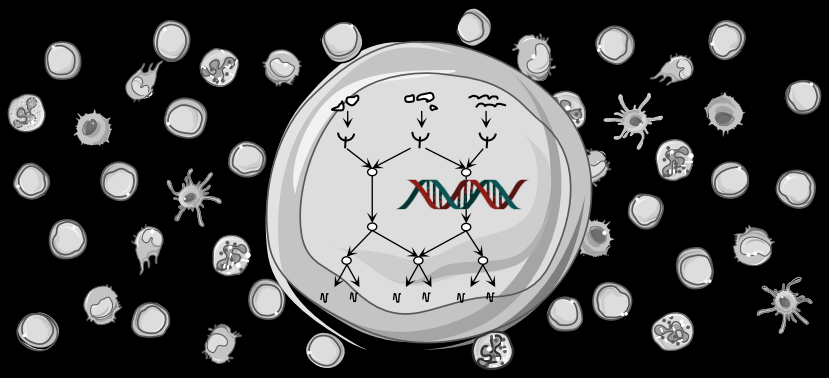


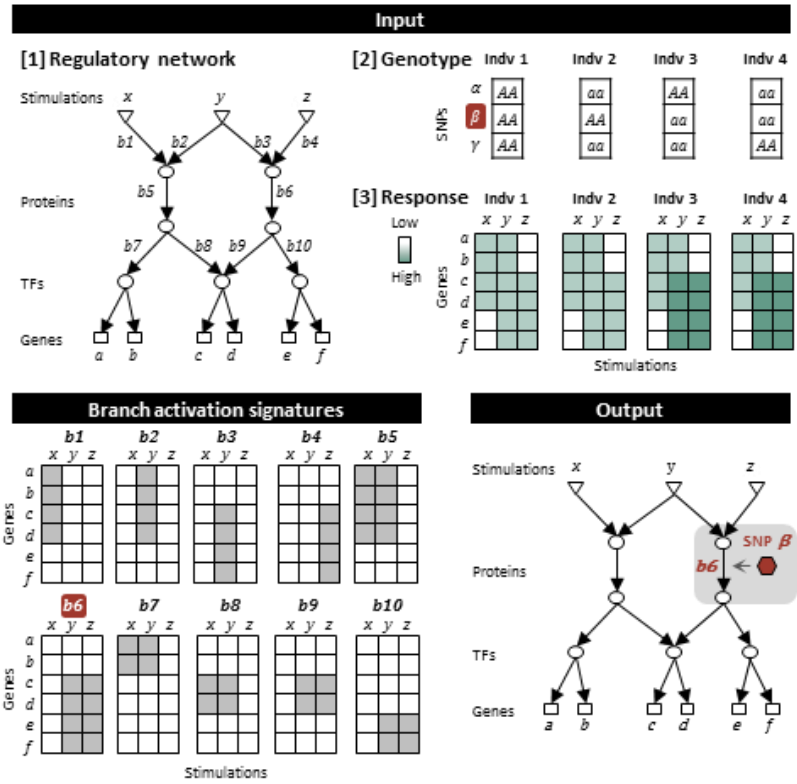
GEVIN algorithm manual

Reconstructing the molecular function of genetic variation in regulatory networks

Roni Wilentzik[1], Chun Jimmie Ye[2], Irit Gat-Viks[1]

[1] Department of Cell Research and Immunology, the George S. Wise Faculty of Life Sciences, Tel Aviv University, Israel

[2] Department of Epidemiology and Biostatistics, University of California, San Francisco, USA



1. Overview

GEVIN (Genome-Wide Embedding of Variation In Networks) is a methodology for identifying reQTLs (response quantitive trait loci) that affect gene regulation in response to triggering stimulations, along with the particular pathway within a complex network that is perturbed by the reQTL [1].

The following manual describes the Matlab code for the GEVIN algorithm—including the input and output data, and how an input regulatory network can easily be built. Example files are based on the murine dendritic cells data from Ref [2].

2. Input data

The GEVIN algorithm takes as input three types of data:

1. **A regulatory network** derived from the scientific literature, consisting of known stimulations that trigger different signaling pathways (termed *network branches*) and regulate a group of embedded genes.
2. **Transcriptional response levels** of multiple genes measured in response to each of the triggering stimulations, across multiple individuals.
3. **Genotyping data** of multiple DNA polymorphic loci (*SNPs*) across multiple individuals.

2.1. Preparing input data: Constructing a regulatory network based on scientific literature

A regulatory network input data is provided to GEVIN algorithm as a `struct` composed of two main elements:

(i) The molecular structure of a regulatory network that propagates signals from triggering stimulations to transcription factors. This structure determines the list of branches that are tested by GEVIN for reQTL perturbations.

(ii) The gene positioning in the network, provided as a detailed list of the transcription factors (TFs) regulating each gene. Branches from a TF to a single gene will not be tested by GEVIN.

The script `BxdTlrNetworkExample.m` is an example for how the user can easily build such a network. In this script, a network (`BxdTlrNet.mat`) is constructed according to the Toll-like/RIG-1-like signaling network for the case of the murine dataset described in **Figure 4A** in Ref [1]. `<<user_start>>` and `<<user_end>>` tags are used to indicate the places within the script that should be changed by the user when building a new network: the names of the signaling components, their functions (which components are stimulations and which are TFs) and the edges from one signaling component to another. Note that the nodes, which represent singling components, should be added to the script in topological order (ancestors before descendents). When running the script a network is automatically created according to this network description. The network will include a list of all derived network branches. Note that the list of derived network branches would be printed to screen. To avoid mistakes, we recommend that this list would be manually examined in comparison with the desired structure of the network.

Genes are further added to the network according to their regulating TFs – this information is given as a `gene_embedding_file` where each row represents a gene, each column represents a transcription factor, and an entry is set to 1 if the gene is regulated by the TF, or set to zero otherwise.

2.2. Preparing input data: Transcriptional response files

Transcriptional response levels are defined as the expression levels of genes measured after a given stimulation, normalized by their expression in steady state. A separate transcriptional response file is required for each of the stimulations in the modeled network. Each file should include response measurements for all genes embedded in the network.

A transcriptional response file of a single stimulation is a $|genes| \times |individuals|$ table. An entry (i, j) is the transcriptional response level of a gene i measured in an individual j following the relevant stimulation. Titles are required for each gene (1^{st} column) and each individual (1^{st} row). The order of genes (in rows) and of individuals (in columns) should be maintained across all response files.

2.3. Preparing input data: Genotyping file

A genotyping data file is a $|SNPs| \times |individuals|$ table in which an entry takes the value of 0, 1, or 2 according to the genotype of an individual in a certain SNP. Titles are required for each SNP (1st column) and each sample (1st row). The order of the individuals (columns) should match the order of the individuals in the response files.

3. Running GEVIN: Assessing significancy of multiple SNPs to perturb network branches

GEVIN algorithm can be applied by the `MainGevinAlgorithm.m` function whose inputs are:

1. `output_file` – the name of GEVIN's results output file.
2. `net_name` – the name of the regulatory network which was created as described in section **2.1**. For example: `'BxdTlrNet'`.
3. `response_files` – a cell array listing the names of all transcriptional response files (each file details the response levels following a different stimulation). Note that the order of the file names in this cell array should match the exact order of the stimulations used in the network-construction script (see `my_net.nodes_name` in `BxdTlrNetworkExample.m`)

   In the provided example three stimulations trigger the network (PAM, LPS and Poly I:C) therefore the array would be:

   ```
   {'response_pam.xls','response_lps.xls','response_poly.xls'}
   ```
4. `genotype_file` – the name of the genotyping file. For example:`'genotypes.xls'`.
5. `zygosity` – indicates the zygosity origin of the samples; either `'homo'` or `'hetero'`. If the genetic background of the samples is homozygous genotypes of 2 would be excluded from the analysis.

4. Output data

The output of `MainGevinAlgorithm.m` is a $|SNPs| \times |branches|$ table depicting the raw scores ($P^{q,b}$; see '**Modeling SNP-branch perturbation**' section in Ref. [1]) calculated by GEVIN for each SNP (q; row) perturbing each network branch (b; column). For instance, `'BxdTlrNet_output.xlsx'` is the output file obtained by applying GEVIN on the example input files.

As initial analysis we suggest to examine the Manhattan plot of each branch (plotting the log-transformed scores obtained for a single branch across all SNPs in chromosomal order). Peaks in this plot suggest significant perturbations of certain reQTLs within the tested branch (see 'False discovery rate' section below).

5. More about GEVIN analysis

- **Running time** – The GEVIN algorithm was designed as a genome-wide algorithm that can analyze multiple genes. On an average computer GEVIN calculated the results of 100 SNPs in approximately 2 minutes. When applying GEVIN on thousands of SNPs we suggest to split the genotyping file to multiple chunks (e.g., files of 500 SNPs each) and to apply GEVIN in parallel running.
- **Inflation correction** – `MainGevinAlgorithm.m` outputs the raw $P^{q,b}$ scores calculated by GEVIN. We suggest that the user would further apply GEVIN on permuted genotyping data in order to correct the results of each branch using genomic inflation factor (normalizing the observed data by the expected data). For more information see '**Genome-wide analysis of genetic perturbations within the signaling network**' section if Ref. [1].
- **False discovery rate (FDR)** – permuted-based FDR thresholds for each branch can further be calculated by applying GEVIN on permuted genotypes in order to determine the significance of perturbations. For more information see **'Genome-wide analysis of genetic perturbations within the signaling network**' section if Ref. [1].
- **Confounding factors** – if the user is interested in controlling for confounding effects (e.g., age, ethnicity, sex etc.) we suggest conducting a pre-processing step of principal component regression and use the residuals as response levels when applying GEVIN. The

factors can also be added to the multivartiate regression as explaining variables (a strategy that might increase the running time of the algorithm). For more information see '**Analysis of biological data; Human data**' section if Ref. [1].

- **Removing outliers** – we recommend removing outlier individuals that poorly correlate with the other individuals before applying the GEVIN algorithm.

Feedback

For any questions, bugs or suggestions please contact: roniwile@post.tau.ac.il or iritgv@post.tau.ac.il.

Reference

[1] Wilentzik, R., et al., Reconstructing the molecular function of genetic variation in regulatory networks. (submitted).

[2] Gat-Viks, I., et al., Deciphering molecular circuits from genetic variation underlying transcriptional responsiveness to stimuli. Nat Biotechnol, 2013. 31(4): p. 342-9.