

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

Nils Eling<sup>\*1,2</sup>, Alan O'Callaghan<sup>3</sup>, John C. Marioni<sup>1,2</sup>, and Catalina A. Vallejos<sup>†3,4</sup>

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

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

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

<sup>4</sup>The 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** package implements a Bayesian hierarchical framework which borrows information across all genes and cells to robustly quantify transcriptional variability [1]. Similar to the approach adopted in **scran**, **BASiCS** infers cell-specific global scaling normalisation parameters. However, instead of inferring these as a pre-processing step, **BASiCS** uses an integrated approach in which data normalisation and downstream analyses are performed simultaneously — whilst propagating statistical uncertainty. To quantify technical noise, the original implementation of **BASiCS** uses information from extrinsic spike-in molecules as control features, but the model has been extended to address situations in which spike-ins are not available [2].

**BASiCS** summarises the distribution of gene expression through gene-specific *mean* ( $\mu_i$ ) and *over-dispersion* ( $\delta_i$ ) parameters. Mean parameters  $\mu_i$  quantify the overall expression for each gene  $i$  across the population of cells under study. Instead,  $\delta_i$  captures the excess of variability that is observed with respect to what would be expected in a homogeneous cell population, after taking into account technical noise. This is used as a proxy to quantify transcriptional variability. Moreover, to account for the strong association that is typically observed between mean expression and over-dispersion estimates, we recently introduced gene-specific *residual over-dispersion* parameters  $\epsilon_i$  [2]. Similar to DM values implemented in **scran**, these are defined as deviations with respect to an overall regression trend that captures the relationship between mean and over-dispersion values.

Parameter estimation is implemented in the **BASiCS\_MCMC** function, which can be run using four different major settings (Table 1). If spike-in molecules are available and `WithSpikes = TRUE` (default), the model uses this information in order to infer technical noise. Alternatively, if `WithSpikes = FALSE`, the approach introduced in [2] is applied. We recommend to use `Regression = TRUE` (default) as the underlying informative prior improves the stability of posterior estimates for small sample sizes and lowly expressed genes [2]. Moreover, it provides additional flexibility for downstream analysis as it also infers gene-specific residual over-dispersion parameters  $\epsilon_i$ .

**Table 1. Four settings available for the 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>

The algorithm implemented in the **BASiCS\_MCMC** function is an adaptive Metropolis within Gibbs sampler [?]. 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 [3]. 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 [4]. 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.

[talk about downstream analyses; hvg/lvg; differential testing then mention the ability to perform differential testing; finally the extension to account for mean/over-dispersion]

## Other steps [title tbc]

The **goseq** Bioconductor package offers functions to detect the enrichment of gene ontologies (GOs) among user-specified gene sets [5]. 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 [6] 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] 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>.
- [3] 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.
- [4] 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.
- [5] 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>.
- [6] 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.