Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Systems biology

RANKS: a flexible tool for node label ranking and classification in biological networks.

Giorgio Valentini [1],*, Giuliano Armano [2], Marco Frasca [1], Jianyi Lin [1], Marco Mesiti [1] and Matteo Re [1]

[1] AnacletoLab, Department of Computer Science, University of Milan, 20135 Milan, Italy and
[2] Department of Electric and Electronic Engineering, University of Cagliari, 09123 Cagliari, Italy.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Summary: *RANKS* is a flexible software package that can be easily applied to any bioinformatics task formalisable as ranking of nodes with respect to a property given as a label, such as automated protein function prediction, gene disease prioritization and drug repositioning. To this end *RANKS* provides an efficient and easy-to-use implementation of kernelized score functions, a semi-supervised algorithmic scheme embedding both local and global learning strategies for the analysis of biomolecular networks. To facilitate comparative assessment, baseline network-based methods, e.g. label propagation and random walk algorithms, have also been implemented.
Availability and implementation: The package is available from CRAN: https://cran.r-project.org/. The package is written in R, except for the most computationally intensive functionalities which are implemented in C.
Contact: valentini@di.unimi.it
Supplementary information: Supplementary Information are available at *Bioinformatics* online.

1 Introduction

Relevant bioinformatics problems can be modeled through networks, where nodes represent biomolecular entities (e.g. proteins or genes) and edges functional relationships between them. In this context a typical class of problems is node label ranking, which consists of ordering nodes with respect to a given property under study –e.g. the annotation with a specific Gene Ontology (GO) or Online Mendelian Inheritance in Man (OMIM) term. Examples of these problems are represented by protein function prediction, disease gene prioritization and drug repositioning.

Several software tools have been recently developed for the analysis of biomolecular networks. The *BioNet* R package (Beisser *et al.*, 2010) provides a set of methods for the integrated analysis of gene expression data and biological networks. HTSanalyzeR (Wang *et al.*, 2011) is a tool optimized for network analysis of High-throughput screens, while SVD-Phy predicts functional associations between non-homologous genes by comparing their phylogenetic distributions Franceschini *et al.*, 2016. GeneRev (Zheng *et al.*, 2012) aims at assessing the functional relevance of genes from high-throughput data. SANTA (Cornish *et al.*, 2014) uses spatial statistics techniques to assess the functional information content of a biological network with respect to a given set of seed genes. The GeneNet Toolbox for Matlab (Taylor *et al.*, 2015) can evaluate the relevance of functional relationships by performing a statistical assessment of gene connectivity using seed nodes, network randomization and permutation techniques.

The aforementioned tools are limited in their application by the usage of a specific source of data (as in the case of BioNet and HTSanalyzeR) or do not allow to integrate custom methods in the analysis workflow (as in the case of GeneRev). Other software tools can use different sources of data but are devised to evaluate the relevance of functional relationships, as in the case of GeneNet, and cannot be obviously used to predict functional labels for the nodes of the network.

To provide a data source-independent bioinformatics tool for solving arbitrary node label ranking and classification problems in biological networks, we devised *RANKS* (RAnking of Nodes with Kernelized Score functions), a flexible algorithmic scheme implemented and distributed as an R software package. *RANKS* embeds *kernelized score functions* that have been successfully applied to gene function prediction (Re *et al.*, 2012), gene disease prioritization (Valentini *et al.*, 2014) and drug repositioning (Re and Valentini, 2013) problems. Other popular network-based algorithms, such as random walk and label propagation, are also provided by the package.

2 Algorithmic framework

RANKS takes as input the adjacency matrix of the graph representing a functional or genetic network, e.g. a coexpression or a protein-protein interaction (PPI) network, and a set of “seed” nodes having the specific biologic property under study, e.g. proteins with a specific GO annotation or drugs having a specific therapeutic indication. Then it tries to propagate this property from seeds to the other nodes, according to a semi-supervised learning strategy which relies on similarity between nodes. In fact, *RANKS* can extend the concept of similarity between nodes, embedded for instance in a PPI network, by applying a kernel (e.g. a graph kernel) to a specific network to compare a node with others according to the topology of the underlying graph (Fig. 1).

In the most general case a kernel (e.g. a random walk kernel) accomplishes a global learning strategy by exploiting the overall topology of the network. If no kernels are applied, *RANKS* adopts a local learning strategy similar to that of classical guilt-by-association methods: i.e. each node learns only from its neighborhood. Finally, nodes are ranked with a specific scoring function, such as the nearest-neighbour or average score (Fig. 1), according to the weights of the edges and to the annotations of the neighborhood nodes (see Supplementary Information for more details).

RANKS implements a modular algorithmic scheme: by choosing different scoring functions or different kernels one may obtain different semi-supervised learning algorithms applicable to a large range of node label ranking problems in bioinformatics. Moreover the low computational complexity of the underlying semi-supervised learning algorithms allows fast and efficient ranking of nodes also in large networks: once the kernel is computed, the complexity is linear in the number of nodes in sparse networks.

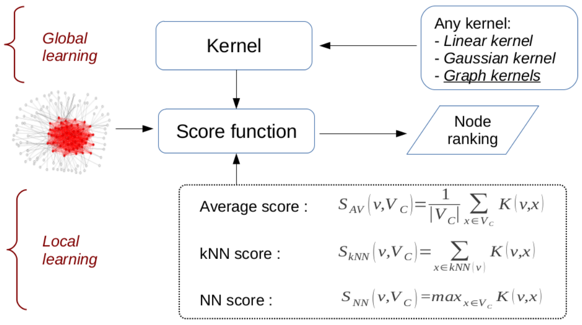

Fig. 1. The RANKS algorithmic framework adopts both local and global learning strategies. Kernels enforce a global learning strategy by extending the notion of similarity between nodes beyond the simple concept of connectivity between adjacent nodes, while score functions adopt local learning strategies by considering only the direct neighborhood of a given node v. V_C denotes the subset of annotated nodes, $K(v, x)$ a kernel function defined on nodes v and x, and $kNN(v)$ the k nodes most similar to a node v.

3 Implementation

The top-level algorithmic scheme is written in R, but the most computationally demanding parts (e.g. the implementation of kernels) are written in C language and invoked from R code as `.C` calls. The user-friendly software interface of *RANKS* allows to independently select different kernels (e.g. linear, Cauchy, random walk kernels) and score functions (e.g. Nearest-Neighbour, Average Sum scores), simply by passing them as parameters to the methods and functions implemented in the package.

Moreover, the user can easily add her/his own kernels, score functions or both to extend the algorithmic scheme (the Supplementary Information shows several examples on how to extend the library). The package provides four main categories of methods:

1. Methods to implement score functions, including k-Nearest-Neighbour, Average Score and Weighted Sum with Linear Decay, the latter being a score function implemented in AraNet (Lee *et al.*, 2010).
2. Methods to implement kernels, including linear, Gaussian and graph-specific kernels able to exploit the overall topology of a network.
3. Methods to automatically apply score functions and kernels to each node of the network: a score is assigned to each node of a network, according to the property under study (e.g. the annotation to a GO or a OMIM term).
4. High-level methods aimed at evaluating the generalization capability of the learning system. These methods include fully automated k-fold cross-validation, held-out or multiple held-out assessment of the generalization error. By a single call to these very high level functions an entire cross-validation cycle can be performed by writing few lines of R code. In particular a leave-one-out (loo) procedure can be efficiently performed at the cost of a single “pass” on the network, without the need of repeating the learning process for each node.

The Supplementary Information and the Reference Manual show several usage examples, which explain how to apply *RANKS* to relevant problems in bioinformatics. For instance, the Supplementary Information provides an example concerning the Human Phenotype Ontology prediction, a ranking task where *RANKS* resulted among the top methods in the recent CAFA2 challenge (Jiang *et al.*, 2016).

4 Conclusion

The *RANKS* learning framework is well suited to perform functional prediction experiments on the whole genome, as its semi-supervised learning strategy allows to efficiently infer node labels in large networks, starting from a small set of annotated examples. The highly modular structure of the functions and methods available in the corresponding R package allows users to easily experiment with different learning algorithms by using a rich collection of interchangeable building blocks. Notably, the library can be extended through user-defined kernels and score functions, and can be easily used as a stand-alone tool or within software pipelines aimed at ranking/classifying node labels in complex biological networks.

References

Beisser, D. *et al.* (2010). BioNet: an R-Package for the functional analysis of biological networks. *Bioinformatics*, **26**(8), 1129–1130.

Cornish, A. *et al.* (2014). Santa: Quantifying the functional content of molecular networks. *Plos Comp. Bio.*, **10**(9), e1003808.

Franceschini, A. *et al.* (2016). SVD-phy: improved prediction of protein functional associations through singular value decomposition of phylogenetic profiles. *Bioinformatics*, **32**(7), 1085–1087.

Jiang, Y. *et al.* (2016). An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *ArXiv e-prints*, article ID: 1601.00891.

Lee, I. *et al.* (2010). Rational association of genes with traits using a genome-scale gene network for arabidopsis thaliana. *Nat. Biotechnol.*, **28**, 149–156.

Re, M. and Valentini, G. (2013). Network-based Drug Ranking and Repositioning with respect to DrugBank Therapeutic Categories. *IEEE/ACM Trans Comput Biol Bioinform*, **10**(6), 1359–1371.

Re, M., Mesiti, M., and Valentini, G. (2012). A Fast Ranking Algorithm for Predicting Gene Functions in Biomolecular Networks. *IEEE/ACM Trans Comput Biol Bioinform*, **9**(6), 1812–1818.

Taylor, A. *et al.* (2015). GeneNet Toolbox for MATLAB: a flexible platform for the analysis of gene connectivity in biological networks. *Bioinformatics*, **31**(3), 442–444.

Valentini, G. *et al.* (2014). An extensive analysis of disease-gene associations using network integration and fast kernel-based gene prioritization methods. *Artificial Intelligence in Medicine*, **61**(2), 63–78.

Wang, X. *et al.* (2011). HTSanalyzeR: an R/Bioconductor package for integrated network analysis of high-throughput screens. *Bioinformatics*, **27**(6), 879–880.

Zheng, S. *et al.* (2012). GenRev: Exploring functional relevance of genes in molecular networks. *Bioinformatics*, **99**, 183–188.