BIOINFORMATICS

Vol. 00 no. 00 2018
Pages 1–3

AnaCoDa: Analyzing Codon Data with Bayesian mixture models

Cedric Landerer [1,2]*, Alexander Cope [3,5] Russell Zaretzki [2,4], and Michael A. Gilchrist [1,2]

[1] Department of Ecology and Evolutionary Biology, University of Tennessee, Knoxville, TN, USA. [2]National Institute for Mathematical and Biological Synthesis. [3]Genome Science and Technology, University of Tennessee, Knoxville, TN, USA. [4]Department of Statistics, Operations, and Management Science, University of Tennessee, Knoxville, TN, USA. [5]Oak Ridge National Laboratory, Oak Ridge, TN, USA.

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

AnaCoDa is an R package for estimating biologically relevant parameters of mixture models, such as selection against translation inefficiency, nonsense error rate, and ribosome pausing time, for sets of genes based on codon or amino acid data from genomic and high throughput datasets. **AnaCoDa** provides an adaptive Bayesian MCMC algorithm, fully implemented in C++ for high performance with an ergonomic R interface to improve usability. **AnaCoDa** employs a generic object-oriented design to allow users to extend the framework and implement their own models for analyzing biological data. Current models implemented in **AnaCoDa** can accurately estimate biologically relevant parameters given either protein coding sequences or ribosome foot-printing data. Optionally, **AnaCoDa** can utilize additional data sources, such as gene expression measurements, to improve aid model fitting and parameter estimation. By utilizing a hierarchical object structure, some parameters can vary between sets of genes while others can be shared. Gene membership in a set can either be pre-assigned or Genes may be assigned to clusters or membership may be estimated by **AnaCoDa**. This flexibility allows users to estimate the same model parameter under different biological conditions and categorize genes into different sets based on shared model properties embedded within the data. **AnaCoDa** also allows users to generate simulated data which can be used to aid model development and model analysis as well as evaluate model adequacy. Finally, **AnaCoDa** also comes with contains a set of visualization routines and the ability to revisit or reinitiate previous model fitting, providing researchers with a well rounded easy to use framework to analyze genome scale data.

Availability: **AnaCoDa** is freely available under the Mozilla Public License 2.0 on CRAN (`https://cran.r-project.org/web/packages/AnaCoDa/`).

Contact: cedric.landerer@gmail.com

*to whom correspondence should be addressed

INTRODUCTION

The exponential increase in publicly available genomes over the past decade and the addition of novel technologies produced a vast amount of data for researchers. This influx of raw data necessitates the development of computational tools for extracting biological information. Such tools and models have to be developed and provided in easy to use software to allow researchers to analyze classical sequence data as well as novel data like ribosome footprinting counts. Here, we describe **AnaCoDa** is an open-source software implemented in R (R Core Team, 2015) that allows researchers to analyze genome-scale data like coding sequences and ribosome footprinting data using evolutionary or analytical models in a Bayesian framework. **AnaCoDa** was developed to analyze selection on synonymous codon usage in the form of ribosome overhead cost (Gilchrist *et al.* (2015); Wallace *et al.* (2013); Shah and Gilchrist (2011)). However, other codon metrics like the codon adaptation index (Sharp, 1987) or the effective number of codons (Wright, 1990) are also provided as reference. Models described by Gilchrist *et al.* (2015), Wallace *et al.* (2013), and Shah and Gilchrist (2011) can be effectively fitted using **AnaCoDa**. In addition, three currently unpublished models to analyze coding sequences for evidence of selection against nonsense errors and estimate ribosome pausing times from ribosome footprinting data are included. **AnaCoDa** implements an adaptive Gibbs sampler within a Metropolis-Hastings Monte Carlo Markov Chain (MCMC) approach. This allows for the incorporation of prior knowledge such as observed gene expression levels and easy sampling from the posterior distribution to estimate parameter values and quantify degree of uncertainty in these estimates. Currently, **AnaCoDa** provides four models to analyze codon counts obtained from coding sequences or ribosome foot-printing experiments. However, **AnaCoDa** provides a modular infrastructure such that additional genome scale or even phylogenetic models can be integrated.

AnaCoDa provides a mixture distribution option to all implemented models, combining genes into sets by estimating the posterior probabilities of set membership based on gene-set specific parameters shared by all genes assigned to a given set. **AnaCoDa** provides a generic, mixture distribution option to all implemented

models, allowing for the estimation of condition specific parameters or the automatic categorization of data into different sets based on differences in their posterior probabilities of set membership.

In addition to the four models provided, **AnaCoDa** provides a modular infrastructure such that additional genome scale or even phylogenetic models can be integrated.

The **AnaCoDa** framework works with **AnaCoDa** requires gene specific data such as codon frequencies obtained from coding sequences or position specific footprint counts. Conceptually, **AnaCoDa** uses allows for three different types of parameters. The first type of parameters are gene specific parameters such as gene expression protein synthesis rate or relative functionality. Gene-specific parameters are estimated separately for each gene and can vary between potential gene categories or sets. The second type of parameters are gene-set specific parameters, such as mutation bias terms or translation error rates. These parameters are shared across genes within a set and can be exclusive to a single set or shared with other sets. While the number of gene sets must be pre-defined by the user, set assignment of genes can be pre-defined or estimated as part of the model fitting. Estimation of the set assignment provides the probability of a gene being assigned to a set allowing the user to asses the uncertainty in each assignment. The third type of parameters are hyperparameters allowing for the construction and analysis of hierarchical model. Hyperparameters , such as parameters controlling the prior distribution for gene and gene-set specific parameters such as mutation bias or error rate protein synthesis rate. Hyperparameterscan be set specific or shared across multiple sets and allow for the construction and analysis of hierarchical models by controlling prior distributions for gene or gene-set specific parameters. In order reduce the effect of the 'curse of dimensionality' on the sampling efficiency of the MCMC chain and allow flexibility in parallelization, **AnaCoDa** uses an adaptive Gibbs sampling approach where the MCMC sampling of one parameter type is conditioned on the other two types.

FEATURES

AnaCoDa provides an interface written in R, a freely available programming language noted for its ease of use for even inexperienced programmers. As a result, **AnaCoDa** is accessible to researchers with minimal computational experience.

The **AnaCoDa** interface is designed for quick and efficient data analysis. The interface of **AnaCoDa** is designed for quick and efficient data analysis. Generally, the only input needed for fitting a model to the data are protein-coding nucleotide codon sequences in the form of a FASTA file or a flat-file containing codon counts obtained from ribosome foot-printing experiments. If available, users may also provide additional types of data such as estimates of gene expression. **AnaCoDa** can simultaneously utilize the information embedded in these additional data types and/or estimate the error associated with it. **AnaCoDa** also provides visualization functionality, including plotting functions to compare parameter estimates for different mixture distributions and display codon usage patterns. In addition, diagnostic functions such as those for calculating and visualizing the degree of autocorrelation in MCMC samples the parameter traces are provided.

Robust and efficient model fitting **AnaCoDa** has built-in features designed to improve the robustness and performance of the implemented MCMC approach. For example, the implemented MCMC approach automatically adapts the proposal width for sampled parameters so such that an user defined acceptance rate range is met, improving sampling efficiency of the MCMC and computational performance. Even though **AnaCoDa** is written in C++, analysis of large datasets and/or complex models can be very computationally intensive. In order To protect users from computer failures or aid in the collection of additional MCMC samples, **AnaCoDa** can periodically produce output checkpoint files, which can be used to restart an MCMC chain from a previous time point. In addition, **AnaCoDa** is capable of thinning the MCMC chain, automatically thins all parameter traces - meaning only every k^{th} sample is kept - increasing the effective number of samples and reducing its memory foorprint. Thinning increases the effective number of samples by reducing the auto-correlation between samples and reduces the amount of memory required by the underlying data structures. **AnaCoDa** is also able to create file R compatible representations of its parameter and MCMC objects. These objects can be loaded into the R environment for model analysis and visualization.

Although **AnaCoDa** is provided as an R package, the main computational work is implemented in C++. Because R does not provide native C++ support, we used the R package Rcpp which allows for the exposure of Rcpp was employed to expose whole C++ classes as modules to R (Edelbuettel and Francois, 2011). Using Rcpp eliminates time consuming data transfers between the R environment and the C++ core during runs model fitting, resulting in improved computational performance and allows for a fully object-oriented code design (Booch, 1993). As expected, the runtime of **AnaCoDa** scales linearly with genome size and number of iterations, and scales polynomially with the number of mixture distributions in the data set. The polynomial increase in the number of mixture distributions is explained by the necessity to estimate the protein production rate for each gene in each mixture distribution, as it is a gene specific parameter and the probability of a gene being assigned to a mixture has to be conditioned on it. The polynomial increase in runtime with the number of mixture distributions is due to the necessity to condition the gene assignment on the estimation of gene specific parameters, such as, protein synthesis rate.

Data Simulation In addition to model fitting to actual to fitting the models to datasets, **AnaCoDa** can be used to generate simulated data sets as well. On their own, simulated datasets are useful for model development and analysis. Simulating data under different conditions allows the user to explore model behavior and explore theoretical scenarios. Different conditions can include the addition or elimination of parameters, or simply allowing a set of parameter values to vary. Fitting models to simulated data can provide users insight into potential pitfalls or shortcomings when fitting observational data and can serve as the basis for evaluating model adequacy of a model fit to observational data (Mi *et al.*, 2015). Significant differences between simulated and observational data suggests the current set of parameters or the model as a whole fail to include or adequately represent biological mechanisms underlying the observational observed data.

Available models **AnaCoDa** currently provides codon models for analyzing genome scale data. The ROC model implements and

extends the codon usage bias (CUB) models developed by Gilchrist *et al.* (2015); Wallace *et al.* (2013); Shah and Gilchrist (2011), which can reliably estimate the strength of selection on ribosome overhead cost, mutation bias and allows for the inference of protein synthesis rates. This model allows for the separation of effects of mutation and selection based on gene ordering by protein synthesis rate, and the added addition of a mixture distribution allows for gene clustering based on these effects mutation bias and selection for translation efficiency. In addition to identifying the most efficient codons, ROC provides information on the direction estimates of mutation bias allowing the approximation of mutation rates ratios between codons (Gilchrist *et al.*, 2015; Wallace *et al.*, 2013). The ability to estimate protein synthesis rates in the absence of empirical data is useful for investigating CUB of non-model organisms for which such data is lacking and enables the usage of protein synthesis rate in comparative frameworks or other analyses requiring protein synthesis rate information (Dunn *et al.*, 2018). Use of the mixture model allows for the investigation of CUB heterogeneity at the genome or gene level. Following the same framework, additional models included in **AnaCoDa** provide estimates of codon-specific nonsense errors rates (FONSE) and ribosome pausing times (PA and PANSE). Furthermore, **AnaCoDa** implements a ribosome Pausing (PA) model to estimate codon specific ribosome pausing times from ribosome foot-printing data. Parameters estimated with the evolutionary models ROC and FONSE represent evolutionary averages and do not depend on experimental conditions. In contrast, PA and PANSE estimate the distribution of biologically relevant parameters like ribosome pausing times along a gene from experimental data such as ribosome footprinting data. The distribution can be dependent (PANSE) or independent (PA) of evidence for nonsense errors in the data.

REFERENCES

Booch, G. (1993). *Object-oriented analysis and design with applications*. Benjamin-Cummings Publishing Co, Redwood City.

Dunn, C., Zapata, F., Munro, C., Siebert, S., and Hejnol, A. (2018). Pairwise comparisons across species are problematic when analyzing functional genomic data. *Proc Natl Acad Sci USA*.

Edelbuettel, D. and Francois, R. (2011). Rcpp: Seamless r and c++ integration. *Journal of Statistical Software*, **40**, 1–18.

Gilchrist, M., Chen, W., Shah, P., Landerer, C., and Zaretzki, R. (2015). Estimating gene expression and codon-specific translational efficiencies, mutation biases, and selection coefficients from genomic data alone. *Genome Biology and Evolution*, **7**, 1559–1579.

Mi, G., Di, Y., and Schafer, D. (2015). Goodness-of-fit tests and model diagnostics for negative binomial regression of rna sequenceing data. *PLOS ONE*, **10**, e0119254.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Shah, P. and Gilchrist, M. (2011). Explaining complex codon usage patterns with selection for translational efficiency, mutation bias, and genetic drift. *Proc Natl Acad Sci USA*, **108**, 10231–6.

Sharp, P. (1987). The codon adaptatoin index - a meassure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Research*, **15**, 1281–1295.

Wallace, E., Airoldi, E., and Drummond, D. (2013). Estimating selection on synonymous codon usage from noisy experimental data. *Molecular Biology and Evolution*, **30**, 1438–1453.

Wright, F. (1990). The 'effective number of codons' used in a gene. *Gene*, **87**, 23–29.