GReNaDIne: Data-Driven Approaches to Infer Gene Regulatory Networks in Python

Sergio Peignier, Pauline Schmitt, Federica Calevro

GReNaDIne: Data-Driven Approaches to Infer Gene Regulatory Networks in Python

Sergio Peignier*, Pauline Schmitt and Federica Calevro
Univ Lyon, INSA-Lyon, INRAE, BF2i, UMR0203, F-69621, Villeurbanne, France

Abstract

Summary: GReNaDIne (Gene Regulatory Network Data-driven Inference) is a Python package that implements 18 Machine Learning data-driven gene regulatory network inference methods. It includes 8 generalist pre-processing techniques, suitable for RNAseq and MicroArray datasets analysis, as well as 4 RNAseq normalization techniques. This package has been successfully assessed under the DREAM5 challenge benchmark dataset.

Availability and implementation: The open source GReNaDIne Python package is freely available at https://gitlab.com/bf2i/grenadine as well as its latest documentation https://grenadine.readthedocs.io/en/latest/

Key words: Python package, Machine Learning, GRN Inference, Gene Regulatory Networks

1 Introduction

Inferring Gene Regulatory Networks (GRN) from gene expression data is a challenging problem for the system biology community, and plethora of methods have been proposed so far to address it (Aibar *et al.,* 2017; Glass *et al.*, 2015; Sanguinetti and Huynh-Thu 2019). GRN inference approaches based on the data-driven paradigm are among the most popular ones, due to their simplicity, computational efficiency and accuracy (Sanguinetti and Huynh-Thu, 2019). Such approaches aim at using experimental gene expression data to score potential regulatory links between transcription factors (TFs) and target genes (TGs). Some of them assume that the regulatory interaction between TFs and TGs can be measured using correlation statistics (Zhang and Horvath 2005) or information theory measures such as Mutual Information (MI) (Faith *et al*., 2007). Finally, other techniques use a feature importance scoring procedure, training regression or classification algorithms to predict the expressions of TGs from those of TFs (Haury *et al*., 2012; Irrthum *et al*., 2010; Peignier *et a*l., 2019). Here we introduce GReNaDIne, a Python package implementing 18 data-driven GRN inference methods, 8 generalist pre-processing techniques for RNAseq and microarray datasets and 5 RNAseq normalization techniques.

2 Program Overview

GReNaDIne consists of three separate modules (Fig. 1) allowing preprocessing gene expression data, scoring potential regulatory links with data-driven approaches, selecting the most promising links to generate GRNs and evaluating the resulting GRNs. GReNaDIne is implemented as a library for Python 3 and relies on widely used libraries: Scikit-learn (Pedregosa *et al*., 2011), NumPy (Oliphant 2006), Pandas (McKinney 2010) and SciPy (Virtanen 2020).

2.1 Module 1: preprocessing

As a preliminary step, the first GReNaDIne module, aims at normalizing and standardizing the datasets: classic RNAseq normalization techniques are included to cope with library size biases (Reads Per Million (RPM)), gene length biases (Reads Per Kilobase (RPK)), or both problems simultaneously (Reads Per Kilobase Million (RPKM) and Transcripts Per Kilobase Million (TPM)). In addition, GReNaDIne includes a wrapper to use DESeq2 (Love, *et al*., 2014) from Python. GReNaDIne also includes five discretization techniques for gene expression data: Equal Frequency Discretization (EFD), Equal Width Discretization (EWD), Kmeans Discretization applied by rows, columns, and the bidirectional Kmeans method (Bi-Kmeans) (Jung, *et al*. 2015). Finally, GReNaDIne incorporates three standardization methods based on z-scores, namely row/column-wise z-score and polishing standardization methods (Olshen and Rajaratnam, 2010).

2.2 Module 2: GRN Inference

Data-driven GRN inference methods score all possible regulatory links between TFs and TGs, based on their gene expression. Traditional GRN inference methods assume that the regulatory relationships between TFs and TGs can be inferred by measuring the correlation/MI between their respective gene expression levels: GReNaDIne includes four methods based on the widely used Pearson, Spearman, Kendall tau and MI statistics.

Inference methods based on classification and regression aim at training a model (respectively a classifier or a regressor) to predict the expression level of TGs from those of a set of TFs. Then, the importance of each TF to the prediction task is computed, as a feature importance score. These scores are directly used as proxies to score the regulatory relation between each TF and its TGs. Regressors are directly trained on continuous gene expression data, while classifiers require the TG expression to be previously discretized. GReNaDIne includes two methods based on Support Vector Machines (SVM) classifiers (C) and regressors (R), as described in Peignier *et al*., 2019. It incorporates eight methods based on decision tree regressors and classifiers: AdaBoost (AB), Gradient Boosting (GB), Random Forest (RF) and eXtreme Randomized Trees (XRT) (Friedman, 2002; Huynh-Thu *et al*., 2010; Peignier *et al*., 2019). GReNaDIne also includes two methods based on regression stability selection criteria (Haury *et al.* 2011) and two novel methods based on Bayesian Ridge Regression or Complement Naive Bayes classification

2.3 Module 3: links selection and evaluation

After scoring all possible regulatory links between TFs and TGs, a classic procedure consists in selecting a subset of regulatory links with high scores to define putative GRNs. GReNaDIne includes functions that allow ranking the possible regulatory links according to their scores, as well as functions that select the top-k links of the dataset as well as the top-k links involving a particular TF or TG. Finally, the third GReNaDIne module includes some methods computing standard evaluation measures for binary classification to assess predicted GRNs, when gold standard datasets describing validated regulatory links are available. The methods implemented in this module are inspired on those described and used by Marbach *et al*. (2012), in their evaluation framework.

3 Evaluation and Conclusion

3.1 Evaluation protocol

The DREAM5 evaluation framework (Marbach *et al*., 2012) was used to assess the GReNADIne performances. This framework relies on three gold standard datasets from living organisms, namely *Escherichia coli*, *Saccharomyces cerevisiae* and *Staphylococcus aureus,* and a synthetic gold standard dataset. The GRN inference task has been evaluated as a binary classification task, which consisted in predicting the presence of true regulatory links from gene expression. The task was assessed using area under the precision recall curve (AUPR) (Davis and Goadrich, 2006) and area under the receiver operating characteristic curve (AUROC) (Fawcett, 2006).

3.2 Results

The different methods included in GReNaDIne provide results that are comparable or outperform those obtained by the 35 DREAM5 competitors, as well as those obtained by the robust community approach that combined all the participants' results (Marbach *et al*., 2012) (Supp. Fig.1. and 2). The inference methods based on ensembles of decision trees are comparable to the community, for most datasets. The new approaches introduced in GReNaDIne based on SVMs, Bayesian Ridge Score, and Complement Naive Bayes, lead to important gains for real organisms' datasets, but exhibit a quality loss for the synthetic dataset. Methods based on correlation/MI lead to comparable results for the real organisms' datasets, while exhibiting poorer results for the synthetic dataset. In average, the best preprocessing techniques are those ensuring that genes have comparable levels of expression, i.e., row z-score, EFD, and row k-means (Supp. Fig. 3 and 4). These positive results support the value of GReNaDIne for the data-driven GRN inference community.

4 References

Aibar,S. *et al.* (2017) Scenic: single-cell regulatory network inference and clusting. Nat. Methods, 14(11), 1083.

Davis,J. and Goadrich,M. (2006), The relationship between precision-recall and roc curves in Proceedings of the 23rd international conference on Machine learning. ACM, pp. 233–240.

Faith,J.J. *et al*. (2007) Large-scale mapping and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. PLoS Biol., 5(1), e8.

Fawcett,T., (2006), An introduction to roc analysis. Pattern recognition letters, 27(8), 861–874.

Glass,K. *et al.* (2015) High performance computing of gene regulatory networks using a message-passing model. In 2015 IEEE High Performance Extreme Computing Conference (HPEC), pages 1–6. IEEE.

Friedman,J.H. (2002). Stochastic gradient boosting. Comput. Stat. Data Anal., 38(4), 367-378.

Glass,K. *et al*. (2015) High performance computing of gene regulatory networks using a message-passing model. In 2015 IEEE High Performance Extreme Computing Conference (HPEC), pages 1–6. IEEE.

Haury,A.-C. *et al*. (2012) Tigress: trustful inference of gene regulation using stability selection. BMC Syst. Biol., 6(1), 145.

Huynh-Thu,VA. *et al*. (2010) Inferring Regulatory Networks from expression data Using tree-based methods. PLoS ONE, 5(9).

Irrthum,A. *et al.* (2010) Inferring regulatory networks from expression data using tree-based methods. PloS One, 5(9), e12776.

Jones,E. *et al.* (2001) SciPy: Open source scientific tools for Python. [Online; accessed 20-06-2019].

Jung, S. et al. (2015) Evaluation of data discretization methods to derive platform independent isoform expression signatures for multi-class tumor subtyping. BMC genomics, 16(11), S3.

Love,M.I. *et al.* (2014). Moderated estimation of fold change and dispersion for RNA-seq data with deseq2. Genome Biol., 15(12):550.

Marbach,D, *et al.* (2012), Wisdom of crowds for robust gene network inference. Nat. Methods, 9(8), 796.

McKinney.W. (2010) Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference, pages 51-56.

Oliphant.T. (2006). A guide to NumPy, USA: Trelgol Publishing.

Olshen,R.A. and Rajaratnam,B. (2010), Successive normalization of rectangular arrays. Ann. Stat., 38(3), 1638.

Pedregosa,F. *et al.* (2011) Scikit-learn: Machine learning in python. JMLR, 2825–2830.

Peignier,S. *et al.* (2019) Data-driven gene regulatory network inference based on classification algorithms. In 31st International Conference on Tools with Artificial Intelligence, pages 1–8. IEEE.

Sanguinetti,G. and Huynh-Thu,V.A. (2019) Gene regulatory network inference: an introductory survey. In Gene Regulatory Networks, pages 1–23. Springer.

Virtanen,P., *et al* (2020) SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, in press.

Zhang,B. and Horvath,S. (2005) A general framework for weighted gene co-expression network analysis. Stat. Appl. Genet. Mol. Biol., 4(1).

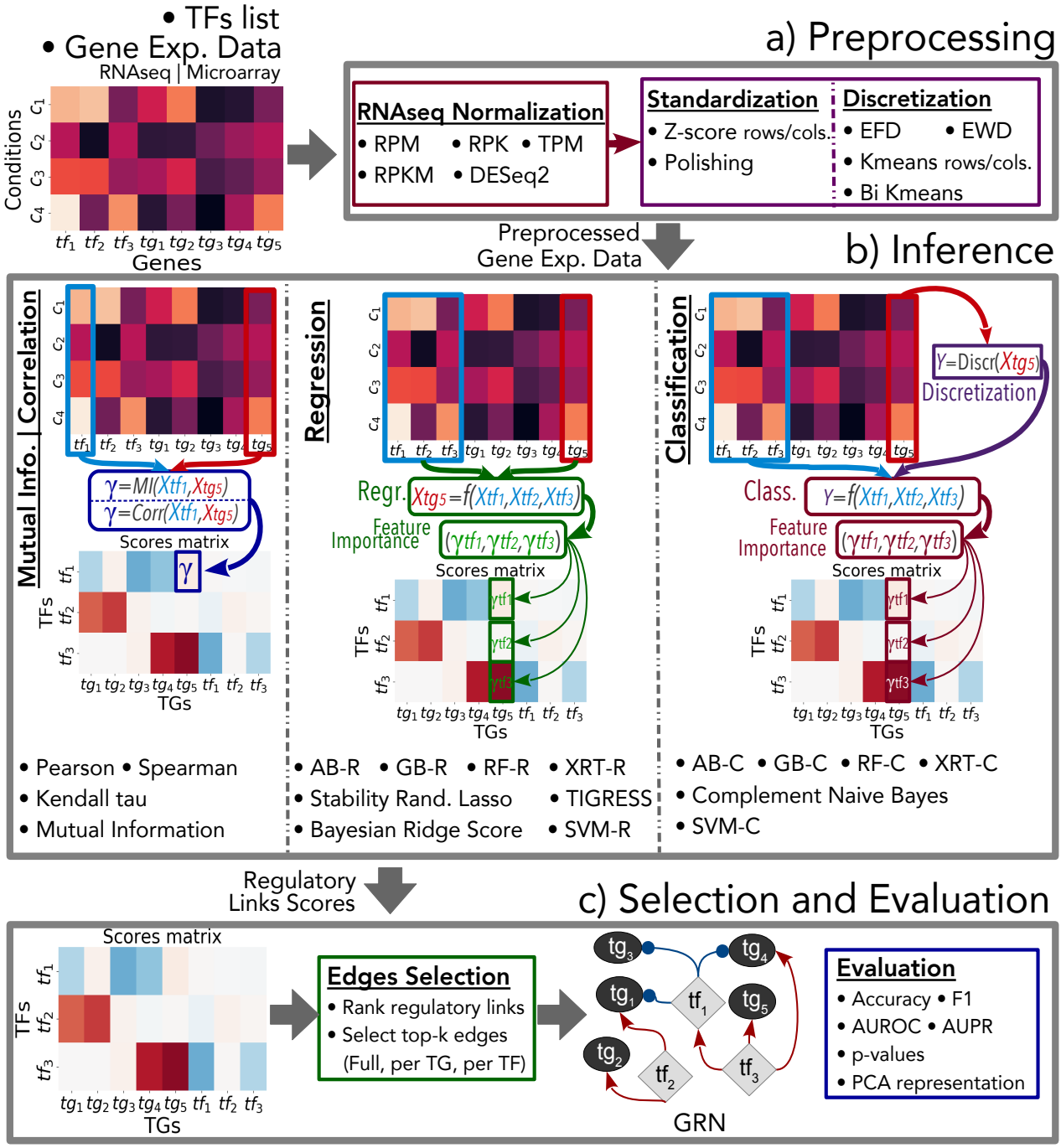

Figure 1 GReNaDIne GRN Inference workflow organized in three modules: a) gene expression preprocessing, including RNAseq normalization, standardization and discretization b) GRN data-driven inference scoring methods and c) regulatory edges selection and GRN evaluation