Detecting differential polyadenylation length

Dr. Paul Harrison
paul.harrison@monash.edu
Monash Bioinformatics Platform

Software available from:

rnasystems.erc.monash.edu

Introduction

This poster describes a software package, Fitnoise. Fitnoise generalises differential expression testing to allow testing of other measurements associated with genes or genomic features. In particular our interest is in differential polyadenylation length of mRNA transcripts. Polyadenylation length is of interest as a short poly(A) tail allows mRNA transcripts to be stored or transported within a cell without being immediately translated, including in the early stages of embryo growth and in neurons.

Fitnoise is written in Python, and can also be used from R.

Fitnoise has many similarities to Limma (Smyth, 2004). Hyperparameters are estimated that allow per-feature moderated t tests or F tests to be performed. In Limma, the hyperparameters are the prior degrees of freedom and prior variance (plus some further hyperparameters that allow it to calculate Bayesian posterior odds of differential expression). Fitnoise allows pluggable “noise models” with an arbitrary number of parameters.

In Limma, the two hyperparameters are estimated from their marginal distributions, using sophisticated and fast approximation methods. Fitnoise uses Restricted Maximum Likelihood (REML) to numerically estimate all parameters simultaneously.

Fitnoise can be used as an adjunct to Limma, producing a weights matrix in similar fashion to `voom` for conventional RNA-Seq data (Law, Chen, Shi, & Smyth, 2014). This makes it a pre-processing step to a proven conservative analysis package. Alternatively Fitnoise can be used as a standalone package. Weights from Limma's `voom` function can also be used with Fitnoise.

Multivariate distributions as objects

Fitnoise defines classes of multivariate distributions. These wrap up many mathematical details, allowing the actual operations to be stated straightforwardly. Fitnoise currently defines classes for the multivariate normal distribution, with parameters mean μ and covariance Σ, and for multivariate t distributions, which have in addition a degrees of freedom parameter ν. Further classes may be added if they support the necessary operations: linear transformation, marginal and conditional distribution, expectation, density, and p-value.

The p-value function asks how likely it would be to sample a point from the distribution with probability density less than a given point. For multivariate normal distributions, this turns out to take the form of a chisquare test. For multivariate t distributions this takes the form of an F test, with the two degrees of freedom parameters being the number of dimensions in the distribution and the degrees of freedom of the distribution itself ν (Liu, 1994).

Linear model

We expect the data vector y of each feature to be the sum of a linear model component and a noise component.

$$y = X\beta + \epsilon$$

$$\epsilon \sim \boldsymbol{E}$$

n samples, m-term linear model
X an $n \times m$ design matrix
y an n-vector of observations for the feature
β an m-vector of coefficients to be estimated
ϵ **an n-vector of random noise, sampled from distribution $\boldsymbol{E}$**

Rotation and partitioning of data vectors

For each feature, we rotate the data vector y, obtaining a new vector z which can then be partitioned into a part affected by both the linear model and noise z_1 and a part only affected by noise z_2.

To do this, we first compute the QR-decomposition, a standard decomposition offered by linear algebra libraries. This gives an $n \times n$ orthonormal matrix Q and an $m \times m$ matrix R. Q can be divided into two sets of columns, the first m columns we will refer to as Q_1 and the remaining $(n-m)$ columns we will refer to as Q_2. We have that $X = Q_1 R$. As Q is orthonormal, each column is orthogonal to each other column, and Q_2 is a null matrix of X ($Q_2^T X = 0$). Let $z = Q^T y$, $z_1 = Q_1^T y$ and $z_2 = Q_2^T y$. So

$$z_2 = Q_2^T y = Q_2^T (X\beta + \epsilon) = Q_2^T \epsilon$$

Hyperparameter estimation by REML

We want to choose a noise distribution $\boldsymbol{E}$ for each feature which maximises the likelihood of z_2 as sampled from $Q_2^T \boldsymbol{E}$. This constitutes REML.

$\boldsymbol{E}$ for each feature is a function of a set of hyperparameters and of contextual information available for that feature. In Limma, this contextual information is the weights matrix. Fitnoise allows arbitrary contextual information to be used.

Hyperparameters are numerically optimised to maximise the total over all features of the log likelihood of each z_2 as sampled from its corresponding $Q_2^T \boldsymbol{E}$. Automatic differentiation with the deep-learning library Theano (Bergstra et al., 2010) allows this to be performed efficiently.

This is a quite general scheme. A noise model incorporating sample weights is provided, similar to `arrayWeights` in Limma. If nominated control genes are given a simplified design matrix, unwanted variation may be identified as a set of random effects, similar to RUV-4 (Gagnon-Bartsch, Laurent, & Speed, 2013)**.**

Weights matrix for use with Limma

Limma accepts a weights matrix, which should be proportional to the inverse of the variance of each measurement. If the noise model is independent between measurements, this is easily produced from the fitted $\boldsymbol{E}$ distributions.

Coefficient posterior distribution and significance testing

The posterior distribution of coefficients $\boldsymbol{B}$ can be calculated form z_1 and $\boldsymbol{E}$ conditional on the value of z_2. Conditioning on z_2 yields a noise distribution no longer centred on zero. Some straightforward manipulation produces $\boldsymbol{B} = R^{-1}[z_1 - (Q_1^T \boldsymbol{E} | z_2 \sim Q_2^T \boldsymbol{E})]$, which is easily implemented using the distribution objects described earlier.

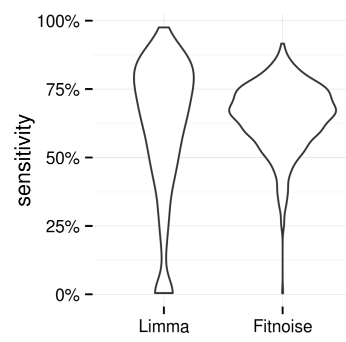
Conditioning $\boldsymbol{E}$ on z_2 when using a multivariate t distribution increases ν by the number of dimensions in z_2 (Liu, 1994). Similarly in Limma `df.total=df.prior+df.residual`.

The expectation of $\boldsymbol{B}$ serves as an estimate of β, and exactly matches weighted least squares estimation (e.g. as produced by Limma). For some given contrast matrix C, the significance level turns out to simply be the p-value of 0 in $C\boldsymbol{B}$ as we defined for distribution objects.

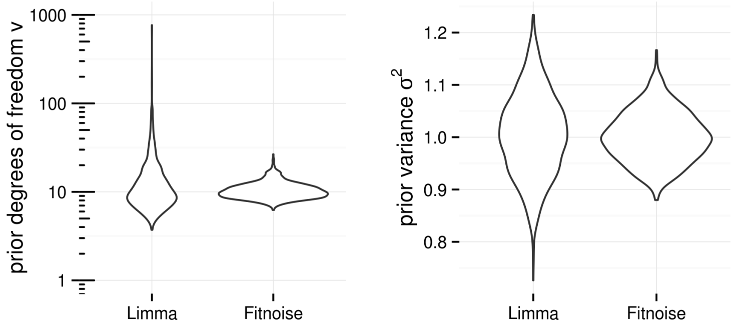
Results with synthetic data

Synthetic data was generated to compare hyperparameters estimated by Fitnoise and Limma. Two groups each with two replicates were used, 1000 features, and 10% of features differential by either -5 or 5. Noise was generated from a multivariate t-distribution with $\nu=10$ degrees of freedom and Σ the identity matrix (this can also be viewed as normally distributed with variance scaled by an inverse chi distribution, simulating variability in variance between genes). 1000 synthetic data sets were generated.

Results are given below as *mean* [*95% interval*].



At a False Discovery Rate of 0.01, overall Fitnoise detects 65% [36%,84%] of actual differences with an actual FDR of 0.010 [0.000,0.039], and Limma detects 61% [0%,92%] of actual differences with an actual FDR of 0.013 [0.000,0.056]. **Fitnoise has less variability between data sets in the number of features declared significant compared to Limma. This is due to the lower variability in hyperparameter estimates of Fitnoise, as shown below.**



The prior degrees of freedom estimate ν is especially variable for Limma. ν is derived from the shape of the distribution, and the two methods perhaps focus on the shape of different parts of the distribution. Clipping the simulated noise consistently produces a lower value of ν from Limma than from Fitnoise (Fitnoise becomes slightly more liberal in terms of actual FDR, Limma becomes more conservative).

PAT-Seq

PAT-Seq is a method for producing high-throughput sequencing reads of polyadenylated RNA developed by Dr. Traude Beilharz (manuscript submitted). PAT-Seq reads contain genomic sequence from just before the polyadenylation site, continuing into the poly(A) tail, then terminating with an adaptor sequence. **PAT-Seq allows:**

- **Polyadenylation site identification, including possibly multiple polyadenylation sites per gene.**
- **Measuring expression levels of polyadenylation sites.**
- **Estimation of poly(A) tail length.**

Bioinformatic analysis of PAT-Seq data can be performed by the Python/R software package Tail Tools, also by the author.

PAT-Seq noise model

PAT-Seq produces a poly(A) tail length for each read with a poly(A) tail.

Tail lengths for a single site in a single sample are variable, due to biological variability. Also, reads are not always long enough to reach the adaptor sequence, so estimates of poly(A) length are underestimates when the tail is long, and a further source of variability.

Considering a single site, call r_i the number of reads with tails in sample i, and y_i the average of the observed tail lengths in sample i.

The contribution to variance from per-read variation is inversely proportional to r_i. Per-sample variance was observed to increase with tail length, so was treated as a coefficient of variation. Hyperparameters to be estimated are per-read standard deviation σ_r, per sample coefficient of variation σ_s, and degrees of freedom ν. We say, for each feature

$$\sigma_i^2 = \frac{\sigma_r^2}{r_i} + \sigma_s^2 y_i^2 \qquad \Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$

Results from a PAT-Seq experiment

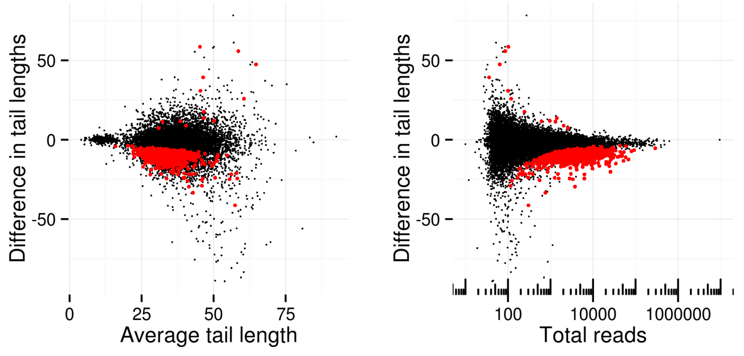
The gld-2 gene extends the poly(A) tail of mRNA during early embryo development, thereby activating it. In the *C. elegans* worm, we compared tail lengths in a mutant with gene gld-2 deactivated against the wildtype. There were three replicates in both groups, with between 7 and 15 million poly(A) reads per sample and 151 bases per read produced by Illumina sequencing. 16,030 polyadenylation sites were identified.

Results are given as *estimate* [*95% CI by bootstrapping, n=1000*].

Fitnoise identifies 931 [850,1021] of these sites as having differential tail length at an FDR of 0.01. Hyperparameters were: ν =12.6 [11.5,13.9], σ_r =28.0 [27.5,28.4], σ_s =0.049 [0.048,0.050]. That is, reads for a site, within a sample, have a standard deviation in tail length of 28 bases, and the average tail length varies between samples in a group by about 5%.

For comparison, using a weight matrix derived from the Fitnoise fit, Limma identifies 493 [424,577] sites at FDR 0.01, with ν =3.1 [3.0,3.3].

All but 13 of the 931 sites identified by Fitnoise have shorter tails in the gld-2 mutant, with a mean change of -10 adenine bases.



We are currently trying to determine why these sites are extended by gld-2 and not others. Comparing differential sites to a set of neutral sites with similar depth of coverage and a difference in length of no more than 2, one difference is that the canonical polyadenylation signal, AAUAAA, is more common upstrand of neutral sites (58% of neutral vs 41% of differential). An alternate form AAUGAA is more common in the differential sites (14% of neutral vs 21% of differential).

Ordering by p-value, the top 10 sites, all of which have a shorter tail in the gld-2 mutant, are for the genes mex-6, mex-5, puf-3, cbd-1, pos-1, C05C10.5, mex-3, oma-1, air-1, and pcn-1. Each of these genes is intimately involved in the regulation and timing of oocyte and early embryo development.

Acknowledgements

Traude Beilharz developed the PAT-Seq method. The *C. elegans* data set was produced in laboratories headed by Traude Beilharz and Peter Boag at Monash University. Thanks to David Powell for extensive discussions on the statistics of tail lengths.

References

Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., … Bengio, Y. (2010). Theano: a CPU and GPU Math Expression Compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*. Austin, TX.
Gagnon-Bartsch, J. A., Laurent, J., & Speed, T. P. (2013). Removing Unwanted Variation from High Dimensional Data with Negative Controls. *Technical Report 820, Berkley Department of Statistics*. Retrieved from http://statistics.berkeley.edu/tech-reports/820
Law, C. W., Chen, Y., Shi, W., & Smyth, G. K. (2014). voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biology*, *15*(2), R29. doi:10.1186/gb-2014-15-2-r29
Liu, C. (1994). *Statistical analysis using the multivariate t distribution*. Harvard University.
Smyth, G. K. (2004). Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments. *Statistical Applications in Genetics and Molecular Biology*, *3*(1). doi:10.2202/1544-6115.1027