explicit

Got a list of genes and you wonder what TFs regulate their expression? EXPLICIT is the right tool to try.

The EXPLICIT approach has been developed to construct a gene expression predictor model for the plant species *Arabidopsis thaliana*. The predictor uses the expression of 1,678 transcription factor (TF) genes to predict the expression of 29,182 non-TF genes. It further enables downstream inference of TF regulators for genes and gene modules functioning in diverse plant pathways. Please check the original paper by Geng *et al.* for more details. The EXPICIT package presented here enables users to: 1. Infer TF regulators for their own gene modules; 2. Draw chord diagrams showing TF-target genes regulation for the modules; 3. Create custom gene expression predictor using their own gene expression data. (*Note: below is an example showing the analysis flow-chart for a gene module involved in vascular system development.*)

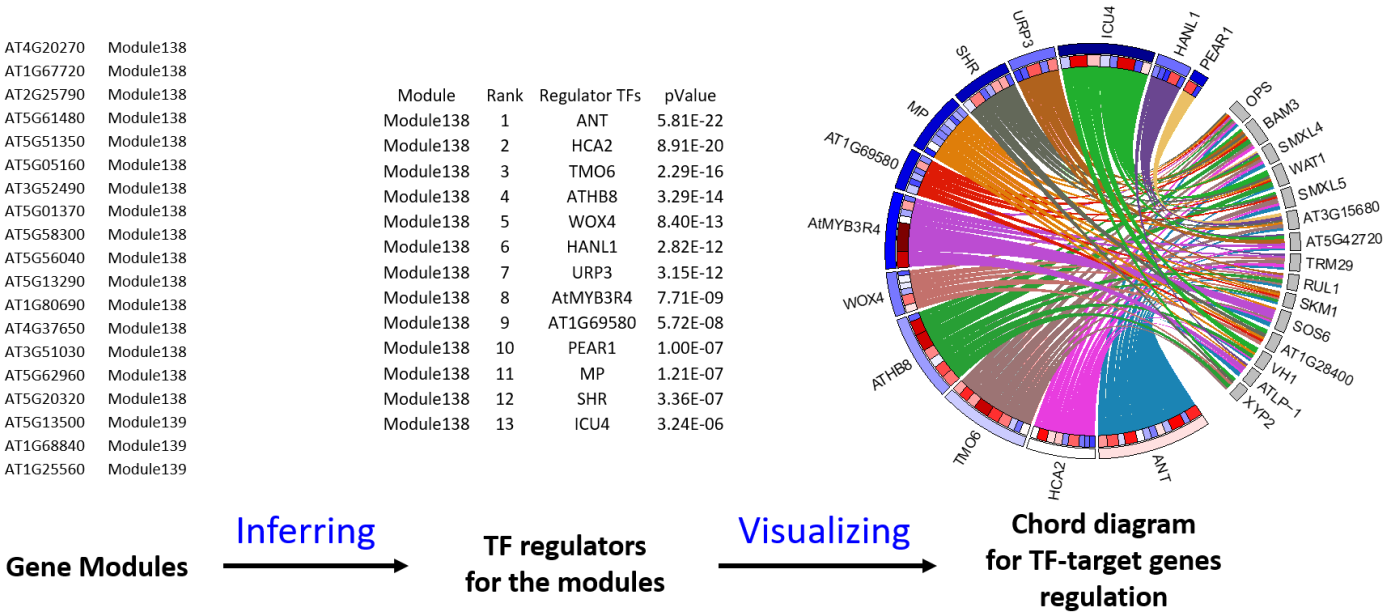


Table of Contents

- Additional Information
- Reference

Install

This package requires Perl, R, and the circlize package in R.
circlize can be installed within an R console via the command:

```
install.packages("circlize")
```

MATLAB is optional. Only required if you want to create your own predictor model using custom expression data.

Once the required softwares are installed, just download or clone the whole package to a local computer and start using it from the package's home directory.

Usage

1. Infer TF regulators for gene modules

a. Prepare the module file

The file used to store gene modules information is "modules_to_analyze.txt". Edit the file with your own modules. The file has following format, with the first column being gene ids and the second column being module names. The two columns are separated by a tab. For gene ids, only standard Arabidopsis AGI ids are currently supported. Multiple modules can be analyzed at the same time. Once finished editing, save the file without changing its name.

```
Gene_Name   ModuleID
AT1G25360   Module138
AT2G22340   Module138
AT5G75660   Moudle138
AT2G22130   Module139
AT4G12350   Module139
.........   ........
```

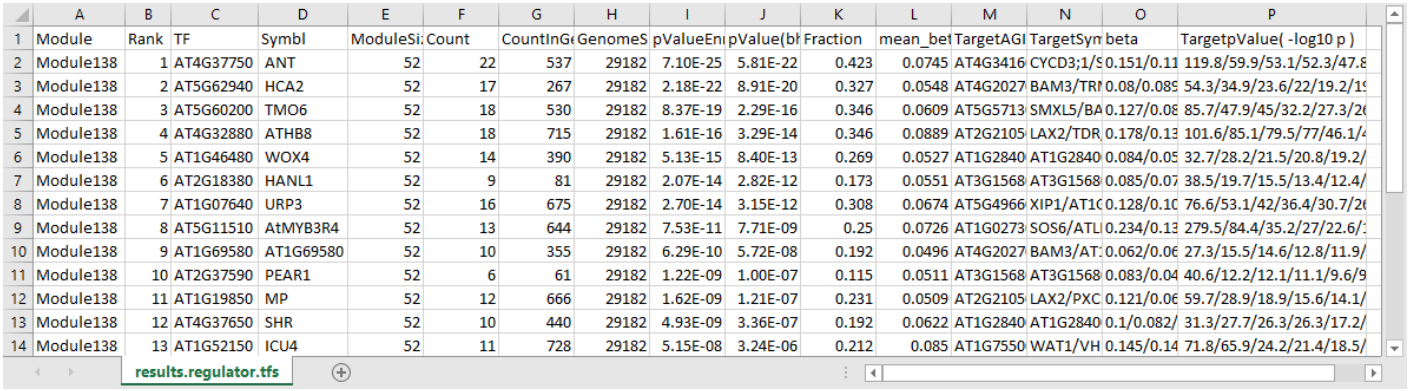
b. Conduct enrichment assay to identify TF regulators for the modules

The Perl script "getArabidopsisRegulatorTFs.pl" will do the job. It takes the modules from the file "modules_to_analyze.txt" to conduct an enrichment assay to identify potential TF regulators. Results are outputted to a file named "results.regulator.tfs.txt", which can be open in EXCEL.

The command to use:

```
perl getArabidopsisRegulatorTFs.pl
```

Here is an example of the output results:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Module	Rank	TF	Symbl	ModuleSi	Count	CountInG	GenomeS	pValueEn	pValue(b	Fraction	mean_bet	TargetAGI	TargetSym	beta	TargetpValue(-log10 p)
2	Module138	1	AT4G37750	ANT	52	22	537	29182	7.10E-25	5.81E-22	0.423	0.0745	AT4G3416	CYCD3;1/S	0.151/0.11	119.8/59.9/53.1/52.3/47.8
3	Module138	2	AT5G62940	HCA2	52	17	267	29182	2.18E-22	8.91E-20	0.327	0.0548	AT4G2027	BAM3/TRI	0.08/0.085	54.3/34.9/23.6/22/19.2/1
4	Module138	3	AT5G60200	TMO6	52	18	530	29182	8.37E-19	2.29E-16	0.346	0.0609	AT5G5713	SMXL5/BA	0.127/0.08	85.7/47.9/45/32.2/27.3/2
5	Module138	4	AT4G32880	ATHB8	52	18	715	29182	1.61E-16	3.29E-14	0.346	0.0889	AT2G2105	LAX2/TDR	0.178/0.13	101.6/85.1/79.5/77/46.1/
6	Module138	5	AT1G46480	WOX4	52	14	390	29182	5.13E-15	8.40E-13	0.269	0.0527	AT1G2840	AT1G2840	0.084/0.05	32.7/28.2/21.5/20.8/19.2/
7	Module138	6	AT2G18380	HANL1	52	9	81	29182	2.07E-14	2.82E-12	0.173	0.0551	AT3G1568	AT3G1568	0.085/0.07	38.5/19.7/15.5/13.4/12.4/
8	Module138	7	AT1G07640	URP3	52	16	675	29182	2.70E-14	3.15E-12	0.308	0.0674	AT5G4966	XIP1/AT1(	0.128/0.1(	76.6/53.1/42/36.4/30.7/2
9	Module138	8	AT5G11510	AtMYB3R4	52	13	644	29182	7.53E-11	7.71E-09	0.25	0.0726	AT1G0273	SOS6/ATL	0.234/0.13	279.5/84.4/35.2/27/22.6/
10	Module138	9	AT1G69580	AT1G69580	52	10	355	29182	6.29E-10	5.72E-08	0.192	0.0496	AT4G2027	BAM3/AT	0.062/0.06	27.3/15.5/14.6/12.8/11.9/
11	Module138	10	AT2G37590	PEAR1	52	6	61	29182	1.22E-09	1.00E-07	0.115	0.0511	AT3G1568	AT3G1568	0.083/0.04	40.6/12.2/12.1/11.1/9.6/9
12	Module138	11	AT1G19850	MP	52	12	666	29182	1.62E-09	1.21E-07	0.231	0.0509	AT2G2105	LAX2/PXC	0.121/0.06	59.7/28.9/18.9/15.6/14.1/
13	Module138	12	AT4G37650	SHR	52	10	440	29182	4.93E-09	3.36E-07	0.192	0.0622	AT1G2840	AT1G2840	0.1/0.082/	31.3/27.7/26.3/26.3/17.2/
14	Module138	13	AT1G52150	ICU4	52	11	728	29182	5.15E-08	3.24E-06	0.212	0.085	AT1G7550	WAT1/VH	0.145/0.14	71.8/65.9/24.2/21.4/18.5/

results.regulator.tfs

2. Draw chord diagrams showing TF-target genes regulation for the modules

a. Obtain the chord-list file for a module of interest

The Perl script "getChordLists.pl" will extract the TF-target gene pairs from the "results.regulator.tfs.txt" for module specified. By default, it will take the top 50 TFs and top 15 target genes. The results are outputted to a file named "chord.lists.txt", which will be used in the next step to draw chord diagram.

```
perl getChordLists.pl XXXXX
```

Replace XXXXX with the name of the module.

b. Draw the chord diagram according the the chord-list in R, using the circlize package

The circlize package in R will be used to draw the chord diagram showing the TF-target genes interaction, as specified in the "chord.lists.txt". Open an R console and navigate to the home directory of the explicit package, which contains the "chord.lists.txt" file. Within the R console, type the following commands:

```
source("Rscripts.R")
library("circlize")
drawChordDiagram(chordfile = "chord.lists.txt", ratio = 1)
drawChordDiagram(chordfile = "chord.lists.txt", ratio = 0.6)
```

Ratio specifies the relative size the target gene area occupy. You can repeat step a and b to draw diagrams for another module.

c. Use a single command in R to draw the diagram

One can also directly issue the following command within a R console to draw a Chord diagram for a module:

```
source("Rscripts.R")
library("circlize")
directChordDiagram( module="XXXXX", ratio = 1, tfnum = 50, targetnum = 15)
```

tfnum and targetnum specify, respectively, the maximum number of TFs and target genes to be included within the chord Diagram.

3. Create custom gene expression predictor

Currently, we have only gene expression predictor model for *Arabidopsis thaliana*. We are working on predictor models for other specices. At the same time, you can also create your won custom gene expression predictor. However, a large number of training samples are required for taining the model. The number should be at least 5 - 10 times larger than the number of input TFs.

The MATLAB function **explicit**, as specified within the file "explicit.m", will be used to create the model. The file can be found within the data folder. MATLAB is required for this analysis.

```
mdl = explicit( TF_expression, TG_expression, TF_name, TG_name)
```

TF_expression: the expression matrix for TF, with rows representing samples and columns representing genes
TG_expression: the expression matrix for target genes, with rows representing samples and columns representing genes
TF_name: the names of the TF genes
TG_name: the names of the target genes

Additional Information

We have constructed an Arabidopsis gene co-expression network based on the graphical Gaussian model (GGM) in our orginal paper by Gene *et al.*. 1,085 gene co-expression modules were identified from the network. These modules are listed within the file "AtGGM2020.gene.modules.txt". You can find this file inside the data directory. It can be used as an example input to infer up-stream TF regulators.

Reference

Will update soon.