Advanced eQTL analysis for identification of regulatory factors and their target genes

Wenjie Qi
Biomedical Engineering
Michigan State University

David Filipovic
Biomedical Engineering
Michigan State University

***Abstract*— Different individuals within a species possess many nucleotides within their genomic sequence that differ from other individuals. These single nucleotide polymorphisms (SNPs) can occur within genes coding for transcription factors (proteins with regulatory function that influences epxression levels of other proteins). These SNPs hold the potential to result in a residue change within the functional parts of the transcription factor protein (missense or nonsense mutations), such as those changes occurring within their DNA binding or cofactor interaction domains. These residue changes, in turn can result in a change in the level of activity of the transcription factor which would be reflected in the expression levels of some or all of its direct targets. The present study investigated whether information about all the SNPs and how it relates with the expression of target genes, could be used for *de novo* predicton of transcription factors, their targets, or important SNPs (last case - when transcription factors and their targets are known). The three mehtods investigates – marker regression, allele transition bias and random forest regression did not produce any significant results.

I. Introduction

DNA is made of a sequence of nucleotides containing one of the four possible nitrogenous bases (A, C, T and G). All genetic information is precisely encoded in this sequence. Gene, a basic unit of heredity, is the portion of the DNA that encodes for a single protein and is used as an instruction manual in protein synthesis. The central dogma of molecular biology states that protein synthesis occurs in two steps: transcription and translation. In transcription DNA is read by the RNA polymerase and a complementary mRNA (messenger RNA) is constructed and processed. In translation mRNA is read by the ribosome and the final protein product is constructed. A specific mRNA transcript or protein product almost exclusively results from a single gene and the level of that product (either mRNA or protein) can be accurately quantified using biochemical methods. These levels are referred to as gene expression and could be used as targets of machine learning or other statistical algorithms. Gene expression, the process of using gene information for protein synthesis, is also used to refer to the amounts of functional gene product (usually mRNA or protein) resulting from this process. One way in which the gene expression process is regulated is through DNA-protein interactions (which can either stimulate or inhibit transcription). These interactions depend on the structure of the interacting protein, which in turn depends on the gene sequence coding for that protein. The most common way gene sequences exhibit variations in nature is the single nucleotide polymorphism (SNP) where a single nucleotide at a specific position has been replaced with another type of nucleotide. These individual genetic variations have been shown to affect phenotypic traits, most notably gene expression, e.g. in the work of Thomas et al, 2012. Therefore, it should be possible, using SNPs as input features, to construct a prediction framework of gene expression levels (or changes in the levels of their expression), i.e. it should be possible to associate gene expression with genetic variations.

II. Current approaches

Currently, the most commonly used method of associating gene expression with genetic variation is expression quantiative trait loci (eQTL) analysis. This analysis is based on a set of statistical methods that are used to ascertain how much of the genetic variation in gene expression can be attributed (or explained) by variation at a specific locus (a place of SNP occurrence). Xie et al., 2016, have developed a deep learning framework for prediction of gene expression levels, based on genetic variation data (specifically SNP features). They used a well-known microarray yeast dataset, collected by Brem and Krugklyak, 2005, and a deep neural network composed of stacked denoising autoencoders (consisting of 2 denoising autoencoders) with dropout and compared it to the same method without dropout, random forest and Lasso. The problem posed here was interpreted as a multiple regression problem involving supervised learning. However, the dimensionality of the problem is very high. The output contained levels of gene expression, which were quantified for 6111 genes, whereas the input contained genetic variation information for 2956 SNPs. Xie et al., 2016, have noted that this kind of problem was previously approached with simple regression methods, however these methods proved too complicated to be of any use for prediciton Therefore, Xie et al., 2016, have focused only on the SNPs. Additionally, methods like ChIP-seq can reveal DNA binding sites of various proteins, thus enabling us to label some of them as potential transcription factors, however, this kind of data does not tell us what gene targets are affected by this transcription factor-DNA interaction.

III. Goals

Our study aims to achieve three distinct goals -

1. De novo identification of transcription factors, when only the genotype and expression data is available.

2. De novo identification of transcription factor targets, when transcription factors are known, in addition to the genotype and expression data.

3. De novo identification of SNPs within transcription factors that are important for their function, when both transcription factors and their targets are known, in addition to genotype and expression data.

IV. Data and Methods

Original experiment

Two strains of brewer's yeast (*S. Cerevisiae*) were crossed and 112 unique F1 progeny strains were created. The parent strains used were - BY4716 also known as S288c (a highly inbred laboratory strain) and RM11-1a (a wild isolate strain from a California vineyard). These two strains are known to exhibit many SNPs within their ORFs. An example of one such SNP is shown in Figure 1.

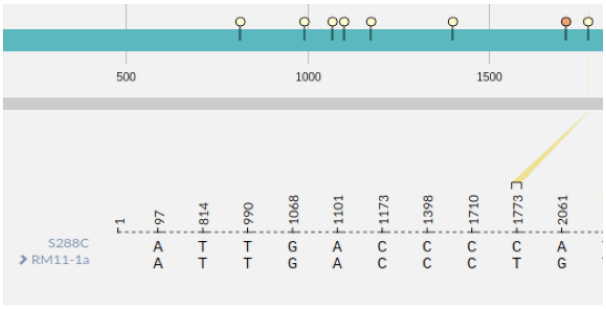

Figure 1 - Example of a SNP between two parent strains (courtesy of yeast genome variant viewer)

Data description and download

The data used in this analysis was originally generated by Brem and Kruglyak. It contains genotype (SNP positions and values) and mRNA expression data (log2 ratio of probe light intensities) for 112 cross-strains of brewer's yeast. We download and use three distinct datasets related to this experiment. All the data used in the analysis is included in the repository, however, a data download script is also included for convenience and future updates.

Data types

Genotype data contains information about the precise genotype of the progeny strains - SNP location within the genome, and its values. Genotype data includes 2957 SNPs and their values across conditions, however, only 2322 of those belong to protein coding ORFs and those are the SNPs considered in the subsequent analysis. SNP conditions are coded with the following values - 0, 1, and 2 (minor homozygot, heterozygot, major homozygot, resprectively). A general overview of the genotype data is shown in Figure 2.

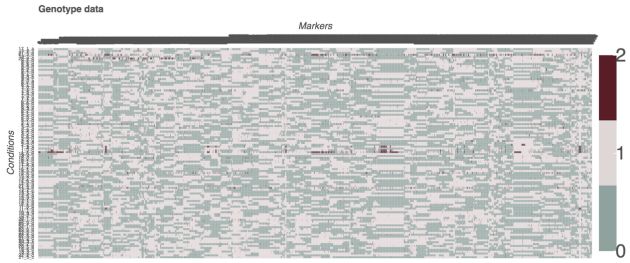

Figure 2 - A general view of the genotype data

Expression data contains expression values for 7084 microarray probes across the 112 cross-strains. However, only 6150 of those are probes for protein coding ORFs and those were the only ones considered.

Datasets

Genotype and expression dataset. Initial preprocessing of the data was performed by Xie et al. for their project of comparing machine learning methods in predicting gene expression based on genotype data. This, initially preprocessed data, is available at: https://github.com/shilab/MLP-SAE. Xie et al. have aggregated the original SOFT formatted expression data into a single tab-delimited matrix file (*expression_matrix.txt*). They have also compiled a tab-delimited matrix of genotypes across conditions (cross-species) and created supplementary files detailing the positions of all the SNPs and locations of all tested ORFs across the genome. This dataset contains a total of four files. The files, their descriptions and the descriptions of their columns are given below.

expression_location - contains locations of all ORFs coding for proteins - dubious genes were left outs, henceforth only the protein coding genes were used in our analysis.

ID - ORF id as it is used in the microarray

Chr - chromosome number

Start - First nucleotide of the ORF

Stop - Last nucleotide of the ORF

snp_position.txt - contains locations of all the SNPs

_probePairKey - id of the SNP probe (not unique across SNPs)

Chromosome - chromosome number

Position - nucleotide position of the SNP (within respective chromosome)

matrix_genotypes.txt

_probePairKey - id of the SNP probe (not unique across SNPs)

1_1_d… 26_2_d - conditions (each of the cross-strains)

expression_matrix.txt

ID - ORF id as it is used in the microarray

1_1_d… 26_2_d - conditions (each of the cross-strains)

Gene annotation dataset. The genotype and expression data did not contain a mapping between microarray ORF probe ids and actual ORF names or the names of their corresponding genes. This mapping is required to link the ORFs with the gene titles and symbols used as row and column headers in the validation dataset. The mapping can be found in the original dataset within the GPL118 microarray platform annotation file. This file is only available in the SOFT format and is downloaded and transformed into csv format by our data downloading script. This dataset contains only one file. The description of this file and the descriptions of its columns are given below. Certain spots on the chip are empty or contain a control probe, so they do not contain a mapping to a gene name, gene symbol or an ORF name. Other spots contain probes for ORFs that are considered dubious and most likely do not code for functional proteins. These spots will have a mapping to an ORF name, but will not contain mappings to a gene title or a gene name. A total of 6150 ORFs were available in the microarray. Out of those only 5670 ORFs were known to code functional proteins.

Id_to_gene_name.csv - contains a mapping of microarray ORF ids (ids of the spots on the chip) to gene titles, gene names and ORF names

ID - ORF id as it is used in the microarray

Gene title - the official title of the corresponding gene

Gene symbol - the official title of the corresponding gene

Platform_ORF - the official name of the ORF corresponding to the probe

Validation dataset download. For validation purposes, an interaction matrix of known transcription factors and their targets is downloaded by our data downloading script from the YEASTRACT repository of regulatory interactions in *Saccharomyces cerevisiae*. This dataset contains only one file. The description of this file is given below.

RegulationMatrix.csv - semicolon separated file, contains an interaction matrix where the rows represent all known transcription factors (referred to by their gene title) and the columns represent all known targets (referred to by their gene symbol). If there is a known interaction between an transcription factor and a protein target the corresponding cell will contain a 1, otherwise it will contain a 0.

All three datasets are downloaded and transformed (if necessary) by running the provided ***download_data.sh*** shell script.

RATIONALE

SNPs occurring within transcription factors hold the potential to result in a residue change within the functional parts of the transcription factor protein (missense mutations), such as those changes occurring within their DNA binding or cofactor interaction domain. These residue changes, in turn can result in a change in the level of activity of the transcription factor which would be reflected in the expression levels of some or all of its direct targets. Additionally, SNPs producing transcription termination codons (nonsense mutation) instead of amino acid codons, could result in a truncated protein which could lose parts or all of its functionality and have an even greater effect on the expression levels of all of its direct targets.

ANALYSIS METHODS

To achieve the three goals of this study we employ three analysis methods that were adapted from existing methods or developed for the purposes of this study. These methods include -

1. **Marker regression** - a simple binary classification method commonly used in eQTL analysis. This method involves initially considering all SNP containing genes as putative transcription factors. SNP containing genes are identified by ensuring that the SNP position is matched within the ORF region of the gene, which is determined by the ORF start and end location and the chromosome number. For every SNP containing gene, we checked if it is expressed differentially under different conditions of a given SNP (SNP values of 0, 1, or 2) by a series of t-tests. Genes that are expressed differentially ($p<0.01$) are removed from the list of candidate transcription factors, due to possible confounding between the effect of the transcription factor expression and their functionality altering SNPs. For every

SNP in all the remaining candidate genes, the expression value of all target genes under different SNP conditions is analyzed, and only differentially expressed genes between different SNP conditions ($p<0.01$) are considered as potential regulatory targets. The code for this method is contained within the R script ***MarkerRegression.rmd***. The result of this analysis is the ***target_gene_with_all_p_values.txt*** file.

2. **Allele transition bias** - simple binary classification method developed specifically for this study. This method involves finding all genes that are upregulated across a pair of conditions (a minimum of 2-fold increase in expression was used as criterion for upregulation) and matching them with SNP condition transitions. Downregulation is not considered separately, as it is the same as upregulation when the transition is reversed. The allele transition bias procedure is composed of two steps. In the first step, we find all the condition (specific cross-strain) transitions that result in upregulation, for every gene. In the second step, we match the SNP condition (0, 1 or 2) for all the condition transitions found in the previous step, for each SNP. For instance, if gene A was upregulated when going from condition 1 to condition 2, then we look at what happened with each SNP between those two conditions.
   If the SNP changed from 0 to 1, 0 to 2, or 1 to 2, we consider that a major bias transition; if the SNP changed from 2 to 1, 2 to 0 or 1 to 0, we consider that a minor bias transition; if the SNP did not change at all we consider that a no change transition. For each SNP and each gene, we find the number of major bias, minor bias and no change transitions. We use the percentage of major bias (or minor bias) as the cutoff for out binary classification. The code for this method is contained within the python jupyter notebook ***AlleleTransitionBias.ipynb***

3. **Machine learning (Random forest regression)** - similarly to Xie et al. we developed and tested a machine learning method for prediction of gene expression using a random forest regressor. Unlike Xie et al., we use the regressor to explore whether the target gene expression prediction could be improved by incorporating information about SNPs within transcription factors. This procedure involves applying the random forest regressor three times (on three slightly different input datasets). The input data in all three cases involves only 49 transcription factors that have been identified as having SNPs within their ORFs. The output (ground truth) data in all three cases is the gene expression data of all known regulatory targets of the 49 transcription factors being tested.

   The first case involves only the expression of the 49 transcription factors as the input data - **only expression case**. The second case adds information about 30 SNPs within these 49 transcription factors to the input data (Note: there was a total of 111 SNPs but due to linkage disequilibrium only 30 of them were found not to be exact copies of any other SNP) - **expression and all unique SNPs case**. The third case uses information about only five SNPs found to be important for regression in the previous step (by using the important features of the regressor), in addition to the transcription factor expression data - **expression and important SNPs case**. In all three cases the Random Forest had 2000 trees with the maximum depth set to 5, and K-fold (k=10) cross validation was applied to validate the regression results. The important features used by the tree was listed to identify the important SNPs in the "expression and important SNPs case".

The marker regression method is used for *de novo* identification of transcription factors and their targets; the allele transition bias methods is used only for *de novo* identification of targets, when transcription factors are known; and the machine learning method was applied to *de novo* identification of important SNPs within transcription factors.

ANALYSIS WORKFLOW

The analysis performed in this study can be grouped into four distinct steps: data downloading, data preprocessing, marker regression analysis, allele transition bias analysis, machine learning analysis. The analysis requires R, and python3. The following python3 packages should be installed: numpy, pandas, bokeh, matplotlib, scipy, sklearn.

1. To download data ***./download_data.sh*** shell script should be done. All the required data will be downloaded into the **./data** directory.
2. Before any analysis can be done the jupyter notebook ***./analysis/src/result_generation/do_preprocessing.ipynb*** should be executed to generate any auxiliary tables/files needed for the analysis. These files will be saved in the **./data/product** directory. The subsequent analysis steps can be done in any order.
3. To perform the marker regression analysis the jupyter notebook ***MarkerRegressionResultGeneration.ipynb*** should be executed. The results are ROC curve graphs, and these shall be saved into ***./results/tf_ROC.png*** (*de novo* TF prediction ROC curve), and ***./results/SNP/*.png*** (*de novo* target prediction ROC curves for each SNP).
4. To perform the allele bias transition analysis the jupyter notebook ***AlleleTransitionBias.ipynb*** should be run. The results are ROC curve graphs and these shall be saved into ***./results/BIAS/SNP/*.png*** (*de novo* target prediction ROC curves for each SNP).
5. To perform the machine learning analysis the jupyter notebook ***RandomForestForTFandTargets.ipynb*** should be run. The results are the ***./results/rf_importances_snps_joined_and_sorted.csv***

table, which contains two columns - feature name (SNP id or ORF id) and its importance for the random forest regressor; and the ***./results/rf_training_scores.csv***

RESULTS

Marker regression method resulted in a binary classifier with poor performance. The ROC AUC was close to 0.5 (no better than random chance) for both *de novo* prediction of TFs (Figure 3) and *de novo* prediction of targets for every SNP (example ROC shown on Figure 4).

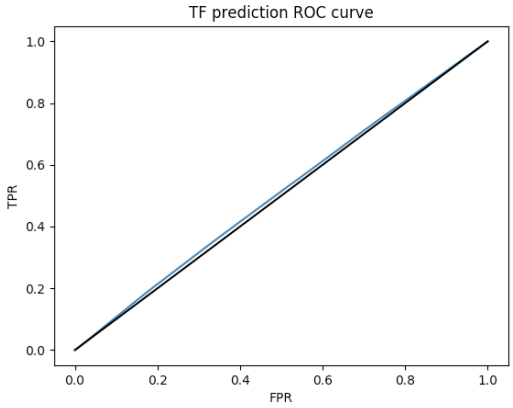

Figure 3 - ROC curve for *de novo* prediction of TFs with the marker regression method

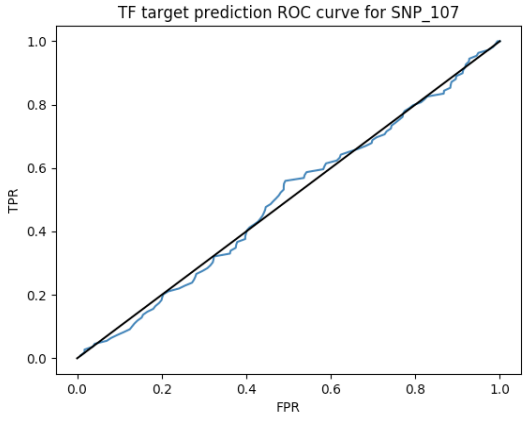

Figure 4 - Example ROC curve for *de novo prediction* of TF targets with the marker regression method

Allele transition bias method produced a classifier with similarly poor performance. Example ROC curves can be seen in Figure 5.

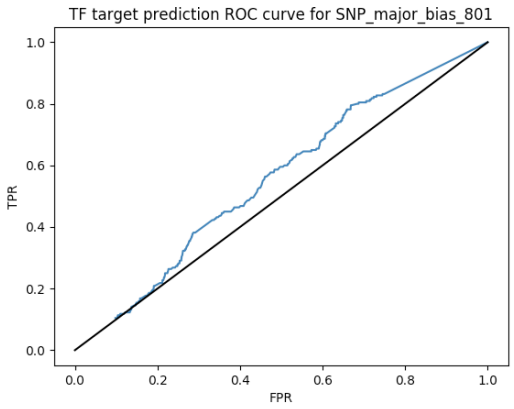

Figure 5 - Example ROC curves for *de novo prediction* of TF targets with the allele transition bias method

The random forest regressor method identified one SNP as particularly important (SNP with id 25, belonging to the Nsi1 gene). This SNP was 8th in the order of importance following seven transcription factors. Unfortunately, due to linkage disequilibrium, the genotype data for this SNP is exactly the same as 14 other SNPs, most of which belong to Nsi1 (10 SNPs); with one other belonging to Otu1; another one belonging to Stp3; and the remaining two belonging to Imp2' genes. Due to this it is unclear what exactly is the source of the elevated importance of the SNP and any interpretation would be dubious at best.

All the other SNPs ranked below all of the TFs. We selected eight other SNPs in decreasing order of importance, in order to see whether the performance of the regressor could be improved with additional input data. The performance metric (R^2) slightly decreased as, first all the SNPs were introduced in the input data, and then slightly increased when just the important SNPs were introduced in the input data. The average testing scores over k=10 folds of cross validation are shown in Figure 6. Among the eight high ranking SNPs, one was identified as having no exact copies resulting from linkage disequilibrium (SNP with id 1643, belonging to Cup2 gene). Cup2 gene only contained this one SNP and this SNP was located in its DNA binding domain.

	rf_test_score
TF_only	0.102488
TF_all_SNPs	0.100261
TF_important_SNPs	0.100967

Figure 6 - Random forest regression test scores for the three conditions of input data - TF expression only, TF expression with all unique SNPs (within TFs), TF expression with only important SNPs.

DISCUSSION

Unfortunately, none of the three methods produced interesting results. However, SNP with 1643 identified via the random forest regression method as important for predicting the expression values of the target genes did prove to be in the DNA binding region of its corresponding gene, which might indicate some value for this method. Regrettably, due to high linkage disequilibrium even when a SNP vector is identified as important it might be impossible to deduce which exact SNP confers the most importance.

LIMITATIONS AND FUTURE WORK

The analysis presented here was severely limited by the number of available SNPs within known transcription factors (111 total SNPs, 30 unique SNPs after accounting for linkage disequilibrium). Linkage disequilibrium complicated the inference and relation to TF SNP importance in explaining the expression data.

COMMENTS

The project was very challenging but also extremely interesting at the same time. We learned a lot more about pandas, we learned that genomic and transcriptomic data can be very messy and should be always checked. We also learned the importance of keeping up with statistics, especially more advanced methods. Most of all, we learned the importance of doing preliminary analysis… in the beginning, not the middle of the project. If we could start over, we would validate the findings of the original paper first, and definitely start with simpler analyses.

REFERENCES

[1] R. B. Brem and L. Kruglyak. The landscape of genetic complexity across 5,700 gene expression traits in yeast. Proceedings of the National Academy of Sciences of the United States of America, 102(5):1572–1577, 2005.

[2] Huang, Yang, Wuchty, Stefan & Przytycka, Teresa M (2013). eQTL epistasis--challenges and computational approaches. Frontiers in genetics, 4, 51.

[3] Shabalin, Andrey A (2012). Matrix eQTL: ultra fast eQTL analysis via large matrix operations. Bioinformatics, 28, 1353-1358.

[4] L. F. Thomas and P. Strom. Single nucleotide polymorphisms can create alternative polyadenylation signals and affect gene expression through loss of microrna-regulation. PLOS Computational Biology, 8(8):1–12, 08 2012.

[5] R. Xie, A. Quitadamo, J. Cheng, and X. Shi. A predictive model of gene expression using a deep learning framework. In Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on, pages 676–681. IEEE, 2016.

[6] Wright, Fred A, Strug, Lisa J, Doshi, Vishal K, Commander, Clayton W, Blackman, Scott M, Sun, Lei, Berthiaume, Yves, Cutler, David, Cojocaru, Andreea, Collaco, J Michael & others (2011). Genome-wide association and linkage identify modifier loci of lung disease severity in cystic fibrosis at 11p13 and 20q13. 2. Nature genetics, 43, 539.