Seidr: a gene meta-network calculation toolkit

Bastian Schiffthaler[1,*], Alonso Serrano[2], Nathaniel Street[1] and Nicolas Delhomme[2,*]

[1] Umeå Plant Science Centre, Department of Plant Physiology, Umeå University, Sweden, [2] Umeå Plant Science Centre, Department of Forest Genetics and Plant Physiology, Swedish University of Agricultural Sciences, Sweden

Abstract

Summary: Gene network analysis is a powerful tool to identify and prioritize candidate genes, especially from data sets where experimental design renders other approaches, such as differential expression analysis, limiting or infeasible. Numerous gene network inference algorithms have been published and are commonly individually applied in transcriptomics studies. It has, however, been shown that every algorithm is biased towards identifying specific types of gene interaction and that an ensemble of inference methods can reconstruct more accurate networks. This approach has been hindered by lack of an implementation to run and combine such combinations of inference algorithms. Here, we present Seidr: a toolkit to perform multiple gene network inferences and combine their results into a unified meta-network.

Availability and implementation: Seidr code is open-source, available from GitHub and also compiled in docker and singularity containers. It is implemented in C++ for fast computation and supports massive parallelisation through MPI. Documentation, tutorials and exemplary use are available from https://seidr.readthedocs.io.

Contact: bastian.schiffthaler@umu.se, nicolas.delhomme@slu.se

Introduction

Broader accessibility to RNA sequencing (RNA-Seq), (Mortazavi *et al.*, 2008) is resulting in a dramatic increase of transcriptomic studies and we observe a shift from binary, low-replication, differential expression (DE) analysis designs towards experiments more complex in nature (*e.g.* time-series studies). While DE analyses can be conducted for such complex study designs, alternative analysis methods, which for example account for gene-gene interactions, can provide more powerful biological insight into the complex network and dynamics of transcriptome modulation.

Gene network inference is one example method to investigate complex transcriptomics data. Briefly, based on "guilt-by-association" inferences computed from expression data, a network of gene interactions is built, then partitioned into clusters allowing for the identification and characterization of sub-networks relevant to the studies' hypothesis. Such inferences are typically calculated by relying on a single scoring statistic, such as correlation, mutual information or regression, and many methods have been implemented over the years. The DREAM5 challenge benchmarked a large number of available inference algorithms and revealed that each method was biased towards certain types of biological

interaction (Marbach *et al.*, 2012). Further, they reported that aggregating several networks into a meta-network significantly improved inference accuracy – the so-called 'wisdom of crowds'.

Despite these findings, there has been no comprehensive implementation of meta-network calculation to enable wider use of the approach. Inference algorithms have been implemented in various programming languages (C, C++, R, MATLAB), often requiring custom code compilation; furthermore, their input and output (I/O) formats are tool-specific and there has been no standardization effort. Finally, the computational expense required to calculate some of these networks is a major hindrance, often additionally impeded by inefficient design, poor implementation and programming language limitations.

In this article, we present Seidr, a comprehensive toolkit to infer multiple gene networks, aggregate them into a meta-network and facilitate common downstream analyses. To address the afore-mentioned limitations, Seidr has been written with a strong emphasis on speed and parallelism optimizations. Furthermore, Seidr uses a standardized I/O format, while also offering a consistent user-friendly command line interface; both of which are inspired by the SAM/BAM format (Li *et al.*, 2009) and the samtools CLI, which has become an established standard in bioinformatics. Seidr is available as open-source code from our GitHub repository and, as both a Docker and Singularity containers, ready to be deployed on a local or cloud-based computational resource.

Methods

Briefly, Seidr implements the meta-network calculation in three steps: network inference, ranking and aggregation (Figure 1) – which are all implemented as optimized, multi-threaded, C++ executables (Supplementary Table 1). The completion of these steps typically returns a fully dense, meta-network, for which Seidr includes functions to further assist in 1) identifying biologically relevant edges (referred to hereafter as thresholding), 2) calculating varied network metrics such as centrality, page rank, *etc.* and 3) exporting the network information for graph partitioning, clustering or gene set enrichment analyses (GSEA).

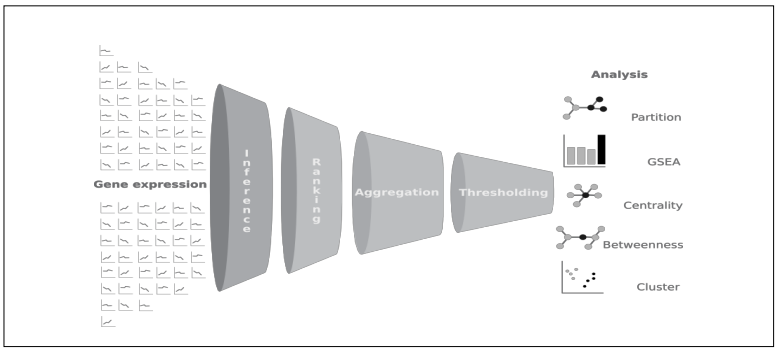

Fig. 1. Seidr workflow. A schematic representation of the Seidr inference, aggregation and thresholding steps and common downstream analyses.

- **Network inferences**

Seidr includes C++ (re-)implementations of 12 published network inference algorithms, leveraging fast open-source libraries (see Supplementary Table S1). Those that are particularly computationally demanding have been (re-)implemented to allow for parallel calculations using the Message Passing Interface (MPI) protocol. This makes Seidr well-suited for running on high performance computing environments, which is further facilitated by a Nextflow (Di Tommaso *et al.*, 2017) implementation of the Seidr pipeline.

- **Ranking**

Prior to their aggregation, the inference-method-specific output is standardized and converted into the Seidr unified 'sf' format. During this process, edge scores are transformed into value ranks, making them comparable across networks. Edges with tied weights are assigned the mid rank of the tied values. Finally, if the input is a non-symmetrical matrix, the best weight is retained, and directionality information is kept.

- **Aggregation**

Seidr implements different aggregation methods. Briefly, for each edge in the network a representative final edge is selected by one of the Borda (mean rank aggregation), Top 1 (highest rank among all available methods) (Hase *et al.*, 2013) or inverse rank product (Zhong *et al.*, 2014) methods. The aggregation output is a fully dense network; *i.e.* every node has a weighted edge to every other node.

- **Thresholding**

This step consists of selecting the most biologically relevant edges from the meta-network. Seidr offers two means of selecting the relevant edges; a backbone approach (Coscia and Neffke, 2017) or the lowest edge rank to retain in the final network is identified programmatically from two network topology statistics: scale free fit and transitivity.

- **Downstream analyses**

Seidr uses the parallel NetworKit library (Staudt *et al.*, 2016) to calculate centrality statistics for further interpretation of the network. Currently, Seidr reports the PageRank, Closeness, Betweenness, Strength, Eigenvector centrality and Katz centrality per node, and the Spanning Edge centrality and Edge Betweenness per edge. These can be used to identify clusters, albeit we also routinely use InfoMap (Rosvall and Bergstrom, 2008) to partition the network in clusters. Finally, Seidr can export the gene list of identified clusters for conducting *e.g.* Gene Ontology enrichment.

Discussion

We present Seidr, a toolkit for meta-network calculations, which aggregates the results of 12 inference approaches in a standardized, time-optimized process. This is, to our knowledge, the first user-oriented implementation of the DREAM5 Challenge postulate. Despite the optimization, some network inferences are still computationally intensive (Supplementary Figure 2), and for that reason, as well as to promote reproducible research, we provide Seidr as a docker container to use in massively parallel computing environments.

Acknowledgements

The authors would like to acknowledge UPPMAX and HPC2N HPC infrastructure support, Chanaka Mannapperuma for the design of Figure 1, Dr. Niklas Mähler for extensive testing and feedback, Dr. Martin Rosvall for discussion.

Funding

ND and BS are supported by the Wallenberg foundation. NRS and AS are supported by the Trees and Crops for the Future (TC4F) project. This project was supported by funds from Vinnova (the Swedish Governmental Agency for Innovation Systems) and KAW (The Knut and Alice Wallenberg Foundation).

Conflict of Interest: none declared.

References

Coscia,M. and Neffke,F.M.H. (2017) Network backboning with noisy data. In, *Proceedings - International Conference on Data Engineering*.

Hase,T. *et al.* (2013) Harnessing Diversity towards the Reconstructing of Large Scale Gene Regulatory Networks. *PLoS Comput. Biol.*, **9**, e1003361.

Li,H. *et al.* (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Marbach,D. *et al.* (2012) Wisdom of crowds for robust gene network inference. *Nat. Methods*, **9**, 796–804.

Mortazavi,A. *et al.* (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Methods*, **5**, 621–628.

Rosvall,M. and Bergstrom,C.T. (2008) Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci. U. S. A.*, **105**, 1118–23.

Staudt,C.L. *et al.* (2016) NetworKit: A tool suite for large-scale complex network analysis. *Netw. Sci.*, **4**, 508–530.

Di Tommaso,P. *et al.* (2017) Nextflow enables reproducible computational workflows. *Nat. Biotechnol.*, **35**, 316–319.

Zhong,R. *et al.* (2014) Ensemble-Based Network Aggregation Improves the Accuracy of Gene Network Reconstruction. *PLoS One*, **9**, e106319.