103C	YNL037C	YPD
17	YKL112W	YCL064C	SCPD
18	YKL112W	YHR051W	SCPD

2 Network Modeling

Given the yeast expression data from the Rapamycin experiments, the `networkBMA` function can be invoked to estimate the probabilities of regulatory relationships using iterative Bayesian Model Averaging (Yeung et al. 2005, 2011):

```
> edges <- networkBMA(data = timeSeries[,-(1:2)],
+                     nTimePoints = length(unique(timeSeries$time)),
+                     prior.prob = reg.prob, known = reg.known,
+                     nvar = 50, ordering = "bic1+prior")
> edges[1:9,]
```

```
   Regulator TargetGene PostProb
1    YBL103C    YBL103C        1
2    YNR053C    YBL103C        1
3    YOR206W    YBL103C        1
4    YKL112W    YKL112W        1
5    YMR229C    YKL112W        1
8    YDR216W    YDR216W        1
9    YDL170W    YDR216W        1
10   YOR302W    YDR216W        1
11   YPL265W    YDR216W        1
```

For each gene g, the observed gene expression of genes at time $t-1$ serve as linear predictors for modeling the observed expression of gene g at time t. BMA modeling results in a weighted average of models consisting of relatively small numbers of predictors. The probability of gene h being a linear predictor in the model for gene g is taken as the probability that gene h regulates gene g in the network.

There are options for including known regulatory relationships and prior priobabilities in the modeling (see Lo et al. 2012), as well as for ordering the variables, and for specifying the number of ordered variables to be included in the modeling.

3 Assessment of Network Models

Although, except for synthetic data, the true underlying network is unknown, the results can be assessed using a set of regulator-target gene network edges reported in the literature. The comparison is done as follows:

- Let E be the set of regulator-target gene edges resulting from `networkBMA`, possibly reduced using a probability threshold. In the context of the example in Section 2, E corresponds to the set of edges represented in the object `edges`.
- Let K be the set of known regulator-target gene edges hardcoded in the modeling. In the example in Section 2, K corresponds to `reg.known`.
- Let L be the set regulator-target gene edges reported in the literature. In the example in Section 2, L corresponds to `referencePairs`.
- Let $E\backslash K$ and $L\backslash K$ be the set of pairs in E and L, respectively, with any hardcoded edges removed. In the example of Section 2, E represented by `edges` contains 483 pairs, and L represented by `referencePairs` contains 287 pairs. Both E and L include all 18 of the known hardcoded edges K represented by `reg.known`. Hence $E\backslash K$ contains 465 pairs, and $L\backslash K$ contains 269 pairs.
- Let U be the set of all directed pairs r-g such that r is a regulator in $L\backslash K$ and g is a target gene in $L\backslash K$. For the example of Section 2, $L\backslash K$ has 11 unique regulator genes and 99 unique target genes. So there are 11×99 or 1089 pairs in U. Assume further that the linked pairs in U are precisely those pairs in $L\backslash K$, and that all other pairs are unlinked.

- Let $U\backslash K$ be the set of pairs in U with any hardcoded eges removed (hardcoded edges may resurface in the unlinked pairs). For the example of Section 2, 17 of the 18 pairs in K occur in U, so there are 1089 - 17 = 1072 edges in $U\backslash K$.

The assessment is done using the contingency table of $(E\backslash K) \cap (U\backslash K)$ relative to $U\backslash K$. For the example of Section 2, the assessment would be done with the 57 of the 465 pairs in $E\backslash K$ that also belong to $U\backslash K$.

A function called `contabs.netBMA` is provided to produce contingency tables from a reference network according the procedure described above. Here we compare the edges produced in Section 2 by `networkBMA` modeling on the yeast data with the reference network `referencePairs` made up of results reported in the literature:

```
> ctables <- contabs.netwBMA( edges, referencePairs, reg.known,
+                                thresh=c(.5,.75,.9))
> ctables

                   TP  FN FP  TN
0.0402434734464825 23 246 33 770
0.0454930593677872 23 246 33 770
0.0547610726848114 23 246 32 771
0.0621177818036108 22 247 31 772
0.120313556351554  22 247 31 772
0.121759397028104  22 247 30 773
0.166119151754663  22 247 28 775
0.193411367909935  22 247 28 775
0.264308050711243  21 248 28 775
0.267765506855866  21 248 26 777
0.439806592523595  21 248 25 778
0.460479253259967  21 248 25 778
0.529176356437926  21 248 24 779
0.618320042575671  21 248 23 780
0.78955796127144   20 249 22 781
0.808602928008677  19 250 22 781
0.815135804441225  19 250 21 782
0.833880848245337  19 250 21 782
0.894277522630439  19 250 20 783
0.933660013920575  19 250 18 785
0.937882218196389  19 250 18 785
1                  18 251 15 788
```

Another function called '`contabs`' is provided for computing contingency tables when the true underlying network is known. The `scores` function can be used to obtain common assessment statistics from the contingency tables, including sensitivity, specificity, precision, recall, and false discovery rate among other measures.

```
> scores( ctables, what = c("FDR", "precision", "recall"))
```

```
                        FDR precision     recall
0.0402434734464825 0.5892857 0.4107143 0.08550186
0.0454930593677872 0.5892857 0.4107143 0.08550186
0.0547610726848114 0.5818182 0.4181818 0.08550186
0.0621177818036108 0.5849057 0.4150943 0.08178439
0.120313556351554  0.5849057 0.4150943 0.08178439
0.121759397028104  0.5769231 0.4230769 0.08178439
0.166119151754663  0.5600000 0.4400000 0.08178439
0.193411367909935  0.5600000 0.4400000 0.08178439
0.264308050711243  0.5714286 0.4285714 0.07806691
0.267765506855866  0.5531915 0.4468085 0.07806691
0.439806592523595  0.5434783 0.4565217 0.07806691
0.460479253259967  0.5434783 0.4565217 0.07806691
0.529176356437926  0.5333333 0.4666667 0.07806691
0.618320042575671  0.5227273 0.4772727 0.07806691
0.78955796127144   0.5238095 0.4761905 0.07434944
0.808602928008677  0.5365854 0.4634146 0.07063197
0.815135804441225  0.5250000 0.4750000 0.07063197
0.833880848245337  0.5250000 0.4750000 0.07063197
0.894277522630439  0.5128205 0.4871795 0.07063197
0.933660013920575  0.4864865 0.5135135 0.07063197
0.937882218196389  0.4864865 0.5135135 0.07063197
1                  0.4545455 0.5454545 0.06691450
```

Areas under the ROC and Precision-Recall curves covered by contingency tables can also be estimated using functions `roc` and `prc`, with the option to plot the associated curves. The following gives the ROC and Precision-Recall curvers associated with the default contingency tables, in which the threshholds are all values for posterior probabilities that appear in `edges`.

```
> roc( contabs.netwBMA( edges, referencePairs), plotit = TRUE)

       area      sector       width
0.552955098 0.002765027 0.021087680

> title("ROC")
> prc( contabs.netwBMA( edges, referencePairs), plotit = TRUE)

      area     sector      width
0.27665213 0.02082793 0.03484321

> title("Precision-Recall")
```

The resulting plots are shown in Figure 1. The output components are as follows:

- `area`: The estimated area under the curve for the horizontal sector ranging from 0 to 1. This should be used with caution when the sector in which the data falls is small.
- `sector`: The estimated area under the horizontal sector covered by the contingency tables.
- `width`: The width of the horizontal sector covered by the contingency tables.

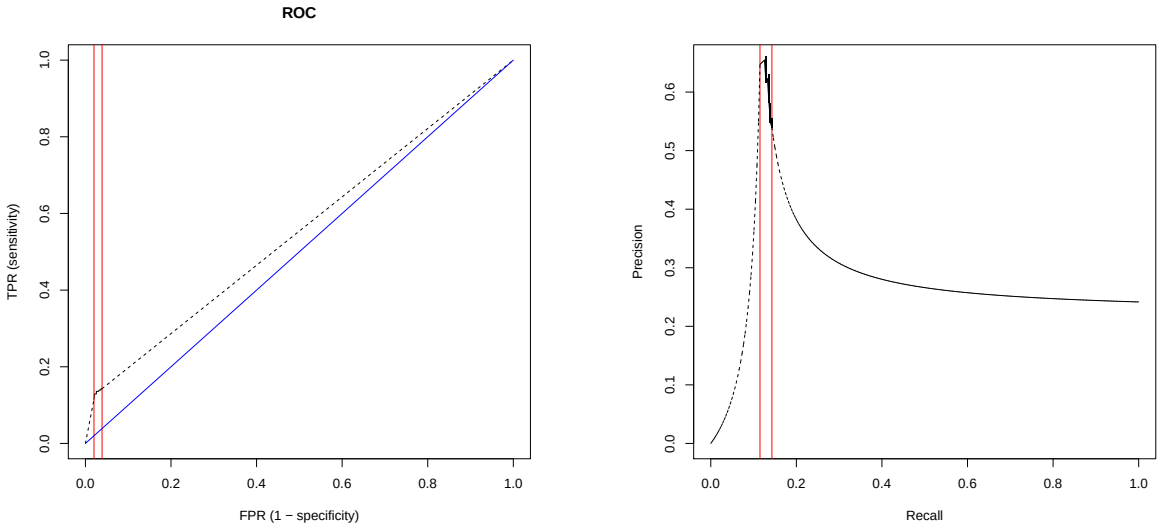

Figure 1: ROC and Precision-Recall curve sectors for a `networkBMA` model of the yeast-rapamycin test data. The black lines delineate the estimated curves. The vertical red lines delineate the range of horizontal values covered by the contingency tables. The dotted black lines are linear interpolants outside this range. The diagonal blue line on the ROC plot indicates the line betwween (0,0) and (1,1).

4 Linear Modeling for Static Gene Expression Data

`networkBMA` relies on sparse linear modeling via iterative Bayesian model averaging (BMA). BMA addresses uncertainty in model selection, and builds a weighted–average model from plausible models. The resulting model has better overall predictive ability than constituent models, and tends to use few variables from among a larger set. BMA has been iteratively extended to data with more variables that observations (Yeung at al. 2005, 2009, 2011). The `networkBMA` package includes a function, `iterateBMAlm`, for linear modeling via iterative BMA. We illustrate its use on a static gene expression dataset (without any time points), `brem.data`, to infer the regulators of a particular gene by regressing it on the expression levels of the other genes. Function `iterateBMAlm` can be applied to each gene so as to infer all edges in the network. For one gene, the procedure is as follows:

```
> gene <- "YNL037C"
> variables <- which(rownames(brem.data) != gene)
> control <- iBMAcontrolLM(OR = 50, nbest = 20, thresProbne0 = 5)
> iBMAmodel.YNL037C <- iterateBMAlm( x = t(brem.data[variables,]),
+                     y = unlist(brem.data[gene,]), control = control)
```

Function `iBMAcontrolLM` facilitates input of BMA control parameters, including `nbest` for specifying the number of best models of each size to be initially retained, `OR` for defining the width of 'Occam's window' for model exclusion, and `thresProbne0` for determining the cutoff for probability (in percent) of a variable being included in the modeling (Raftery et al.

2005). See the R help documentation for `iBMAcontrolLM` for a detailed description of these parameters, and Hoeting et al. (1999) for a tutorial on the underlying BMA paradigm. The estimated posterior probabilities (in percent) for genes that regulate `YBL103C` can be seen as follows:

```
> iBMAmodel.YNL037C$probne0[iBMAmodel.YNL037C$probne0 > 0]

   YDL170W    YHR051W    YPR002W    YML123C    YIL136W    YAL062W    YJR148W
 36.178933  75.032631 100.000000   7.010839 100.000000  32.299339  32.299339
   YNL036W    YFR022W    YPL265W    YOR348C    YCL064C    YFL014W    YOR388C
 63.821067 100.000000 100.000000 100.000000  74.315241  15.737407   7.281450
   YDR380W    YGR183C    YJL153C
 67.700661  24.967369  32.299339
```

References

[1] R. B. Brem and L. Kruglyak. The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proceedings of the National Academy of Sciences*, 102(5):1572–1577, 2005.

[2] R. B. Brem, G. Yvert, R. Clinton, and L. Kruglyak. Genetic dissection of transcriptional regulation in budding yeast. *Science*, 296(5568):752–755, 2002.

[3] C. Fraley, K. Y. Yeung, and A. E. Raftery. networkBMA: Regression-based network inference using BMA, 2012. R package distributed through Bioconductor.

[4] P. E. Hodges, A. H. Z. McKee, B. P. Davis, W. E. Payne, and J. I. Garrels. The Yeast Proteome database (YPD): a model for the organization and presentation of genome-wide functional data. *Nucleic Acids Research*, 27(1):69–73, 1999.

[5] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averging: a tutorial. *Statistical Science*, 14(4):382–401, 1999.

[6] K. Lo, A. E. Raftery, K. M. Dombek, J. Zhu, E. E. Schadt, R. E. Bumgarner, and K. Y. Yeung. Integrating external biological knowledge in the construction of regulatory networks from time-series expression data. *BMC Systems Biology*, 6:101, 2012.

[7] H. Parkinson, M. Kapushesky, M. Shojatalab, N. Abeygunawardena, R. Coulson, A. Farne, E. Holloway, N. Kolesnykov, P. Lilja, M. Lukk, R. Mani, T. Rayner, A. Sharma, E. William, U. Sarkans, and A. Brazma. ArrayExpress — a public database of microarray experiments and gene expression profiles. *Nucleic Acids Research*, 35(suppl 1):D747–D750, 2007.

[8] A. Raftery, J. Hoeting, C. Volinsky, I. Painter, and K. Y. Yeung. BMA: Bayesian model averaging, 2005. R package distributed through CRAN, revised in 2012.

[9] M. C. Teixeira, P. Monteiro, P. Jain, S. Tenreiro, A. R. Fernandes, N. P. Mira, N. Alenquer, A. T. Freitas, A. L. Oliveira, and I. Sá-Correia. The YEASTRACT database: A tool for the analysis of transcription regulatory associations in *Saccharomyces cerevisiae*. *Nucleic Acids Research*, 34(Issue supplement 1):D446–D451, 2006.

[10] K. Y. Yeung. iterativeBMA: The iterative bayesian model averaging (BMA) algorithm, 2009. R package distributed through Bioconductor, includes contributions from A. Raftery and I. Painter.

[11] K. Y. Yeung, R. E. Bumgarner, and A. E. Raftery. Bayesian model averaging: development of an improved multi-class, gene selection and classfiication tool for microarray data. *Bioinformatics*, 21(10):2394–2402, 2005.

[12] K. Y. Yeung, K. M. Dombek, K. Lo, J. E. Mittler, J. Zhu, E. E. Schadt, R. E. Bumgarner, and A. E. Raftery. Construction of regulatory networks using expression time-series data of a genotyped population. *Proceedings of the National Academy of Sciences*, 108(48):19436–19441, November 2011.

[13] J. Zhu and M. Q. Zhang. SCPD: A promoter database of the yeast *Saccharomyces cerevisiae*. *Bioinformatics*, 15(7):607–611, 1999.