FastMEDUSA: A Parallelized Tool to Infer Gene Regulatory Networks

Serdar Bozdag  
*Marquette University, serdar.bozdag@marquette.edu*

Aiguo Li  
*National Institutes of Health*

Stefan Wuchty  
*National Institutes of Health*

Howard A. Fine  
*National Institutes of Health*

FastMEDUSA: a parallelized tool to infer gene regulatory networks

Serdar Bozdag1, Aiguo Li1, Stefan Wuchty1,2, and Howard A. Fine1,*

1Neuro-Oncology Branch, National Cancer Institute, National Institute of Neurological Diseases and Stroke
2National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD 20892, USA

ABSTRACT

Motivation: In order to construct gene regulatory networks of higher organisms from gene expression and promoter sequence data efficiently, we developed FastMEDUSA. In this parallelized version of the regulatory network-modeling tool MEDUSA, expression and sequence data are shared among a user-defined number of processors on a single multi-core machine or cluster. Our results show that FastMEDUSA allows a more efficient utilization of computational resources. While the determination of a regulatory network of brain tumor in Homo sapiens takes 12 days with MEDUSA, FastMEDUSA obtained the same results in 6 h by utilizing 100 processors.

Availability: Source code and documentation of FastMEDUSA are available at https://wiki.nci.nih.gov/display/NOBBioinfo/FastMEDUSA

Supplementary information: Supplementary data are available at Bioinformatics online.

Received on January 19, 2010; revised on April 21, 2010; accepted on May 21, 2010

1 INTRODUCTION

Among the numerous ways to determine gene regulatory networks of higher organism (Bussmaker et al., 2001; Elemento et al., 2007; Segal et al., 2003), MEDUSA is a well-known and powerful computational tool (Kundaje et al., 2008). Utilizing a boosting algorithm (Freund and Shapire, 1997), MEDUSA models promoter sequences and gene expression data from various experimental conditions. Providing a global predictive model of condition-specific expression states of target genes, MEDUSA has been successfully used to discover novel regulators in oxygen and heme regulatory expression states of target genes, MEDUSA has been successfully used to discover novel regulators in oxygen and heme regulatory networks in Saccharomyces cerevisiae (Kundaje et al., 2008).

Although MEDUSA is a powerful computational tool for inferring gene regulatory networks, it has certain limitations. MEDUSA requires the commercially available software package, MATLAB. In addition, the execution time of MEDUSA increases dramatically with larger sizes of input data sets. For example, MEDUSA needed >4 weeks for the modeling and analysis of about 7000 genes using 1000 iterations. Parallel computing has been used to solve the latter problem for several bioinformatics applications to date (Xiaohong et al., 2009). Here we introduce FastMEDUSA, a fast, parallelized, open source implementation of MEDUSA in C++ that uses freely available libraries. Our benchmark results showed that FastMEDUSA allows to model gene sets about 40 times faster than the original implementation of MEDUSA, utilizing 100 processors.

2 METHODS

2.1 Algorithm

In MEDUSA, a training set consists of a matrix of discretized expression pairs (g, r) of a gene g in an experiment r. MEDUSA applies boosting on the training set to iteratively build an alternating decision tree (ADT), which consists of weighted weak classifiers. A weak classifier, which consists of motifs that classify a set of examples (g, r), is a classifier score where motif m, regulator r and expression state s of r, classifies a set of examples (g, r) where motif m is present in the promoter of gene g and the expression state of regulator r is s in experiment r. A weak classifier with maximum classifier score is added to the ADT iteratively. Each element in the training set is reweighted for the next iteration where misclassified elements get a higher weight than accurately classified elements. In a final ADT, the total weight (i.e. prediction score) of the weak classifiers that satisfy an element in the test set represents the overall prediction for this element.

Initially, MEDUSA computes all motifs (i.e. k-mers and optionally dimers) from promoter sequences of genes (Fig. 1a). At each iteration, first, optimum weak classifier and its optimum position in the ADT are computed based on a classifier score. Subsequently, the algorithm generates probabilistic motifs by clustering the top n motifs that optimize classifier score, where n is a user-defined parameter. After computing pairwise distances, motifs are clustered iteratively. In each clustering step, classifier score of the clustered motif is computed by scanning the motif against the promoter sequences. If a clustered motif has a higher score than the motif in the weak classifier has, the motif is replaced by this clustered motif. The final weak classifier is added to the ADT, and each element in the training set is reweighted for the next iteration. For algorithmic details of MEDUSA, see Kundaje et al. (2007) and Middendorf et al. (2005).

FastMEDUSA, a parallelized implementation of the MEDUSA algorithm, designates one of the processors as the root processor to manage inter-processor communication and compute global results. The root processor assigns a unique subtree of the ADT to each remaining processor, which computes optimum weak classifier and its position for the assigned subtree and sends results back to the root processor. In addition, the root processor partitions genes among remaining processors equally to parallelize gene-wise operations. For instance, each processor scans the clustered motif against promoter sequences of the assigned genes in each clustering step in parallel. Parallelizing reweighting elements, each processor reweights only the assigned genes. Summarizing all parallelized steps, we present a flowchart of FastMEDUSA in Figure 1a.

2.2 Implementation

FastMEDUSA was implemented in C++, utilizing the message passing interface (MPI) implementation MPICH2 for inter-process communication (http://www.mcs.anl.gov/research/projects/mpich2/) and GotoBLAS2
We tested the performance of FastMEDUSA and MEDUSA utilizing yeast and human data sets for 200 and 400 iterations. We observed that FastMEDUSA consistently outperforms MEDUSA. The graphs and raw data of output ADTs are reported in Supplementary Figures S1–S8 and Supplementary Tables S1–S8.

We would like to thank Dr Christina Leslie and Steve Lianoglou for their useful discussion about MEDUSA.

ACKNOWLEDGEMENTS

We would like to thank Dr Christina Leslie and Steve Lianoglou for their useful discussion about MEDUSA.

**Funding:** Intramural Research Program of the National Institutes of Health; National Cancer Institute.

**Conflict of Interest:** none declared.

REFERENCES


