fbseq: An R package for fully Bayesian analysis of RNAseq data

Jarad Niemi and Will Landau
Department of Statistics, Iowa State University

Contact Information:
Department of Statistics
Iowa State University
2208 Snedecor Hall, Ames, IA
niemi@iastate.edu

Abstract

RNAseq data comprise a set of counts for G genes on a set of N samples. The counts are positively associated with the expression levels of mRNA of particular genes. Since G is typically much larger than N, we construct a hierarchical overdispersed count regression model that borrows information across genes. We estimate the high-dimensional posterior using a Markov chain Monte Carlo algorithm that utilizes a step-out slice sampler whenever Gibbs steps are unavailable. We implemented our approach in a set of three R packages: `fbseq`, `fbseqOpenMP`, and `fbseqCUDA`. The `fbseq` package provides a user interface to the two backends. The `fbseqOpenMP` backend provides an implementation that is parallelized across CPU cores and is therefore useful for testing. The `fbseqCUDA` backend provides an implementation on a NVIDIA graphics processing unit (GPU) and therefore is suitable to analysis of realistic data sets. The GPU version provides a million iterations of the sampler in a couple of hours with computation time scaling linearly in the number of genes and the number of samples.

Model

Let y_{gn} be the RNA-seq count for sample n ($n = 1, \ldots, N$) and gene g ($g = 1, \ldots, G$). Let X be the $N \times L$ model matrix for gene-specific effects $\beta_g = (\beta_{g1}, \ldots, \beta_{gL})$ and let X_n be the n^{th} row of X. We assume an over-dispersed hierarchical regression model depicted below.

$$
\begin{aligned}
y_{gn} &\overset{\text{ind}}{\sim} \text{Poisson}\left(\exp\left(h_n + \varepsilon_{gn} + X_n\beta_g\right)\right) \\
\varepsilon_{gn} &\overset{\text{ind}}{\sim} \text{Normal}(0, \gamma_g) \\
\gamma_g &\overset{\text{ind}}{\sim} \text{Inverse-Gamma}\left(\frac{\nu}{2}, \frac{\nu\tau}{2}\right) \\
\nu &\sim \text{Uniform}(0, d) \\
\tau &\sim \text{Gamma}(a, \text{rate} = b) \\
\beta_{g\ell} &\overset{\text{ind}}{\sim} \text{Normal}(\theta_\ell, \sigma_\ell^2) \\
\theta_\ell &\overset{\text{ind}}{\sim} \text{Normal}(0, c_\ell^2) \\
\sigma_\ell &\overset{\text{ind}}{\sim} \text{Uniform}(0, s_\ell)
\end{aligned}
$$

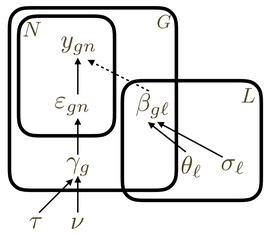

Figure 1: Directed acyclic graph (DAG) representation of the RNA-seq model, along with a formulaic representation on the left. The box with G in the corner indicates that each parameter inside represents multiple nodes, each specific to a value of $g = 1, \ldots, G$. The analogous interpretation holds for the boxes with N and L, respectively. The dashed arrow from $\beta_{g\ell}$ to y_{gn} indicates that an edge is present if and only if $X_n\beta_g$ is a non-constant function of $\beta_{g\ell}$: that is, if and only if $X_{n\ell} \neq 0$, where X is the model matrix and X_n is its n'th row.

The h_n's are normalization constants estimated from the data, and they take into account sample-specific nuisance effects such as sequencing depth. The γ_g parameters are analogous to the typical gene-specific negative-binomial dispersion parameters used in many other methods of RNA-seq data analysis. Generally, we are interested in the β_g terms which relate elements of the model parameterization to gene expression levels.

GPU parallelization

To fit the model to RNA-seq data, we use an overall Gibbs sampling structure and apply the univariate stepping-out slice sampler [3] within each of several Gibbs steps. In each of the steps of the algorithm below, a slice sampler is used to sample from all non-normal full conditionals. Each slice-sampled parameter (γ_1, γ_2, $\varepsilon_{50,5}$, etc.) has its own tuning variable and auxiliary variable. Slice sampling is used for the gamma and inverse-gamma full conditionals in addition to the full conditionals with unknown distributional form. This is because CURAND, the random number generation library for CUDA, has no gamma sampler.

Gibbs sampler

1. In parallel, sample the ε_{gn}'s.
2. In parallel, sample the γ_g's.
3. Reduction to calculate $\sum_{g=1}^{G}\left[\log\gamma_g + \frac{\nu}{\gamma_g}\right]$. Then sample ν from its full conditional density, which is proportional to
$$\exp\left(-G\log\Gamma\left(\frac{\nu}{2}\right) + \frac{G\nu}{2}\log\left(\frac{\nu\tau}{2}\right) - \frac{\nu}{2}\sum_{g=1}^{G}\left[\log\gamma_g + \frac{\nu}{\gamma_g}\right]\right).$$
4. Reduction to calculate $\sum_{g=1}^{G}\frac{1}{\gamma_g}$. Then sample $\tau \sim \text{Gamma}\left(a + \frac{G\nu}{2},\ \text{rate} = b + \frac{\nu}{2}\sum_{g=1}^{G}\frac{1}{\gamma_g}\right)$.
5. For $\ell = 1, \ldots, L$, in parallel, sample $\beta_{1\ell}, \ldots, \beta_{G\ell}$.
6. Reduction to calculate means and variances of the relevant $\beta_{g\ell}$'s. Then sample $\theta_1, \ldots, \theta_L$.
7. Reduction to calculate the shape and scale parameters of the inverse-gamma distributions. Then sample $\sigma_1, \ldots, \sigma_L$.

In the algorithm above, we highlight the two types of steps: in parallel for the steps with conditionally independent parameters and reduction for the parameters whose full conditionals depend on sufficient quantities calculated from other parameters. In step 5, the $\beta_{g\ell}$'s are conditionally independent across g for a given ℓ, but not necessarily conditionally independent across ℓ, as the conditional independence of the $\beta_{g\ell}$'s depends on the model matrix. In steps 6 and 7, parameter sampling after the parallelized reductions could be parallelized, but the efficiency gain is small if L is small. In our application, L is 5.

Computation time

We studied computation time as a function of the number of genes G and number of samples N. The figure below indicates that, within the range of values we tested, computation time appears to scale linearly with both G and N.

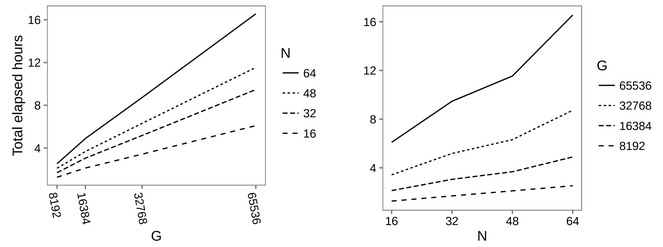

Figure 2: Elapsed runtime (hours) plotted against the number of genes (G) and the number of RNA-seq samples (N) for 2×10^5 total MCMC iterations for four chains run in sequence.

Application to heterosis

Heterosis, or hybrid vigor, is the biological phenomenon in which hybrid progeny surpasses each of its inbred parents with respect to some characteristic. Ever since Dawrin documented heterosis, the term has usually referred to traits at the phenotypic level, and phenotypic heterosis has long been used to enhance crops and livestock. For example, one well-known maize hybrid described by [1] has taller, faster-growing stalks with more grain yield than either inbred parent. Similar breeding techniques have used heterosis to improve rice, alfalfa, tomatoes, and fish. However, the underlying genomic mechanisms of phenotypic heterosis remain unclear [2].

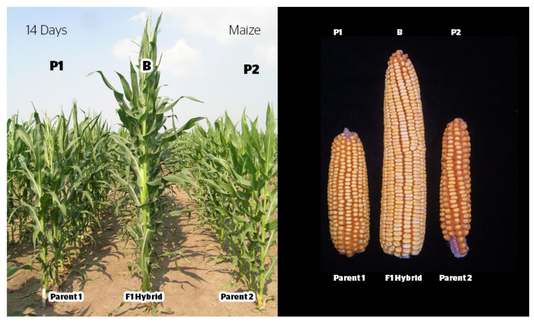

Figure 3: Phenotypic heterosis observed in maize between two parental lines P1 and P2 and their hybrid offspring B.

In our analysis, we used the B73 and Mo17 parental lines with both the B73xMo17 and Mo17xB73 crosses. The model matrix below provides a parameterization where the resulting $\beta_{g\ell}$ are approximately independent (as assumed in our model). The interpretations of the βs on the log scale are β_1 is the parental mean, β_2 is the half difference of hybrid mean vs Mo17, β_3 is the half difference of hybrid mean vs B73, β_4 is the half difference between hybrids, and β_5 is the flow cell block effect.

$$X = \left(\begin{bmatrix} 1 & 1 & -1 & 0 \\ 1 & -1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \otimes J_{(N/4)\times 1} \qquad J_{(N/4)\times 1} \otimes \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix}\right)$$

An example of a hypothesis of interest is high-parent heterosis for the B73xMo17 hybrid, i.e. this hybrid has mean expression level that exceeds the mean for both parents, which is equivalent to the proposition that both $2\beta_{g2} + \beta_{g4} > 0$ and $2\beta_{g3} + \beta_{g4} > 0$.

Execution in R

As an example, we use the data set analyzed in [4].

```
library(fbseq)
data(paschold)  # see https://github.com/jarad/Paschold2012

paschold@contrasts[[5]]
## beta_1 beta_2 beta_3 beta_4 beta_5
##      0      2      0      1      0
paschold@contrasts[[6]]
## beta_1 beta_2 beta_3 beta_4 beta_5
##      0      0      2      1      0
paschold@propositions$`high-parent_B73xMo17`
## high-parent_B73xMo17_1 high-parent_B73xMo17_2
##                      5                      6

configs    = Configs(burnin = 10, iterations = 10, thin = 1)
chain      = Chain(paschold, configs)
chain_list = fbseq(chain, backend = "CUDA")
```

Forthcoming research

Much of this is available on `https://arxiv.org/abs/1606.06659` and the packages themselves are available on Will Landau's github site `https://github.com/wlandau`. Please see Ignacio Alvarez-Castro's poster titled "Fully Bayesian analysis of allele-specific RNA-seq data using a hierarchical, overdispersed, count regression model" for application of this model to allele-specific RNA-seq analysis. Please see Eric Mittman's poster titled "Bayesian nonparametric analysis of RNA-seq data" for relaxing the independent normal assumptions for β_g.

References

[1] Arnel R Hallauer, Marcelo J Carena, and JB Miranda Filho. *Quantitative genetics in maize breeding*, volume 6. Springer, 2010.

[2] Z.B. Lippman and D. Zamir. Heterosis: revisiting the magic. *Trends in Genetics*, 23(2):60–66, 2007.

[3] Radford M. Neal. Slice Sampling. *The Annals of Statistics*, 31(3):705–767, 2003.

[4] Anja Paschold, Yi Jia, Caroline Marcon, Steve Lund, Nick B Larson, Cheng-Ting Yeh, Stephan Ossowski, Christa Lanz, Dan Nettleton, Patrick S Schnable, et al. Complementation contributes to transcriptome complexity in maize (*Zea mays L.*) hybrids relative to their inbred parents. *Genome research*, 22(12):2445–2454, 2012.

Acknowledgements

This research was supported by National Institute of General Medical Sciences (NIGMS) of the National Institutes of Health and the joint National Science Foundation / NIGMS Mathematical Biology Program under award number R01GM109458. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health or the National Science Foundation.