Fonseca Lab
Research Institute of the McGill University Health Centre

Inferring Transcriptional Regulatory Networks via Elastic Net Regression

Orsolya Lapohos

November 7, 2023

Contents

1 Defining Transcriptional Regulatory Networks

TRNs are bipartite graphs that separate nodes into transcription factors (TFs) and target genes, and their edges represent regulatory effects between the two types of nodes (see Figure 1). The value in inferring such networks lies in the identification of potential upstream mediators that can explain mechanisms driving disease. TRN inference can build on a differential gene expression analysis by finding associations between TFs and genes that are differentially expressed in two different contexts.

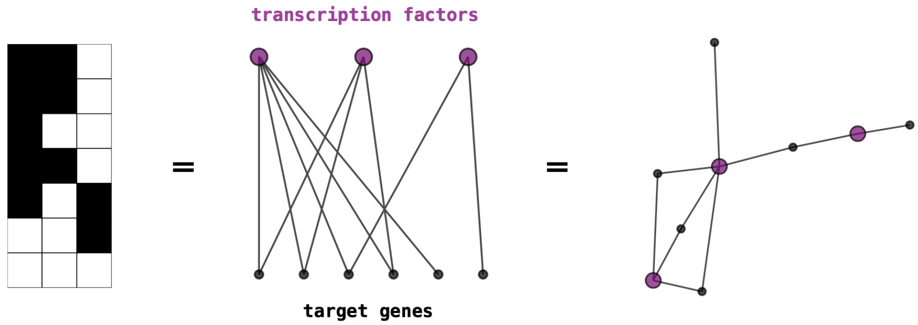

Figure 1: Three representations of the same TRN. Left: adjacency matrix with each column representing a TF and each row representing a target gene. A filled cell indicates the existence of a relationship or "edge" between a TF and target gene. Center: bipartite graph with TFs as large purple nodes and target genes as small black nodes. Edges connect only TFs and targets. Right: a "spring" layout of the same bipartite graph.

2 Objective

To infer relationships between TFs and target genes that may explain mechanisms driving differences between two groups or phenotypes.

3 Methods

3.1 General Principle

To infer weights between transcription factors (TFs) and target genes, we regress each target gene against a set of TFs, for all target genes of interest. Figure 2 shows a high-level schematic of the implementation.

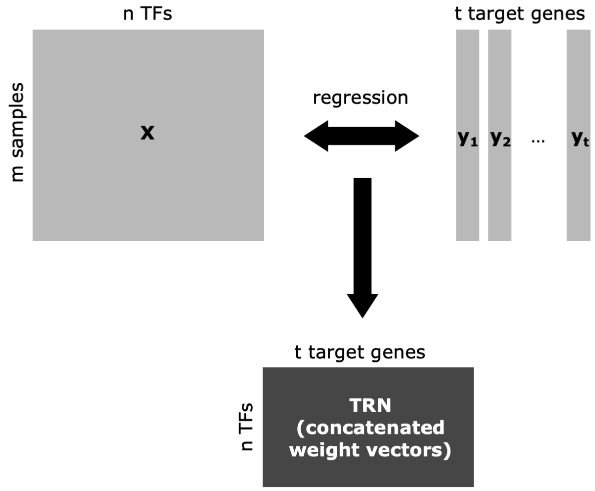

Figure 2: The regression process is repeated for each target gene. Obtained weight vectors are concatenated to form the full adjacency matrix (TRN).

The matrix multiplication shown below represents a single regression task (simplified to ordinary least squares), where $\mathbf{X}$ denotes the matrix of gene expression values corresponding to m samples and n TFs, $\mathbf{w}$ denotes the vector of weights inferred for each of n TFs, and $\mathbf{y}$ denotes the vector of gene expression values corresponding to m samples and one target gene.

$$\underset{(m\times n)}{\mathbf{X}} \quad \underset{(n\times 1)}{\mathbf{w}} \quad = \quad \underset{(m\times 1)}{\mathbf{y}}$$

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,n} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

In other words, we model the expression of a target gene k in sample s as a weighted sum of n TFs' gene expression values:

$$y_{s,k} = \sum_{i=1}^{n} w_i x_{s,i} \tag{1}$$

3.2 Candidate Node Selection

To reduce the dimensionality of the regression problem, we select a workable set of candidate TRN nodes.

3.2.1 Target Gene Selection

Target genes are selected via differential gene expression analysis comparing the two groups or phenotypes of interest. If the number of differentially expressed genes is large, a more stringent log fold change cutoff is considered.

3.2.2 TF Selection

Candidate TF selection is more complicated than target gene selection, due to the fact that TFs are typically lowly expressed and therefore missed in differential gene expression analyses. Subtle changes in their expression can greatly affect their target genes.

When paired ATAC-seq data is available for the RNA-seq data in question, a global motif enrichment analysis is first performed on the open regions defined by ATAC-seq peaks. Under the assumption that globally enriched motifs are more likely to drive a global phenotype, enriched motifs are paired with their corresponding TF families. The set of candidate TFs is further filtered by a minimum expression cutoff in the paired RNA-seq data.

Without ATAC-seq data, two approaches are possible. One approach involves global motif enrichment analysis using arbitrarily-defined promoter regions of the selected set of target genes, followed by the same procedure described in the previous paragraph. The other approach simply ranks the detected TFs' expression values by variance, and applies a cutoff. The former option is preferred, as incorporation of prior knowledge is expected to enhance results.

3.3 Elastic Net Regression

In order to reconstruct biologically relevant TRNs, we consider two fundamental properties of biological networks: sparsity and scale-free topology. Living systems achieve robustness to perturbation via sparsity—by restricting the number of connections. More specifically, the number of active interactions in living systems tends to scale inversely with network size [1]. Furthermore, scale-free topology means that connectivity of nodes in a network follow a power-law distribtion, such that central "hubs" oversee the majority of regulatory effects [2]. Considering these properties and the small sample size given, Elastic Net regression is a good candidate for inferring TRNs.

Elastic Net is a modified form of ordinary least squares (OLS) regression in which two penalties (Lasso and Ridge) are added to the mean squared error (MSE) loss function (Eqn. 2). The Lasso penalty imposes sparsity, while the Ridge penalty shrinks coefficients. Further, both penalties contribute to the elimination of multicollinearities and allow coefficient estimation with fewer samples [3]. The loss function for this regression model can be expressed:

$$\frac{1}{2n}\|\mathbf{y}-\mathbf{X}\mathbf{w}\|_2^2 \;+\; \alpha\rho\|\mathbf{w}\|_1 \;+\; \frac{1}{2}\alpha(1-\rho)\|\mathbf{w}\|_2^2 \tag{2}$$

where $\mathbf{X}$ is the matrix of TF expression, $\mathbf{y}$ is a vector of target gene expression, and $\mathbf{w}$ is the vector of weights (coefficients) to optimize. The first term in this function represents the MSE, the second term represents the Lasso penalty (L_1 norm), and the third term represents the Ridge penalty (L_2 norm). The α hyperparameter controls the level of penalization while ρ is the mixing parameter between the two penalty terms. In our approach, ElasticNetCV from the scikit-learn [4] python package is used to optimize α for each target gene by cross-validation, and ρ is set to 0.5.

3.4 Implementation & Analysis

The input values to the pipeline should be gene length-normalized expression values. These values are log-transformed and the features are then scaled to a standard Gaussian distribution. Using the selected set of TFs and target genes, samples are divided into two groups (by phenotype or binary variable) before inferring corresponding TRNs by Elastic Net regression. In the inferred group/phenotype-specific TRNs, a useful analysis involves comparing the change in regulon size of a TF. Edge weights may also be ranked to extract the most pronounced regulatory effects.

The absolute differences between the two resulting TRNs (adjacency matrices) are then calculated in order to extract the relationships with the greatest change between the two groups. This is referred to as the “differential” TRN. In the differential TRN, a large change in magnitude of a TF’s regulatory effects suggests that it has phenotype-specific roles and is a good target for biological validation experiments.

4 Appendix: RNA-seq normalization prior to TRN inference

Let's examine how the weights might be affected by the normalization method applied to the gene expression values prior to regression. We can substitute the variables $y_{s,k}$ and $x_{s,i}$ in Equation 1 to see their effects. First, let's define the 3 main normalization methods for RNA-seq data.

CPM (counts per million mapped reads) normalizes only for library size. The CPM of a gene k in sample s can be expressed:

$$\mathrm{CPM}_{s,k} = \frac{q_{s,k}}{\sum_{j=1}^{a} q_{s,j}} \cdot 10^6$$

where q is the number of raw reads mapped to a gene, and a is the total number of genes detected.

RPKM (reads per kilobase of transcript per million mapped reads) normalizes both for library size and gene length. The RPKM of a gene k in sample s can be expressed:

$$\mathrm{RPKM}_{s,k} = \frac{q_{s,k}/l_k}{\sum_{j=1}^{a} q_{s,j}} \cdot 10^9$$

where l is the length of a gene.

TPM (transcripts per million mapped reads) normalizes both for library size and gene length, but in a different order. The TPM of a gene k in sample s can be expressed:

$$\mathrm{TPM}_{s,k} = \frac{q_{s,k}/l_k}{\sum_{j=1}^{a} q_{s,j}/l_j} \cdot 10^6$$

Now, we can observe what happens when we substitute CPM, RPKM, or TPM values into the linear equation.

CPM:

$$\begin{aligned} y_{s,k} &= \sum_{i=1}^{n} w_i x_{s,i} \\ \mathrm{CPM}_{s,k} &= \sum_{i=1}^{n} w_i \mathrm{CPM}_{s,i} \\ \frac{q_{s,k}}{\sum_{j=1}^{a} q_{s,j}} \cdot 10^6 &= \sum_{i=1}^{n} w_i \frac{q_{s,i}}{\sum_{j=1}^{a} q_{s,j}} \cdot 10^6 \\ q_{s,k} &= \sum_{i=1}^{n} w_i q_{s,i} \end{aligned}$$

RPKM:

$$
\begin{aligned}
y_{s,k} &= \sum_{i=1}^{n} w_i x_{s,i} \\
\mathrm{RPKM}_{s,k} &= \sum_{i=1}^{n} w_i \mathrm{RPKM}_{s,i} \\
\frac{q_{s,k}/l_k}{\sum_{j=1}^{a} q_{s,j}} \cdot 10^9 &= \sum_{i=1}^{n} w_i \frac{q_{s,i}/l_i}{\sum_{j=1}^{a} q_{s,j}} \cdot 10^9 \\
q_{s,k}/l_k &= \sum_{i=1}^{n} w_i q_{s,i}/l_i
\end{aligned}
$$

TPM:

$$
\begin{aligned}
y_{s,k} &= \sum_{i=1}^{n} w_i x_{s,i} \\
\mathrm{TPM}_{s,k} &= \sum_{i=1}^{n} w_i \mathrm{TPM}_{s,i} \\
\frac{q_{s,k}/l_k}{\sum_{j=1}^{a} q_{s,j}/l_j} \cdot 10^6 &= \sum_{i=1}^{n} w_i \frac{q_{s,i}/l_i}{\sum_{j=1}^{a} q_{s,j}/l_j} \cdot 10^6 \\
q_{s,k}/l_k &= \sum_{i=1}^{n} w_i q_{s,i}/l_i
\end{aligned}
$$

We can see that with CPM, gene length does not influence the weights. This may be problematic since inferred weights between longer target genes and shorter TF-encoding genes would be disproportionally large.

With RPKM and TPM, we can see that gene length is properly accounted for. Further, despite the difference in the order of operations (normalization for gene length and library size), both RPKM and TPM simplify to the same expression describing the inferred weights.

5 References

[1] Daniel M. Busiello, Samir Suweis, Jorge Hidalgo, and Amos Maritan. Explorability and the origin of network sparsity in living systems. *Scientific Reports*, 7:1–8, 9 2017.

[2] Albert László Barabási and Zoltán N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5:101–113, 2 2004.

[3] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67:301–320, 4 2005.

[4] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.