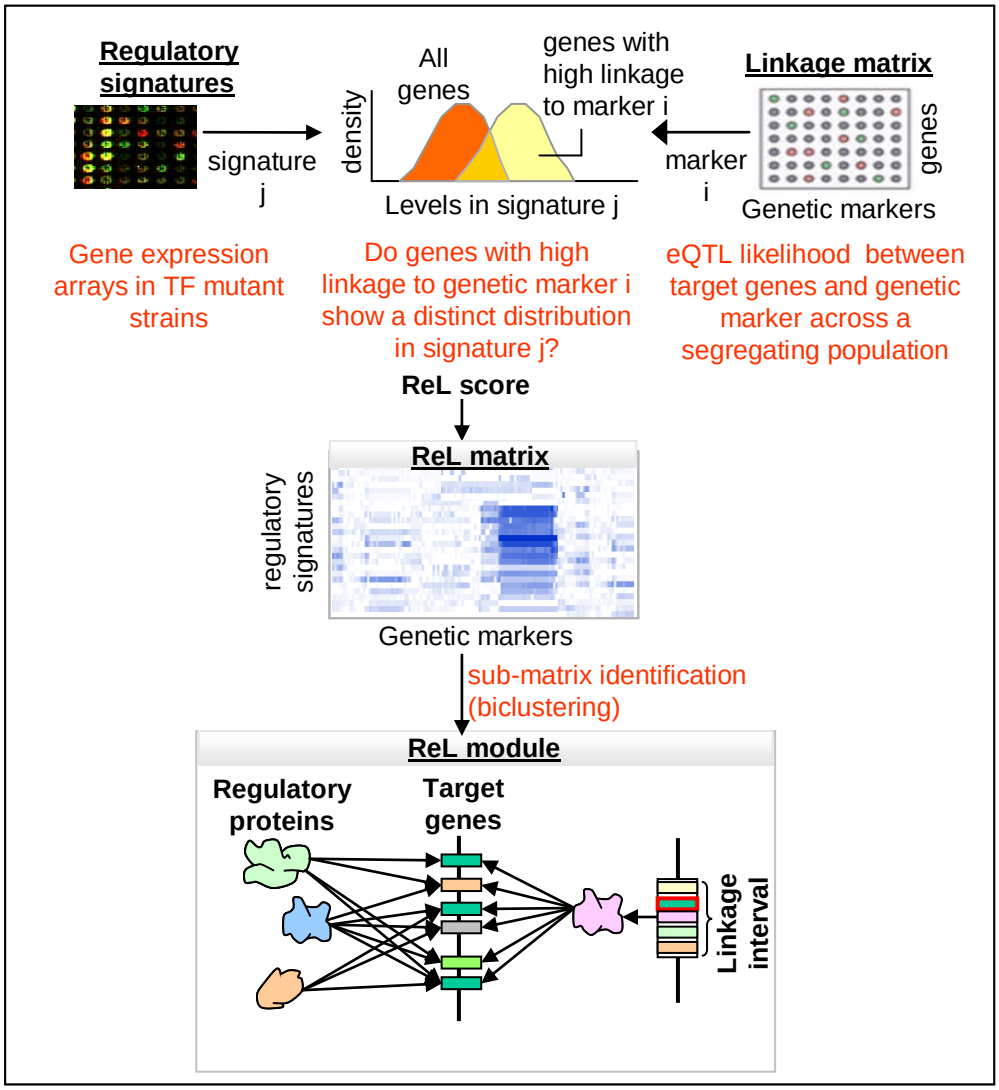
ReL package
Quick start manual



Understanding gene sequence variation in the context of transcription regulation in yeast

Irit Gat-Viks, Renana Meller, Martin Kupiec, Ron Shamir

1. Quick start

The following manual describes how to create ReL modules using default parameters. The detailed manual should be used in a case of a more sophisticated usage.
The package runs on 64 bit Linux environment and requires JRE version 1.5.0_05 or higher and c library libc.so.6 with Glibc 2.4 or higher.

The c shell script RunFlow.csh runs the entire flow automatically. The input is the compendium file, linkage matrix, and several organism-specific files (see below). The output is a collection of ReL modules, each consists of a set of regulatory proteins, a set of target genes, and a single linkage interval.

Step1: Download ReLPackage.zip and extract it. Make sure that you have rwx permissions for the entire ReLPackage directory.

Step 2: Edit RunFlow.csh: The variable REL_PACKAGE_ROOT should be set to the full path in which you have extracted the zip file.
For example:

```
## Update the root directory and required environment variable.
setenv REL_PACKAGE_ROOT "<YOUR_HOME_FOLDER>/ReLPackage"
```

Step 3: Run RunFlow.csh as follows:

```
> RunFlow.csh <Input compendium file> <Organism directory> <Output
file> <Log file>
```

Input compendium file – the compendium of regulatory signatures.
Organism directory – the directory that includes the linkage matrix and all other organism-specific files.
Output file – the output file (a collection of ReL modules).
A Log file

Input and output files specification appears in sections 2 and 3. An example of input files for the yeast system can be found in the Organism/Yeast directory and in the Compendium/RegulatorySignatures.txt file. An example output file can be found in Output/ReLModule.xls. The complete output file could be generated as follows:

```
> RunFlow.csh Compendium/RegulatorySignatures.txt Organism/Yeast
Output/ReLModules.txt RunFlow.log
```

Additional comments:
- For the yeast's linkage matrix and 100 signatures, the ReL package calculates modules within 3-4 hours on a standard linux machine.
- Intermediate files are created in the Output directory and may be ignored.

2. Organism designated files

Four organism-specific files are required: LinkageMatrix.txt, GenomicFeatures.txt, GeneticMarkersDictionary.txt and ProteinDifferences.txt. Yeast files are available in the Organism/Yeast directory. If you wish to work with a different organism, create your own directory under the Organism folder and prepare four input files with the same names within your directory. Importantly, apply the dos2unix command on all input files.

2.1 Linkage matrix file

The linkage matrix contains eQTL likelihood score for each pair of gene and genetic marker.
There are several requirements for the linkage matrix file:

- The file should be tab-delimited file
- The rows should be genetic marker ids and the columns should be genes. If this is not the case, use the Transpose.pl script found in the PerlScripts directory to transpose the file.
  `> perl Transpose.pl <Input file> <output file>`
- The genetic markers ids are sequential numbers that matches the ids as appear in GeneticMarkersDictionary.txt.
- Missing or unknown values should have the value -1000.

2.2 Genomic features file

The genomic feature file is a tab delimited file that contains data about features along the genome. The genomic feature file may contain several feature types, however only "ORF" and "telomere" features will be read from the file. The required structure of the file:

Column #	Data
1	feature type (we read only "ORF" and "telomere" features)
2	empirical evidence (e.g., Dubious, Uncharacterized, Verified)
3	gene name (e.g., ORF name in yeast)
4	alias name 1
5	alias name 2
8	chromosome
9	start position
10	end position
11	C/W strand
15	Description

(Column # starts from 0). All the other columns will be ignored.

2.3 Genetic markers dictionary

The genetic market dictionary is a tab delimited file that matches genetic marker id to a genomic locus. Missing or unknown values should have the value "none". The genetic marker dictionary has a header line and its required structure is:

Column #	Data
0	marker id (sequential number)
2	ORF name
3	Chromosome
4	genomic position

All the other columns will be ignored.

2.4 Protein differences file (optional)

The protein differences file is a tab-delimited file, which contains the protein difference between the two parental strains. The required structure of the file:

Column #	Data
0	gene name of the first strain (the same gene name as in GenomicFeatures.txt)
1	the gene name of the ortholog protein in the second strain
2	a textual description of the differences between the proteins

All the other columns will be ignored.

The protein differences file is not mandatory. Instead, it is possible to create an empty ProteinDifferences.txt file.

3. Regulatory signatures (compendium) file

A tab-delimited file of –log (intensity ratio). Rows are regulatory signatures and columns are genes. Missing or unknown values should get the value -1000. An example can be found in Compendium/RegulatorySignatures.txt.

4. ReL modules output file

To illustrate the structure of the output file we provide excel spreadsheet that contains example of a ReL module with conditionally formatting and comments. The spreadsheet - ReLModule.xls - could be found in the Output directory. You can copy the conditional formatting using the excel paste special command to your output file. (First paste format to column A and then to columns B-N.)