

BASiCS workflow: a step-by-step analysis of expression variability using single cell RNA sequencing data

Nils Eling^{*1,2}, Alan O'Callaghan³, John C. Marioni^{1,2}, and Catalina A. Vallejos^{†3,4}

¹European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK

²Cancer Research UK Cambridge Institute, University of Cambridge, Li Ka Shing Centre, Cambridge, CB2 0RE, UK

³MRC Human Genetics Unit, Institute of Genetics & Molecular Medicine, University of Edinburgh, Western General Hospital, Crewe Road, Edinburgh, EH4 2XU, UK

⁴The Alan Turing Institute, British Library, 96 Euston Road, London, NW1 2DB, UK

Abstract Cell-to-cell gene expression variability is an inherent feature of complex biological systems, such as immunity and development. Single-cell RNA sequencing is a powerful tool to quantify this heterogeneity, but it is prone to strong technical noise. In this article, we describe a step-by-step computational workflow which uses the BASiCS Bioconductor package to robustly quantify expression variability within and between known groups of cells (such as experimental conditions or cell types). BASiCS uses an integrated framework for data normalisation, technical noise quantification and downstream analyses, whilst propagating statistical uncertainty across these steps. Within a single seemingly homogeneous cell population, BASiCS can identify highly variable genes that exhibit strong heterogeneity as well as lowly variable genes with stable expression. BASiCS also uses a probabilistic decision rule to identify changes in expression variability between cell populations, whilst avoiding confounding effects related to differences in technical noise or in overall abundance. Using two publicly available datasets, we guide users through a complete pipeline which includes preliminary steps for quality control as well as data exploration using the scater and scran Bioconductor packages. Data for the first case study was generated using the Fluidigm® C1 system, in which extrinsic spike-in RNA molecules were added as a control. The second dataset was generated using a droplet-based system, for which spike-in RNA is not available. This analysis provides an example, in which differential variability testing reveals insights regarding a possible early cell fate commitment process. The workflow is accompanied by a Docker image that ensures the reproducibility of our results.

Keywords

Single-cell RNA sequencing, expression variability, transcriptional noise, differential expression testing

*eling@ebi.ac.uk

†catalina.vallejos@igmm.ed.ac.uk

Quantifying cell-to-cell transcriptional variability - BASiCS

The **BASiCS** Bioconductor package contains an assembly of functions to estimate and analyse gene- and cell-specific model parameters [1, 2, 3]. BASiCS is built upon a hierarchical Bayesian framework and as such samples posterior distribution for each model parameter. In mathematical terms, the gene expression count X_{ij} for gene i in cell j is modelled as:

$$\begin{aligned} X_{ij} | \mu_i, \nu_j &\sim \text{Poisson}(\nu_j \mu_i) \\ \nu_j | s_j, \theta &\sim \text{Gamma}(1/\theta, 1/(s_j \theta)) \\ \rho_{ij} | \delta_i &\sim \text{Gamma}(1/\delta_i, 1/\delta_i) \end{aligned}$$

where μ_i explains the gene's mean expression, ν_j the technical effect characterised by the mRNA capture efficiency s_j and the unexplained technical noise parameter θ . Here, ρ_{ij} the biological random effect fluctuating around the gene-specific over-dispersion hyper-parameter δ_i . It is important to note that, in further analyses, μ_i represents the mean expression and δ_i the biological over-dispersion of each gene.

Due to a strong association between mean expression and biological over-dispersion, the model has been extended to correct for such confounding effect. To this end, the prior distribution for the over-dispersion parameters has been changed to:

$$\delta_i | \mu_i \sim \log\text{-t}_\eta(f(\mu_i), \sigma^2).$$

where $f(\mu_i)$ describes a smooth regression trend between the over-dispersion and the mean expression parameters. This extension introduced the residual over-dispersion parameters $\epsilon_i = \delta_i - f(\mu_i)$ that do not scale with mean expression [3].

From a data analysis perspective, the `BASiCS_MCMC` function is the heart of the BASiCS package, and can be run in four different settings (see Table 1).

Table 1. Four settings that can be used to run the BASiCS_MCMC function.

	No regression	Regression
Using spike-in reads	<code>WithSpikes = TRUE</code>	<code>WithSpikes = TRUE</code>
	<code>Regression = FALSE</code>	<code>Regression = TRUE</code>
No spike-ins available	<code>WithSpikes = FALSE</code>	<code>WithSpikes = FALSE</code>
	<code>Regression = FALSE</code>	<code>Regression = TRUE</code>

If spike-in counts are available and should be used to estimate technical noise, the parameter `WithSpikes` is set to `TRUE` (default). If the regression between over-dispersion and mean expression should be performed, the `Regression` parameter is set to `TRUE` (default). If the user decides to set `Regression = FALSE`, BASiCS will not estimate the regression trend, and will not supply the residual over-dispersion parameters ϵ_i .

The `BASiCS_MCMC` function returns a `BASiCS_Chain` object, which can be used for further downstream analyses, many of which are detailed in this workflow. These objects contain draws from Markov chain Monte Carlo (MCMC) samplers, which are used to infer the posterior distribution over the model parameters [4]. Briefly, the posterior distribution quantifies how probable different parameter values are given the observed data. However, before assessing the posterior distribution, we must first ensure that the MCMC sampler has converged to its stationary distribution, and has sampled efficiently from this distribution [5]. If these conditions are not met, then the estimated parameters may be inaccurate. The **coda** CRAN package contains a variety of functions to assess the convergence of a sampled MCMC chain. To use `coda` functions, the individual chains returned by BASiCS need to be transformed into a MCMC object that `coda` recognises using the `coda::mcmc` function. BASiCS also offers a number of functions to visualise and assess the convergence of MCMC chains. In particular, we will use `BASiCS_EffectiveSize` and `BASiCS_DiagPlot` to calculate and visualise the effective sample size generated by the MCMC samplers.

Other steps [title tbc]

The **goseq** Bioconductor package offers functions to detect the enrichment of gene ontologies (GOs) among user-specified gene sets [6]. Furthermore, **goseq** corrects for gene length biases, which is useful for full length scRNA-seq data as highlighted in the first section. In this workflow, we will use **goseq** to detect GO enrichment among differentially expressed sets of genes.

For downstream analysis, such as GO enrichment analysis or the biological interpretation of individual genes, we need to (i) link each gene's ID to its symbol and (ii) calculate each gene's length. For the first task, the **biomaRt** Bioconductor package annotates a wide range of gene and gene product identifiers [7] by accessing

the BioMart software suite (<http://www.biomart.org>). We can use biomaRt to link the **Mus musculus** gene IDs and to their gene symbols (also referred to as 'gene name'):

References

- [1] Catalina A. Vallejos, John C. Marioni, and Sylvia Richardson. BASiCS: Bayesian analysis of single-cell sequencing data. *PLOS Computational Biology*, 11(6):e1004333, 2015. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1004333. URL <http://dx.plos.org/10.1371/journal.pcbi.1004333>.
- [2] Catalina A Vallejos, Sylvia Richardson, and John C Marioni. Beyond comparisons of means: understanding changes in gene expression at the single-cell level. *Genome Biology*, 17(70), 2016. doi: 10.1101/035949. URL <http://biorxiv.org/content/early/2016/01/05/035949.abstract>.
- [3] Nils Eling, Arianne C. Richard, Sylvia Richardson, John C. Marioni, and Catalina A. Vallejos. Correcting the Mean-Variance Dependency for Differential Variability Testing Using Single-Cell RNA Sequencing Data. *Cell Systems*, 7(3): 1–11, 2018. ISSN 24054712. doi: 10.1016/j.cels.2018.06.011. URL <https://linkinghub.elsevier.com/retrieve/pii/S2405471218302783>.
- [4] A. F. M. Smith and G. O. Roberts. Bayesian Computation Via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 55(17):3–23, September 1993. ISSN 00359246. doi: 10.1111/j.2517-6161.1993.tb01466.x.
- [5] Mary Kathryn Cowles and Bradley P. Carlin. Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review. *Journal of the American Statistical Association*, 91(434):883–904, June 1996. ISSN 0162-1459, 1537-274X. doi: 10.1080/01621459.1996.10476956.
- [6] Matthew D Young, Matthew J Wakefield, Gordon K Smyth, and Alicia Oshlack. Gene ontology analysis for RNA-seq: accounting for selection bias. *Genome Biology*, 11, 2010. URL <http://genomebiology.com/2010/11/2/R14>.
- [7] Steffen Durinck, Yves Moreau, Arek Kasprzyk, Sean Davis, Bart De Moor, Alvis Brazma, and Wolfgang Huber. BioMart and Bioconductor: A powerful link between biological databases and microarray data analysis. *Bioinformatics*, 21(16): 3439–3440, 2005. ISSN 13674803. doi: 10.1093/bioinformatics/bti525.