MR-microT: A MapReduce-based MicroRNA Target Prediction Method

Ilias Kanellos
NTU Athens, Greece
kanellos@dblab.ece.ntua.gr

Thanasis Vergoulis
IMIS, R.C. Athena, Greece
vergoulis@imis.athena-innovation.gr

Dimitris Sacharidis
IMIS, R.C. Athena, Greece
dsachar@imis.athena-innovation.gr

Theodore Dalamagas
IMIS, R.C. Athena, Greece
dalamag@imis.athena-innovation.gr

Artemis Hatzigeorgiou
BSRC Al. Fleming, Greece
hatzigeorgiou@fleming.gr

Stelios Sartzetakis
IMIS, R.C. Athena, Greece
stelios@imis.athena-innovation.gr

Timos Sellis
RMIT University, Australia
timos.sellis@rmit.edu.au

ABSTRACT

MicroRNAs (miRNAs) are small RNA molecules that inhibit the expression of particular genes, a function that makes them useful towards the treatment of many diseases. Computational methods that predict which genes are targeted by particular miRNA molecules are known as target prediction methods. In this paper, we present a MapReduce-based system, termed MR-microT, for one of the most popular and accurate, but computational intensive, prediction methods. MR-microT offers the highly requested by life scientists feature of predicting the targets of ad-hoc miRNA molecules in near-real time through an intuitive Web interface.

1. INTRODUCTION

MicroRNAs (*miRNAs*) are small non-protein coding RNA molecules that bind on the transcripts of particular genes, called *targets*, and, then, inhibit their expression. Since this function associates them to the causes and the treatment of many diseases (e.g., various types of cancer [2, 9]), many efforts have been made to discover the targets of each miRNA molecule.

A variety of experimental techniques for the identification of miRNA targets exist (e.g., Northern Blot, PCR). However, due to (a) the large number of miRNAs that have been discovered and (b) the large cost of these experimental techniques, many computational methods for rapid prediction of miRNA targets have been developed. A good survey presenting the majority of these methods can be found in [1].

MicroT [7] is one of the most popular and accurate miRNA target prediction methods. Finding the targets of a particular miRNA molecule that reside in a particular genome, consists of three steps in microT. In the first step, a sequence alignment algorithm is used to provide a set of candidate targets based on the first 9 base pairs of the miRNA sequence. Each of the candidate targets contains at least one location where these base pairs can be aligned. These locations are candidate binding sites for the miRNA molecule. In the second step, a conservation score is computed for each candidate binding site by considering the number of species in which the binding site is preserved. Finally, in the third step, the scores calculated for the binding sites of each candidate target are aggregated, considering also some external parameters (e.g., the folding of the involved molecules), to produce a final score for each target that reflects the probability that the candidate target is real.

Although microT, as well as other target prediction methods, provide Web interfaces to return to users precomputed predicted targets of all known miRNA molecules [6], little effort is spent on providing near-real time execution of target prediction methods for novel miRNA molecules through a Web interface. This is a highly desired, yet difficult to provide, feature. The difficulty lies on the fact that target prediction methods involve heavy computations. For reference, microT requires dozens of minutes to produce the targets of a particular miRNA molecule, when executed on a single server.

Cloud computing platforms have been used in the past to expedite miRNA target prediction methods. An existing system, termed TarCloud [8], utilizes Microsoft's Azure Cloud platform to also implement the microT method. TarCloud, however, does not parallelize the prediction of the targers for a given miRNA. As a result, the execution time for a *single* miRNA is not accelerated and thus cannot be scaled up by assigning more processing nodes. While, TarCloud can find the targets for *multiple* miRNAs at the same time by distributing miRNA tasks, the response time is bounded by the execution time for a single miRNA. Moreover, the TarCloud implementation is dependent on the Azure platform making it impossible to port the system to another

SSDBM '14, June 30 – July 2, 2014, Aalborg, Denmark.
ACM 978-1-4503-2722-0/14/06.
http://dx.doi.org/10.1145/2618243.2618289

Cloud provider or to a private cluster if needed.

This paper introduces the *MR-microT* system, which is a MapReduce-based adaptation of the microT method. The key feature of MR-microT is the parallelism of the prediction process, so that the execution time for a *single* miRNA can be accelerated as desired by allocating more resources. This is achieved by carefully dividing the input data (genome and cross-species conservation information) among different processing nodes. Moreover, similar to TarCloud, MR-microT can accelerate the execution for *multiple* miRNAs. Finally, MR-microT comes with an intuitive and powerful Web interface[1] which can be used by researchers to produce target predictions for arbitrary miRNA sequences. As a result, MR-microT is the first microT implementation capable of near-real time prediction of ad hoc miRNA targets. Since, several design choices in MR-microT could apply to other similar computationally intensive bioinformatics method, our ambition for MR-microT is to serve as an exemplary system in the field.

2. THE MR-MICROT SYSTEM

In Section 2.1, we present the technologies used by MR-microT system. Next, in Section 2.2, the architecture of the system is described. Then, in Section 2.3, we discuss the specifics of the MapReduce implementation. Finally, in Section 2.4, we overview the user interface.

2.1 Technologies

MR-microT is a distributed implementation of the microT target prediction method designed to run on computational clusters of arbitrary sizes. It utilizes the Hadoop[2] framework for the distribution of microT computations in the nodes of the cluster. Hadoop is an open source framework that implements the *MapReduce* distributed programming paradigm [4] and can be easily installed in almost any contemporary computational cluster. The execution of a MapReduce program consists of the *Map* and the *Reduce* phase. During the Map phase, the input is split into disjoint pieces and each piece is independently processed by a different node of the cluster to produce a set of key-value pairs. The Reduce phase groups these key-value pairs by key and processes their values together to produce the output of the program. We use the Hadoop streaming utility[3] to exploit large amounts of existing Perl scripts written by bioinformaticians.

Most of the data required by the microT execution are stored in the Hadoop Distributed File System (HDFS). This distributed file system lies on the hard disks of the nodes of the cluster and it is used for the storage of the input and the output of any Hadoop code. We use HBase[4], a non-relational distributed database inspired by Google's BigTable [3], to store data that must be retrieved ad hoc during run-time.

The cluster which hosts MR-microT is a Virtual Network consisting of 17 Ubuntu Virtual Machines, provided by the Cloud service of ∼okeanos[5] [5]. ∼okeanos is an IaaS platform providing resources to the Greek research and academic community.

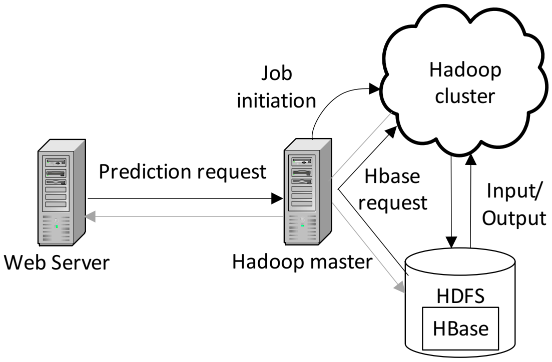


Figure 1: MR-microT system architecture

2.2 System architecture

The architecture of MR-microT is depicted on Figure 1. In particular, MR-microT consists of (a) a Web server, that collects user requests for target predictions of miRNA sequences, (b) a Hadoop cluster, on the nodes of which our code is executed, and (c) a Hadoop master, which is a Virtual Machine responsible for managing the Hadoop cluster and the HDFS and HBase storage resources.

The Web server is an Ubuntu Virtual Machine carrying an instance of the Apache server and holding MR-microT's user interface, described in Section 2.4. This front-end is written in PHP and collects user's HTTP requests to the MR-microT system for given miRNA sequences. The Web server sends these requests to the Hadoop master, which distributes the computations required by the requests to nodes of the Hadoop cluster. Each node reads input files from the HDFS during the initialization phase and requests HBase records when needed. Note that the master node is also the proxy of the HDFS and the server of the HBase requests.

As the cluster nodes execute their tasks, the master monitors their progress. The Web server polls the master for progress reports and presents them to the user in a human-friendly form. The last progress report of the master informs the Web server that the process is completed, and, then, the output data are transfered from the HDFS to the Web server in order to be presented to the user.

2.3 MapReduce microT

2.3.1 Overview

There are three basic execution steps in the microT method for finding the targets of a miRNA molecule for a particular species. First, a set of candidate targets is produced by aligning the first 9 base pairs of the miRNA sequence inside the protein coding and the 3-UTR region of each gene sequence of this species. A gene is called a candidate target if it contains at least one location in its protein coding or 3-UTR region where these 9 base pairs can be aligned. These locations are called *miRNA recognition elements (MREs)* and they are candidate binding sites for the miRNA molecule inside the target. For each MRE, the quality of the alignment is recorded as its *alignment score*. In the second step, a *conservation score* is calculated for each MRE based on the number of species in which the MRE is preserved. Finally,

[1] http://diana.imis.athena-innovation.gr/DianaTools/index.php?r=mrmicrot

[2] http://hadoop.apache.org/

[3] http://hadoop.apache.org/docs/stable1/streaming.html

[4] https://hbase.apache.org/

[5] https://okeanos.grnet.gr/

in the third step, an aggregated score is calculated for each gene of the species, taking into consideration the computed scores for all its MREs, as well as some other factors, such as the folding of the involved molecules.

Our MapReduce design is based on the following observation. It is possible to perform the computations involved in the first two steps of the microT method independently for each gene. Therefore we choose to *map* these independent tasks to different nodes of the cluster (of course, if the total number of tasks is greater than the number of available nodes, then some nodes may need to execute consecutively multiple tasks). Such a task distribution can be realized by implementing the first two steps of the microT method as the Map phase of a Hadoop code. The input required in this Map phase is stored in the HDFS as a set of records, where each one contains data related to a region (protein coding or 3-UTR) of a particular gene. The output of the Map phase would be a set of records as well, where each record encodes the location of an MRE, the scores calculated for it, and the identifier of the target, i.e., the gene, into which it resides.

The third step of microT consists of calculating a final score for each target by combining the alignment and conservation scores of all the MREs residing in it. Therefore, in MR-microT, this step is implemented as the Reduce phase, consuming the output of the previously described Map phase. For all records having the same target identifier, the Reduce phase calculates the final score considering also the molecular folding.

In the next sections, we describe in more detail the Map and Reduce phases of the MR-microT system.

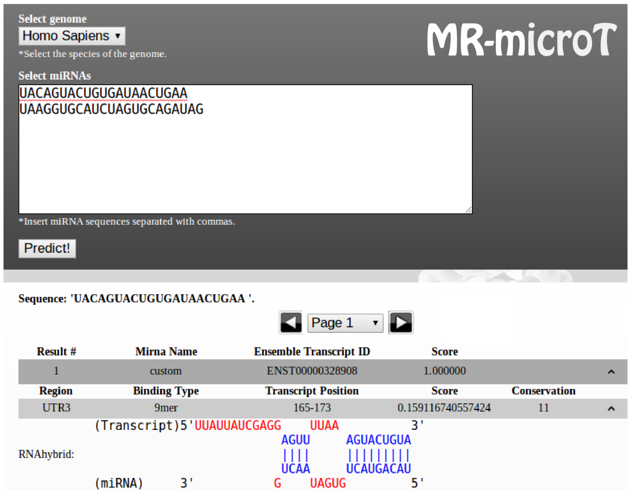
2.3.2 *Map phase*

The Map phase gets the protein coding or the 3-UTR region of a gene sequence along with some information related to its preservation in a predefined set of species, and produces the MREs of that sequence along with their alignment and conservation scores. The input is organized in files stored in HDFS. Each line of these files corresponds to one (protein coding or 3-UTR) region of a gene and it is structured as a key-value pair. The key of the pair is the concatenation of gene's Ensembl identifier with a string that denotes the type of the region (protein coding or 3-UTR). The value of the pair is a struct containing the sequence of the gene region along with some basic information about the gene and its conservation in a number of species.

For each key-value pair, the Map code first executes a sequence alignment algorithm to find all the alignments of the first 9 base pairs of the miRNA sequence in the gene region sequence contained in the value of the pair. Each found alignment of these base pairs is an MRE and an alignment score is calculated for it based on the quality of the alignment and the strength of the bonds which are going to be created in case of a molecular binding.

Then, the conservation information of the gene sequence is considered and a conservation score is calculated. The more species that preserve the gene sequence unaltered exist, the larger the resulting conservation score is. Note that, the calculation of this score depends on precomputed weights for each possible pair of 3-grams (a pair consists of a gram from the reference species and a gram from another species). These weights are required ad hoc during the Map phase execution and are thus stored in HBase.

Finally, a set of key-value pairs, one for each found MRE, is produced as an output of the Map phase. The key of each pair is the the Ensembl identifier of the gene in which the MRE resides and the value encodes the location of an MRE along with its calculated alignment and conservation scores.



Select genome

Homo Sapiens

*Select the species of the genome.

Select miRNAs

UACAGUACUGUGAUAACUGAA
UAAGGUGCAUCUAGUGCAGAUAG

*Insert miRNA sequences separated with commas.

Sequence: 'UACAGUACUGUGAUAACUGAA '.

Result #	Mirna Name	Ensemble Transcript ID	Score
1	custom	ENST00000328908	1.000000

Region	Binding Type	Transcript Position	Score	Conservation
UTR3	9mer	165-173	0.159116740557424	11

```
           (Transcript)5'UUAUUAUCGAGG    UUAA        3'
                                     AGUU    AGUACUGUA
RNAhybrid:                           ||||    |||||||||
                                     UCAA    UCAUGACAU
           (miRNA)     3'           G    UAGUG       5'
```

Figure 2: Screenshot of MR-microT's user interface

2.3.3 *Reduce phase*

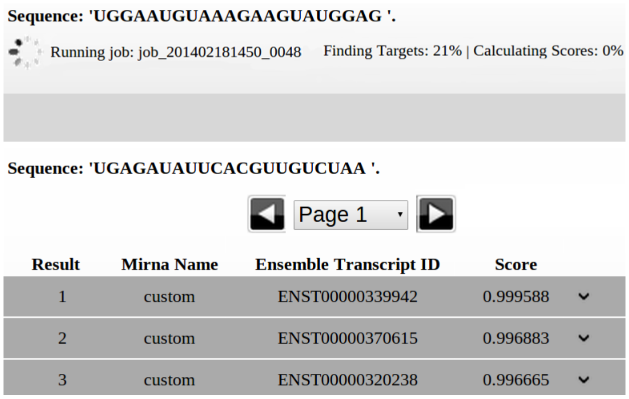
The Reduce phase consumes the output of the Map phase. The Hadoop framework ensures that all the key-value pairs having the same key (i.e., the same gene identifier) are going to be processed by the same cluster node. This node aggregates the alignment and the conservation scores of all MREs found for this gene, and derives a prediction score for the gene itself. During this calculation, the Reduce phase also considers the folding of the involved molecules (this is important as the folding could actually destroy a possible binding). Note that the folding information is stored in HBase and is requested ad hoc during the execution. The output of the Reduce phase is recorded in HDFS and the Hadoop master is informed about its storage location.

2.4 User interface

The services of MR-microT are accessible through a powerful yet intuitive Web interface. A screenshot of the interface is presented in Figure 2. First, the user selects the species in which she wants to find targets from a drop down list. Currently, there are two available choices: Homo Sapiens (i.e., human) and Mus Musculis (i.e., mouse). Then, she inserts one or more miRNA sequences in a text box.

When the species and the miRNA sequences are specified, the user can send his prediction request to the system. This is done by clicking the button labeled "Predict!". After that, the user has the opportunity to monitor the target prediction progress for *each* of the sequences she had specified. The system displays a separate *progress view* for each miRNA sequence into which the progress of target prediction is rendered. An example progress view is shown in the upper part of Figure 3.

When the target prediction method for a miRNA sequence is completed, the progress view is augmented with the result view, shown in the lower part of Figure 3, which contains information about the computed targets. Since the number of targets is usually large, they are organized into pages.



Sequence: 'UGGAAUGUAAAGAAGUAUGGAG '.

Running job: job_201402181450_0048 Finding Targets: 21% | Calculating Scores: 0%

Sequence: 'UGAGAUAUUCACGUUGUCUAA '.

Result	Mirna Name	Ensemble Transcript ID	Score
1	custom	ENST00000339942	0.999588
2	custom	ENST00000370615	0.996883
3	custom	ENST00000320238	0.996665

Figure 3: Progress view of MR-microT, displaying the current status of the target prediction of a single miRNA; and Result view presenting the list of predicted targets

Each page displays information related to a number of targets. The information for target is rendered within the grey boxes shown in Figure 3. This information box contains the gene identifier and the final target prediction score calculated for this gene. Details regarding the list of the predicted binding sites of the target can be found by clicking on the arrow button located at the right of the grey box.

3. DEMONSTRATION SCENARIOS

We demonstrate the functionality and the scalability of MR-microT by executing queries provided by us and by members of the audience. Below we discuss three concrete demonstration scenarios.

Scenario 1. Through a Web interface developed for the needs of the demo, the member of the audience requests the predictions for a miRNA sequence to be produced by a microT instance running on a single server. Concurrently, she sends the same request to the MR-microT system through its Web interface. This scenario will demonstrate the near-real time efficiency of our system, especially with respect to the single server microT algorithm.

Scenario 2. The member of the audience requests the predictions for a small number of miRNA sequences. After a few moments, during which she monitors the rate of progress for each of the given sequences, she sends more prediction requests for some additional sequences. This scenario will demonstrate the large capacity of MR-microT for handling multiple requests with no observable reduction in its efficiency.

Scenario 3. The member of the audience sends the prediction requests for the same miRNA sequences to MR-microT clusters of different sizes. Then, she monitors the progress of microT execution in both clusters (through a Web interface developed for the needs of the demo). The objective of this scenario is to show the scalability of MR-microT, i.e., the fact that as more resources are allocated, the efficiency of our system improves.

4. ACKNOWLEDGMENTS

This work was supported by the project “MEDA” by the Greek General Secretariat for Research and Technology.

5. REFERENCES

[1] Panagiotis Alexiou, Manolis Maragkakis, Giorgos L. Papadopoulos, Martin Reczko, and Artemis G. Hatzigeorgiou. Lost in translation: an assessment and perspective for computational microrna target identification. *Bioinformatics*, 25:3049–3055, 2009.

[2] Richard W Carthew. Gene regulation by micrornas. *Current Opinion in Genetics & Development*, 16:203–208, 2006.

[3] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Tushar Chandra Mike Burrows, Andrew Fikes, and Robert E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems*, 26, 2008.

[4] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51:107–113, 2008.

[5] Evangelos Koukis and Panos Louridas. ∼okeanos iaas. *Proceedings of Science*, 2012.

[6] Maria D. Paraskevopoulou, Georgios Georgakilas, Ioannis S. Vlachos Nikos Kostoulas, Thanasis Vergoulis, Martin Reczko, Theodore Dalamagas Christos Filippidis, and A.G. Hatzigeorgiou. Diana-microt web server v5.0: service integration into mirna functional analysis workflows. *Nucleic Acids Research*, 41:W169–W173, 2013.

[7] Martin Reczko, Manolis Maragkakis, Panagiotis Alexiou, Ivo Grosse, and Artemis G. Hatzigeorgiou. Functional microrna targets in protein coding sequences. *Bioinformatics*, 28:771–776, 2012.

[8] Thanasis Vergoulis, Michail Alexakis, Manolis Maragkakis Theodore Dalamagas, Artemis G. Hatzigeorgiou, and Timos Sellis. Tarcloud: A cloud-based platform to support mirna target prediction. *Lecture Notes in Computer Science*, 7338:628–633, 2012.

[9] Lin Zhang, Stefano Volinia, Tomas Bonome, George Adrian Calin, Joel Greshock, Nuo Yang, Chang-Gong Liu, Antonis Giannakakis, Pangiotis Alexiou, Kosei Hasegawa, Cameron N. Johnstone, Molly S. Megraw, Sarah Adams, Heini Lassus, Jia Huang, Sippy Kaur, Shun Liang, Praveen Sethupathy, Arto Leminen, Victor A. Simossis, Raphael Sandaltzopoulos, Yoshio Naomoto, Dionyssios Katsaros, Phyllis A. Gimotty, Angela DeMichele, Qihong Huang, Ralf BÃijtzow, Anil K. Rustgi, Barbara L. Weber, Michael J. Birrer, Artemis G. Hatzigeorgiou, Carlo M. Croce, and George Coukos. Genomic and epigenetic alterations deregulate microrna expression in human epithelial ovarian cancer. *Proceedings of the National Academy of Sciences*, 105(19):7004–7009, 2008.