Application Note

pulseR: Versatile computational analysis of RNA turnover from metabolic labeling experiments

Uvarovskii Alexey [1,2]*, Christoph Dieterich [1,2]*

[1] Section of Bioinformatics and Systems Cardiology, Klaus Tschira Institute for Integrative Computational Cardiology, Department of Internal Medicine III, University Hospital Heidelberg, Im Neuenheimer Feld 669, 69120 Heidelberg, and [2] German Center for Cardiovascular Research (DZHK), Im Neuenheimer Feld 669 69120 Heidelberg

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Abstract

Motivation: Metabolic labelling of RNA is a well established and powerful method to estimate RNA synthesis and decay rates. The pulseR R package simplifies the analysis of RNA-seq count data that emerges from corresponding pulse-chase experiments.

Results: The pulseR package provides a flexible interface and readily accommodates numerous different experimental designs. To our knowledge, it is the first publicly available software solution that models count data with the more appropriate negative-binomial model. Moreover, pulseR handles both, labelled and unlabelled, spike-ins in its workflow and accounts for potential labeling biases (e.g. number of uridine residues).

Availability: The pulseR package is freely available at https://github.com/dieterich-lab/pulseR under the GPLv3.0 licence

Contact: a.uvarovskii@uni-heidelberg.de and christoph.dieterich@uni-heidelberg.de

Supplementary information: Supplementary data are available https://goo.gl/xmonsg

Introduction

Gene expression abundance levels are defined by rates of RNA synthesis and degradation. Understanding how certain gene levels are regulated by these processes across different experimental conditions helps to gain deeper insights into RNA control mechanisms. Pulse-chase experiments facilitate measurement of such kinetics (Wachutka and Gagneur, 2016). Generally, "tagged" nucleoside analogs are introduced to the medium, taken up by cells and incorporated into nascent RNA molecules. For example, 4-thiouridine labelling (4sU), which was pioneered by Dölken *et al.* (2008) in this context, is used to estimate kinetic rates of RNA metabolism in an increasing number of studies (see Wachutka and Gagneur (2016) for review). Briefly, RNA labelling facilitates to separate newly synthesized from pre-existing RNA. RNA-seq data, which are generated from sequencing these RNA pools, have a discrete nature. To date, there is no publicly available software for kinetic parameter estimation of gene expression, which is specifically designed to handle fragment count data. Here we present the pulseR package, which allows to process RNA-seq data from 4sU-labelling or similar experiments.

Implementation

Parameter definition

RNA dynamics can be described by ordinary differential equations, which have a simple analytic solution if the degradation and synthesis rates are assumed to be constant. In the pulseR package, users need to specify the expressions for the mean RNA abundances. Alternatively, formulas can be generated using package functions for the most frequent use cases, see package documentation.

Although most interest is focused on the gene-specific parameters, pulseR allows to introduce shared parameters. This can be useful for taking into account the difference in the uridine content, since it can introduce a bias in the estimations (Miller *et al.*, 2011; Schwalb *et al.*, 2012). In this case, the RNA abundances are multiplied by a probability that at least one uridine in the molecule is substituted by 4sU. The shared parameter then is the probability for a single base to be substituted by a 4sU.

Normalization

The pull-down procedure may have an effect on the amount and purity of captured RNA. To account for difference in sequencing depth and pull-down efficiency, we introduce additional normalization factors α_{j*}. For example, if the labelled fraction consists of the labelled RNA L_{ij} and the unlabelled RNA U_{ij} molecules, for a sample j and gene i we have

$$[\text{labelled fraction}]_{ij} = \alpha_{j1} L_{ij} + \alpha_{j2} U_{ij} \qquad (1)$$

In case spike-ins are present, α_{j*} can be directly estimated from spike-in read counts. To this end, the user provides lists of spike-ins, which represent unlabelled or labelled RNA.

In the absence of spike-ins, normalization factors are derived from gene counts because the system is overdetermined. Inside a given RNA-seq group (e.g. [total, labelled, unlabelled] × conditions) samples are normalized for sequencing depth d_j following the DESeq procedure (Anders and Huber, 2010). Normalization between the groups is performed during the fitting procedure, and these coefficients α_{G*} are shared between the samples from the same group G:

$$[\text{labelled fraction}]_{ij} = d_j(\alpha_{L1} L_{ij} + \alpha_{L2} U_{ij}) \qquad (2)$$

Parameter estimation

We use the maximum likelihood method (MLE) to obtain parameter values. A typical RNA-seq experiment estimates gene abundance levels by read counts. It has been previously shown that read counts are well represented by a negative-binomial model, which takes over-dispersion into account (Robinson and Smyth, 2007). The NB distribution has two parameters, the mean m and the dispersion parameter r. Hence, a read number of a gene i in a sample j follows

$$K_{ij} \sim \text{NB}(m_{ij}, r). \qquad (3)$$

We assume r to be shared between all samples and genes. Otherwise it would not be possible to infer all parameters from a small number of replicates (usually, only 2 or 3 points are available).

We separated the fitting procedure into several simpler steps:

1. fitting of gene-specific parameters (e.g. degradation rate)
2. fitting of shared parameters
3. fitting of the normalization factors (for a spike-in-free design)
4. estimation of the dispersion parameter

We repeat the steps 1-4 until user-specified convergence criteria are met. We do not consider gene-gene interactions in this model, which allows us to fit gene-specific parameters independently for every individual gene.

We optimize the likelihood functions by using the L-BFGS-U method (Byrd *et al.*, 1995), which is available in the `stats` R package (R Core Team, 2017).

Discussion

Comparison with existing approaches

All published approaches differ in terms of data normalization, statistical model and level of detail with regard to RNA metabolism. We have compared the following software packages with `pulseR` : DRiLL (Rabani *et al.*, 2014), INSPEcT (De Pretis *et al.*, 2015), DTA (Schwalb *et al.*, 2012), HALO (Friedel *et al.*, 2010). In most cases, samples are normalized by utilizing overdetermination of the system. The normalization coefficients are either estimated via regression (DTA, HALO) or during the MLE procedure together with other parameters (INSPEcT, DRiLL). Additionally, `pulseR` allows to use spike-in counts as an alternative normalization strategy. HALO and DTA estimate degradation rates simply as a ratio of labelled and total RNA fractions. In

DRiLL, expression levels are fitted to the binomial distribution. However, the kinetic rates are estimated via optimization of residual sum of squares in both, DRiLL and INSPEcT. In contrast, `pulseR` assumes the NB distribution in MLE of all parameters, which allows to work directly with count data. Experiments may vary in design and the number of conditions and `pulseR` offers unprecedented flexibility. While DTA and HALO estimate rates on the basis of a single time point only (using all three fractions), `pulseR`, DRiLL and INSPEcT can infer rates from several time points. While pulseR can handle different designs including pulse, chase- or combined experiments, all other packages work only with pulse experiments. However, the DRiLL and INSPEcT packages can model time-dependent rates out of the box. Moreover, the DRiLL software is able to model the gene-transcript dependence structure. Table 1 summarizes all relevant software features. We simulated count data for a pulse experiment

	`pulseR`	DRiLL	INSPEcT	DTA	HALO
statistical model	NB	N, BIN	N	–	–
spike−ins	+	–	–	–	–
several time points	+	+	+	–	–
variable design [1]	+	–	–	–	–
non−constant rates	–	+	+	–	–
uridine bias	∗	–	–	+	+
RNA processing	∗	+	+	–	–
language	R	MATLAB	R	R	Java

Table 1. Comparison of available software for parameter estimation in pulse-chase experiments. +: present, −: not implemented, N: normal, NB: negative binomial, BIN: binomial. ∗ - must be defined by a user. [1] - pulse, chase or combination thereof experiments.

in order to evaluate `pulseR` performance. The detailed protocol of this procedure is described in the supplementary material.

Acknowledgements

References

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome biology*, **11**(10), R106.

Byrd, R. H., Lu, P., *et al.* (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, **16**(5), 1190–1208.

De Pretis, S., Kress, T., *et al.* (2015). INSPEcT: a computational tool to infer mrna synthesis, processing and degradation dynamics from rna-and 4su-seq time course experiments. *Bioinformatics*, **31**(17), 2829–2835.

Dölken, L., Ruzsics, Z., *et al.* (2008). High-resolution gene expression profiling for simultaneous kinetic parameter analysis of rna synthesis and decay. *Rna*, **14**(9), 1959–1972.

Friedel, C. C., Kaufmann, S., *et al.* (2010). HALO – a java framework for precise transcript half-life determination. *Bioinformatics*, **26**(9), 1264–1266.

Miller, C., Schwalb, B., *et al.* (2011). Dynamic transcriptome analysis measures rates of mrna synthesis and decay in yeast. *Molecular systems biology*, **7**(1), 458.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rabani, M., Raychowdhury, R., *et al.* (2014). High-resolution sequencing and modeling identifies distinct dynamic rna regulatory strategies. *Cell*, **159**(7), 1698–1710.

Robinson, M. D. and Smyth, G. K. (2007). Moderated statistical tests for assessing differences in tag abundance. *Bioinformatics*, **23**(21), 2881–2887.

Schwalb, B., Schulz, D., *et al.* (2012). Measurement of genome-wide rna synthesis and decay rates with dynamic transcriptome analysis (dta). *Bioinformatics*, **28**(6), 884–885.

Wachutka, L. and Gagneur, J. (2016). Measures of rna metabolism rates: Toward a definition at the level of single bonds. *Transcription*, page e1257972.