Bayesian Inference of Regulatory influence on Expression (biRte)

Holger Fröhlich

May 2, 2019

1 Introduction

Expression levels of mRNA is regulated by different processes, comprising inhibition or activation by transcription factors (TF) and post-transcriptional degradation by microRNAs (miRNA). *biRte* (Bayesian Inference of Regulatory influence on Expression (biRte)) uses the regulatory networks of TFs and miRNAs together with mRNA and miRNA expression data to infer the influence of regulators on mRNA expression. Furthermore, *biRte* allows to consider additional factors such as CNVs. *biRte* has the possibility to specify Bayesian priors for the activity of each individual regulatory factor. Moreover, interaction terms between regulators can be considered. *biRte* relies on a Bayesian network model to integrate data sources into a joint likelihood model. In the model mRNA expression levels depend on the activity states of its regulating factors via a sparse Bayesian linear regression using a spikes and slab prior [**?**]. Moreover, miRNA expression levels depend on miRNA activity states. *biRte* uses Markov-Chain-Monte-Carlo (MCMC) sampling to infer activity states of regulatory factors. During MCMC, switch moves - toggling the state of a regulator between active and inactive - and swap moves - exchanging the activitiy states of either two miRNAs or two TFs - are used [8].

biRte is meant as a replacement for the earlier package *birta*. *biRte* offers several advantages compared to *birta*.

- possibility to include additional regulatory factors and data apart from TFs and miRNAs
- possibility to include target specific regulation strength values
- possibility to define a prior probabilities for activity of each individual regulator and even regulator pairs.
- significantly faster inference (about 15 fold speed-up)
- significantly higher accuracy of inference due to improved likelihood calculation
- inference of regulatory networks as a follow-up step
- possibility to work with arbitrarily complex statistical designs, if log fold changes are used.

The package can be loaded by typing:

```
> rm(list=ls())
> library(birte)
```

2 Usage of biRte

The two main functions of the package are `birteRun` and `birteLimma`. `birteLimma` is a convenience function, which passes the output of `limmaAnalysis` to `birteRun`. The most important input arguments to `birteRun` are

- **dat.mRNA**. Matrix of mRNA expression data with row names indicating genes.
- **affinities**. A weighted regulator-target graph. This is a list with at most three components (TF, miRNA, other). Each of these lists again contains a weighted adjacency list representation. See `affinities` for more information and `humanNetworkSimul` for an example. Per default weights are ignored in the inference process. IMPORTANT: gene names used in this network have to match with row names of dat.mRNA.
- **nrep.mRNA** is an integer vector, which specifies the number of replicates per condition for mRNA data

3 Applying biRte to RNAseq Data

biRte relies on the assumption that data are (multivariate) normally distributed. Application to RNAseq data is thus not immediately possible. Data should thus be transformed appropriately, e.g. via the voom + limma mechanism [5].

4 Example: Aerobic vs. anaerobic growth in E. Coli

To demonstrate the use of *biRte* we here show a most basic application to a microarray dataset by [2] together with a filtered TF-target graph [1]. The gene expression data comprises three replicates from E. Coli during aerobic growth and four replicates during anaerobic growth. The TF-target graph contains annotations for 160 transcription factors. Expression values are stored in an *ExpressionSet*.

```
> library(Biobase)
> data(EColiOxygen)
> EColiOxygen

ExpressionSet (storageMode: lockedEnvironment)
assayData: 4205 features, 7 samples
  element names: exprs
protocolData: none
phenoData
  rowNames: GSM18261 GSM18262 ... GSM18289 (7 total)
  varLabels: Strain GrowthProtocol GenotypeVariation Description
  varMetadata: labelDescription
featureData
  featureNames: 1 2 ... 4205 (4205 total)
  fvarLabels: symbol Entrez
  fvarMetadata: labelDescription
experimentData: use 'experimentData(object)'
  pubMedIds: 15129285
Annotation: org.EcK12.eg.db

> head(exprs(EColiOxygen))
```

```
        GSM18261 GSM18262  GSM18263  GSM18286  GSM18287  GSM18288  GSM18289
947315 10.277125 10.22119 10.410919 10.208393 10.179176 10.186009 10.009045
945490 10.138638 10.17328 10.215396 10.170649  9.993040 10.277822  9.968522
944896 11.016805 11.28574 11.308092 11.287854 11.582083 11.632015 11.463312
945321  8.726455  9.00633  8.973156  9.149897  9.245039  9.298647  9.113609
944895 11.179725 11.09959 11.270414 10.792218 10.750200 11.289802 10.960788
947758 12.399980 12.50940 12.043803 12.460848 12.531210 12.440010 12.510939
```

Before starting our biRte analysis we try to simplify the TF-target by clustering regulators with highly overlapping target gene sets. Then we determine possible interactions between regulators by looking for regulators, which have an overlap that is large enough to be considered, but not as large that the effect is indistinguishable from main effects by individual regulators.

Afterwards, differentially expressed genes are calculated using `limmaAnalysis`. The result is then passed to `biRteLimma`, together with the TF-target graph `EColiNetwork`. As a final step we use biRte to look for regulator activities that can explain differential gene expression between anaerobic and aerobic growth. In a real application the number of MCMC iterations should be increased significantly:

```
> # prepare network
> affinities = list(TF=sapply(names(EColiNetwork$TF), function(tf){w = rep(1, length(EColiNetwork
> affinities = simplify(affinities)
> affinities$other = proposeInteractions(affinities)
> # prepare data
> mydat = exprs(EColiOxygen)
> colnames(mydat) = make.names(paste(pData(EColiOxygen)$GenotypeVariation, pData(EColiOxygen)$Gro
> limmamRNA = limmaAnalysis(mydat, design=NULL, "wild.type.anaerobic - wild.type.aerobic")
> mydat = cbind(mydat[,colnames(mydat) =="wild.type.aerobic"],  mydat[,colnames(mydat) == "wild.t
> ecoli_result = birteLimma(dat.mRNA=mydat, limmamRNA=limmamRNA, affinities=affinities, niter=500

> plotConvergence(ecoli_result, title="E. Coli")
```

The log-likelihood is shown in Figure 1. Below we show those TFs, who reveal a marginal activity probability of larger than a cutoff corresponding to an expected false positve rate of 0.001. We look at the total number of target genes together with the number of differentially expressed target genes for the predicted TFs:

```
> tau = suggestThreshold(ecoli_result$post[,1])

start values: (alpha, beta) =  1.5 5 lambda =  0.5 0.5
logLik =  -999.1348 (alpha, beta) =  4.839323 149.4132 lambda =  0.1104342 0.8895658
logLik =  -999.2411 (alpha, beta) =  4.83932 149.4132 lambda =  0.1198346 0.8801654
logLik =  -999.2411 (alpha, beta) =  4.83932 149.4132 lambda =  0.1198346 0.8801654
[1] "converged!"

> activeTFs = rownames(ecoli_result$post)[ecoli_result$post[,1] > tau]
> activeTFs

 [1] "betI"             "caiF"             "dcuR"             "fnr"
 [5] "fhlA"             "fruR"             "fur"              "gadW"
 [9] "galR"             "gatR"             "gcvA"             "iclR"
[13] "narL"             "glnG"             "prpR"             "soxR"
[17] "arcA_crp"         "arcA_fnr"         "crp_fnr"          "gadE_gadW"
[21] "fhlA_hyfR"        "arcA_ihfA_U_ihfB" "fis_narL"         "modE_narL"
[25] "modE_narP"        "dnaA_nrdR"        "iscR_oxyR"        "gadE_pdhR"
[29] "gadE_torR"
```

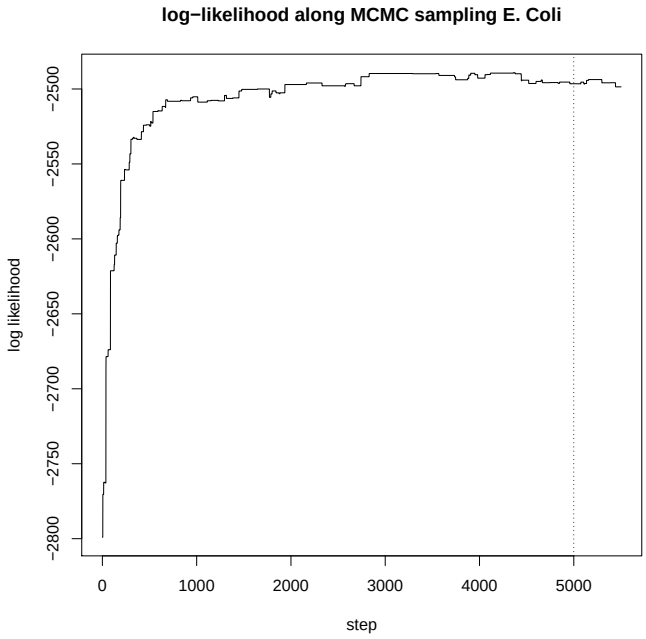

Figure 1: Log-likelihood during MCMC sampling for the E. Coli data set.

```
> if(length(activeTFs) > 0){
+         DEgenes = rownames(limmamRNA$pvalue.tab)[limmamRNA$pvalue.tab$adj.P.Val < 0.05 & abs(li
+         genesetsTF = c(sapply(affinities$TF, names), sapply(affinities$other, names))
+         DEgenesInTargets = sapply(genesetsTF[intersect(activeTFs, names(genesetsTF))],
+         function(x) c(length(which(x %in% DEgenes)), length(x)))
+         rownames(DEgenesInTargets) = c("#DEgenes", "#targets")
+         DEgenesInTargets[,order(DEgenesInTargets["#targets",], decreasing=TRUE)]
+ }
```

```
         fnr narL fur crp_fnr arcA_fnr arcA_crp glnG arcA_ihfA_U_ihfB fruR fhlA
#DEgenes  35   31   1      10        6        4    0                2    3   15
#targets 264  101  78      77       74       57   43               40   36   29
         fis_narL modE_narL modE_narP fhlA_hyfR caiF galR dcuR gadW gatR
#DEgenes        7         2         0         0    0    0    5    2    0
#targets       26        18        15        11   10    9    7    6    6
         gadE_gadW iscR_oxyR gadE_pdhR gadE_torR prpR betI gcvA iclR soxR
#DEgenes         2         0         0         1    0    0    0    0    0
#targets         6         6         5         5    4    3    3    3    2
         dnaA_nrdR
#DEgenes         0
#targets         2
```

We can ask, how well log fold changes predicted by our biRte model agree with obseved log fold changes:

```
> pred = birtePredict(ecoli_result, rownames(mydat))
> cor(pred[[1]][[1]]$mean, limmamRNA$pvalue.tab[rownames(mydat), "logFC"])
```

```
[1] 0.4892384
```

Once again it should be noted that in a real application the MCMC sampler should run much longer and hence better results are expected.

5 Using Regulator Expression Data

One of the strength of *biRte* is that measurements of regulators can be integrated smoothly into the inferenceprocess.

In our example situation no miRNA expression data is available, but some transcription factors have been measured on the microarray. In accordance with published results [7], *biRte* does not suppose that the mRNA expression levels of a TF and its (putative) target genes are correlated. However, differential TF expression on mRNA level might still give a hint on activity differences on protein level. Thus, *biRte* allows to integrate expression data of differentially expressed TFs.In our case `TFexpr` contains an excerpt of `EColiOxygen`. It comprises mRNA expression for all 160 TFs in `EColiNetwork`. The row names of the expression matrix were converted to the corresponding TF identifiers in `EColiNetwork`.

```
> head(exprs(TFexpr))
```

```
      GSM18261  GSM18262  GSM18263  GSM18286  GSM18287  GSM18288  GSM18289
acrR  8.277473  8.309069  8.504610  7.857166  7.686808  8.111077  7.915678
ada   9.277946  9.540328  9.186303  9.578132  9.646316  9.444881  9.384217
adiY  6.330554  6.555999  6.686157 10.801038 10.986309  8.788498  8.612713
agaR 10.854649 10.726303 10.782988 10.936007 11.041171 11.200971 11.130323
allR 11.324718 11.193124 11.389784 11.102606 11.274170 11.273160 10.936546
allS  8.520564  8.764251  8.693574  8.806117  8.806997  8.705715  8.272239
```

Differential expression of these TFs can be assessed by subsetting our previous `limmamRNA` object. We use the obtained results to define an informative prior for each regulator and regulator-regulator interaction, before running a biRte analysis, and to set up a reasonable initial state for the sampler:

```
> limmaTF = limmamRNA
> limmaTF$pvalue.tab = limmaTF$pvalue.tab[rownames(limmaTF$pvalue.tab) %in% fData(TFexpr)$Entrez,
> names(limmaTF$lm.fit$sigma) = as.character(fData(EColiOxygen)$symbol[match(names(limmaTF$lm.fit
> rownames(limmaTF$pvalue.tab) = as.character(fData(EColiOxygen)$symbol[match(rownames(limmaTF$pv
> diff.TF = rownames(limmaTF$pvalue.tab)[limmaTF$pvalue.tab$adj.P.Val < 0.05 & abs(limmaTF$pvalue
> theta.TF = rep(1/length(affinities$TF), length(affinities$TF))
> names(theta.TF) = names(affinities$TF)
> theta.other = rep(1/length(affinities$other), length(affinities$other))
> names(theta.other) = names(affinities$other)
> theta.other[unique(unlist(sapply(diff.TF, function(tf) grep(tf, names(theta.other)))))] = 0.5 #
> init.TF = theta.TF
> init.TF = (init.TF >= 0.5)*1
> init.other = theta.other
> init.other = (init.other >= 0.5)*1
> # note that niter and nburnin are much too small in practice
> ecoli_TFexpr = birteLimma(dat.mRNA=mydat, data.regulators=list(TF=exprs(TFexpr)), limmamRNA=lim

> tau = suggestThreshold(ecoli_TFexpr$post[,1])

start values: (alpha, beta) =  1.5 5 lambda =  0.5 0.5
logLik =  -899.1556 (alpha, beta) =  9.403047 149.4132 lambda =  0.1860337 0.8139663
logLik =  -899.2085 (alpha, beta) =  9.403042 149.4132 lambda =  0.194215 0.805785
[1] "converged!"
```

```
> activeTFs = ecoli_TFexpr$post[ecoli_TFexpr$post[,1] > tau,1]
> activeTFs
```

```
          adiY           appY           caiF           gadE
         1.000          1.000          0.578          1.000
          lrhA           ompR           rstA           zntR
         1.000          1.000          1.000          1.000
          betI           bolA           cusR            fnr
         1.000          1.000          1.000          1.000
          feaR            fur           glcC           hcaR
         1.000          1.000          1.000          1.000
          iscR           lldR           marA           mhpR
         1.000          1.000          1.000          1.000
          mntR           narL           soxS           dcuR
         1.000          1.000          1.000          1.000
          fhlA           galS           gatR           gcvA
         1.000          1.000          1.000          1.000
          hyfR           glnG       arcA_crp       arcA_fnr
         1.000          1.000          1.000          1.000
     gadE_gadW      gadE_gadX arcA_ihfA_U_ihfB      appY_iscR
         1.000          0.048          1.000          1.000
     iscR_narL      appY_narP      iscR_narP      narL_narP
         1.000          0.932          1.000          1.000
     argP_nrdR      lrhA_ompR      iscR_oxyR      gadE_pdhR
         1.000          0.346          1.000          1.000
     csgD_rstA      ompR_rstA rcsA_U_rcsB_rstA      gadE_torR
         0.448          0.090          0.242          1.000
     uidR_uxuR
         0.334
```

6 Network Inference

After having determined active regulators one may ask, in which way these regulators influence each other. Bayesian Networks are a principal possibility, but would usually require direct measurements of regulators, which is difficult to obtain for TFs. Moreover, the typically small sample size imposes a principal limitation. We thus restrict ourselves to subset relationships between differentially expressed target genes. These subset relationships can have two possible interpretations: One possibility is that egulator A acts upstream of regulator B, if differential targets of B are a subset of those of A. Another possibility is that A and B jointly co-regulate certain target genes. The idea of (noisy) subset relationships has striking similarities to Nested Effects Models (NEMs) [**?**, 4], which have been introduced for causal network inference from perturbation data. Although in our case we do not have targeted perturbations of individual regulators, probabilistic inference of subset relationships between differentially expressed targets of regulator pairs can be effectively solved via NEM inference. *biRte* uses the pair-wise inference algorithm discussed in [6] as default.

biRte offers a convenience function `estimateNetwork` for this purpose. The function decomposes clusters of active regulators into individual regulators and performs appropriate calls to functions from *nem* [3]. The output is a network indicating subset relationships between differential targets of active regulators. In our example this would be done as follows:

```
> DEgenes = rownames(limmamRNA$pvalue.tab)[limmamRNA$pvalue.tab$adj.P.Val < 0.05 & abs(limmamRNA$
> net = estimateNetwork(ecoli_TFexpr, thresh=tau, de.genes=DEgenes)
> library(nem)
> if(require(Rgraphviz)){
```

```
+   plot(net, transitiveReduction=TRUE)
+ }
```

This yields the network shown in Figure 2. In addition to the network structure we can investigate the estimated dependencies regulator-gene dependencies in more depth. This may give additional insights whether a particular gene is a direct target of a particular transcription factor or not and hence allow for filtering out false positive target predictions:

```
> net$mappos
```

In our case there are several totally unspecific target genes (assigned to "null"), which may indicate false positive target gene predictions.

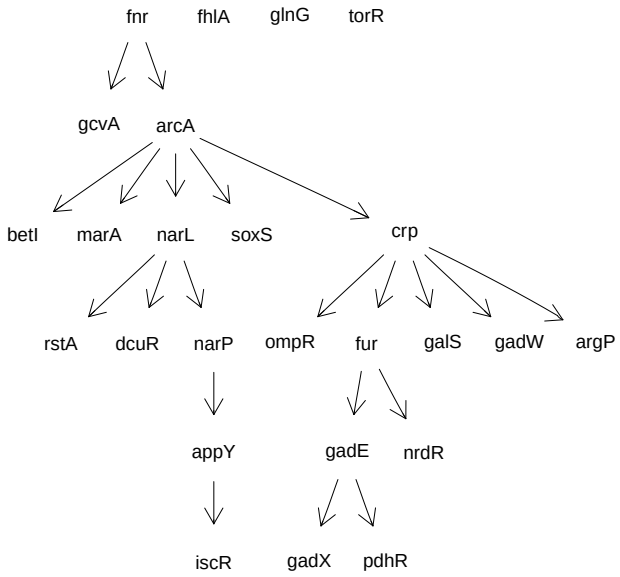

Figure 2: Inferred network between active TFs.

7 Regulator Activities in Single Samples

One may ask, in how far activity of regulators differs from one sample to another. *biRte* now allows for addressing such a question. Essentially, the idea is to run *biRte*'s Bayesian inference procedure independently for each single sample. That means for each individual sample we look for the regulator combination that could explain the observed expression level of genes. The output is a samples x regulators matrix containing either the marginal posterior probabilities for each regulator to be active in a certain sample, or the most likely configurations of active regulators in each sample. In our example we run single sample analysis as follows:

```
> ss_ecoli_TFexpr = birteLimma(dat.mRNA=mydat, data.regulators=list(TF=exprs(TFexpr)), limmamRNA=
> ss_ecoli_TFexpr[,colSums(ss_ecoli_TFexpr) > 0] # regulators that are active in at least one sam
```

8 Conclusion

biRte integrates regulator expression and mRNA data into a probabilistic framework to make

inference on regulator activities. It is a step towards the important goal to unravel causal mechanisms of gene expression changes under specific experimental or natural conditions. A unique feature is the combination with network inference.
This vignette was generated using the following package versions:

- R version 3.6.0 (2019-04-26), `x86_64-w64-mingw32`
- Locale: `LC_COLLATE=C`, `LC_CTYPE=English_United States.1252`, `LC_MONETARY=English_United States.1252`, `LC_NUMERIC=C`, `LC_TIME=English_United States.1252`
- Running under: `Windows Server 2012 R2 x64 (build 9600)`
- Matrix products: default
- Base packages: base, datasets, grDevices, graphics, grid, methods, parallel, stats, utils
- Other packages: Biobase 2.44.0, BiocGenerics 0.30.0, Rcpp 1.0.1, RcppArmadillo 0.9.400.2.0, Rgraphviz 2.28.0, birte 1.20.0, graph 1.62.0, nem 2.58.0
- Loaded via a namespace (and not attached): MASS 7.3-51.4, Matrix 1.2-17, RBGL 1.60.0, RColorBrewer 1.1-2, boot 1.3-22, class 7.3-15, codetools 0.2-16, compiler 3.6.0, e1071 1.7-1, foreach 1.4.4, glmnet 2.0-16, iterators 1.0.10, knitr 1.22, lattice 0.20-38, limma 3.40.0, plotrix 3.7-5, statmod 1.4.30, stats4 3.6.0, tools 3.6.0, xfun 0.6

References

[1] R. Castelo and A. Roverato. Reverse engineering molecular regulatory networks from microarray data with qp-graphs. *J Comput Biol*, 16(2):213–227, Feb 2009.

[2] M. W. Covert, E. M. Knight, J. L. Reed, M. J. Herrgard, and B. O. Palsson. Integrating high-throughput and computational data elucidates bacterial networks. *Nature*, 429(6987):92–96, May 2004.

[3] H. Fröhlich, T. Beißbarth, A. Tresch, D. Kostka, J. Jacob, R. Spang, and F. Markowetz. Analyzing gene perturbation screens with nested effects models in R and bioconductor. *Bioinformatics*, 24(21):2549–2550, Nov 2008.

[4] H. Fröhlich, A. Tresch, and T. Beissbarth. Nested effects models for learning signaling networks from perturbation data. *Biom J*, 51(2):304–323, Apr 2009.

[5] C. W. Law, Y. Chen, W. Shi, and G. K. Smyth. voom: Precision weights unlock linear model analysis tools for rna-seq read counts. *Genome Biol*, 15(2):R29, 2014.

[6] F. Markowetz, D. Kostka, O. Troyanskaya, and R. Spang. Nested effects models for high-dimensional phenotyping screens. *Bioinformatics*, 23:i305 – i312, 2007.

[7] M. Wu and C. Chan. Learning transcriptional regulation on a genome scale: a theoretical analysis based on gene expression data. *Brief Bioinform*, May 2011.

[8] B. Zacher, K. Abnaof, S. Gade, E. Younesi, A. Tresch, and H. Fröhlich. Joint bayesian inference of condition-specific mirna and transcription factor activities from combined gene and microrna expression data. *Bioinformatics*, 28(13):1714–1720, Jul 2012.