Motif Discovery, Selection, and Post-Analysis Manual

List of projects:

OVAC, two sets of DEGs, **Yichao**

HRGP, pollen-specific genes, **Dr. Welch**

Brugia Malayi, L3 stage specific genes, **Yichao**

Chagas disease, stage specific, cell type specific, **Prashant**

Shigella, High iron vs. Low iron, **Dr. Drews?**

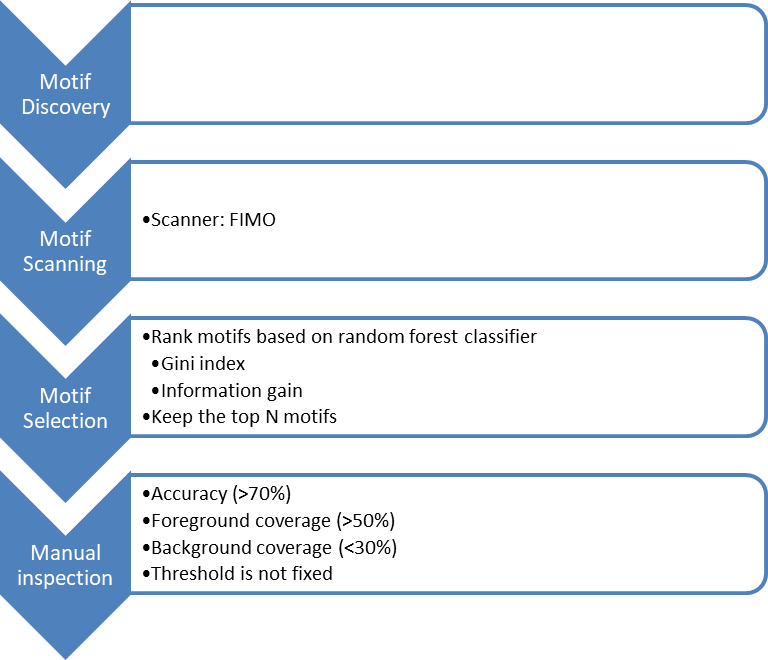
Diabetic nephropathy, (what condition?), **Dr. Drews?**

Endocrine, (what condition?), **Yichao**

1. General problem

The binding of transcription factors to regulatory elements is the key step in gene regulation. Motif discovery is the computational approach for finding the putative transcription factor binding sites.

1. Motif discovery and selection pipeline



Foreground sequences

Background sequences

Discovered motifs

Motif Discovery Ensembles: Gimmemotifs, info-gibbs\*, DME, DECOD, gkm-SVM\*

\* not in this manual

1. Dependencies
2. GimmeMotifs

GimmeMotifs is a collection of motif discovery tools. The most straightforward way to install GimmeMotifs (<https://github.com/simonvh/gimmemotifs> ) is:

$ conda install gimmemotifs -c bioconda

Of course, you first need to install **conda**.

Weeder (a motif discovery tool) is not included in GimmeMotifs, you need to follow the instructions in the website.

1. DME (<https://github.com/smithlabcode/dme> )

After installation, please put “dme2” in your $PATH.

1. DECOD (<http://www.sb.cs.cmu.edu/DECOD/> )

The reference paper said it is fast, it is not!

1. Python scikit-learn (<http://scikit-learn.org/stable/> )

A popular machine learning library in python.

1. R ggplot2 (<https://github.com/hadley/ggplot2> )

Please install the development version.

# install.packages("devtools")

devtools::install\_github("hadley/ggplot2")

1. MEME suite (<http://meme-suite.org/doc/install.html?man_type=web> )

You will use **meme**, **meme2images**, **fimo**, and **mast**.

1. BioPython (<http://biopython.org/DIST/docs/install/Installation.html> )
2. Install Emotif (Ensemble motif)

Emotif is a python program that takes foreground & background sequences and predicts motifs. It does motif discovery, motif scanning and motif selection (in the above pipeline) and output an html report for the top ranked motifs. It was created last October, which I combined several scripts from Rami and wrote some new functionalities.

To install Emotif, please first install all the dependencies.

Access Emotif\_alpha at : <http://www.ohio.edu/people/yl079811/Motif_discovery_manual/Emotif_alpha.tar.gz>

Then follow the readme file to install Emotif.

1. A toy example for Emotif

I have put the stress-response gene set as a test case with Emotif. To run the test case:

Emotif\_alpha –copy my\_configuration\_file.conf

Emotif\_alpha -jid toy\_example -confFile my\_configuration\_file.conf

The html report is under “filtered motif” folder.

1. Post-analysis

All the scripts can be found at: <http://www.ohio.edu/people/yl079811/Motif_discovery_manual/>

1. Manually select the motifs

Open the html report. It should be similar to fig 1. You can look at the motif logo and the statistics. I often select the motifs with accuracy >70%, foreCov >50%, and backCov < 30%. But it depends.

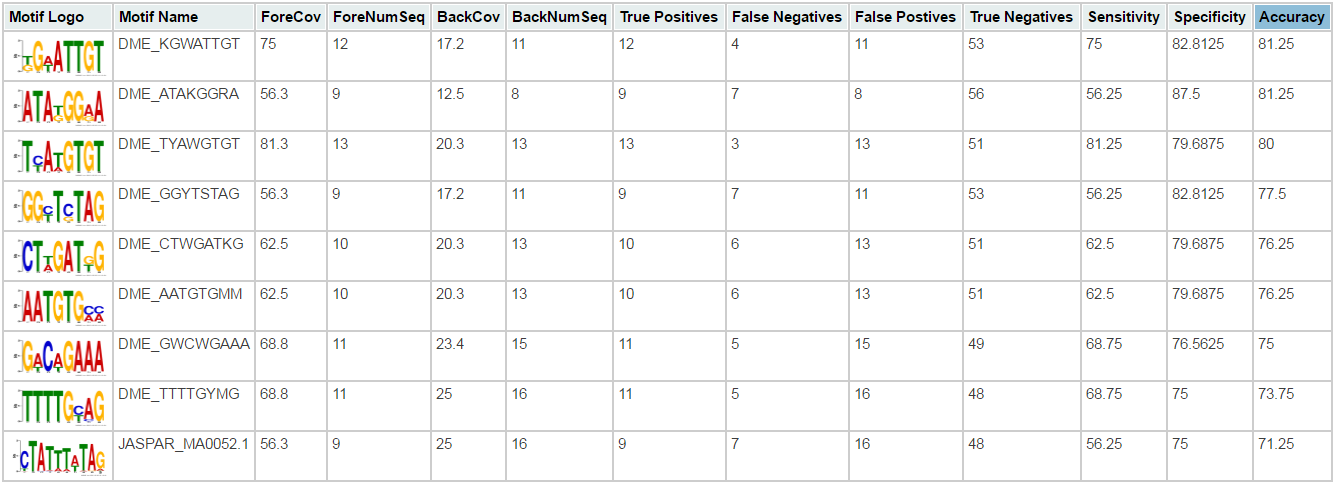


Figure . motif report

1. Make a new html report

After manually selecting the motifs, you may want to make an html file that only contains the motifs being selected. To do this, you can simply edit the original html file. In case of any wrong operations, you may want to make a copy of the original file. Now, use a text editor to open the html file and you can see something like :



That’s one row in the html file. So to make the html report, you just need to delete the texts.

1. Make a new PWM file.

You need to make a new PWM file containing only the motifs being selected. To do this, I have created a script subPWM.py (<http://www.ohio.edu/people/yl079811/Motif_discovery_manual/subPWM.py> ). It requires you give a list of selected motif names; this is not a parameter, it is hard coded. Then you can run it like:

Python subPWM.py output\_pwm\_name.pwm original\_pwm\_file

1. Violin plot of all the statistics for selected motifs

An R script is used to generate violin plot for all the statistics. It is available at: <http://www.ohio.edu/people/yl079811/Motif_discovery_manual/box_violin_scatter.py>.

The script should give you a figure similar to fig 2. The script takes a csv file as input. You can find the example csv file at: <http://www.ohio.edu/people/yl079811/Motif_discovery_manual/ovac_17_SE_SP_ACC_manually_filter.csv>

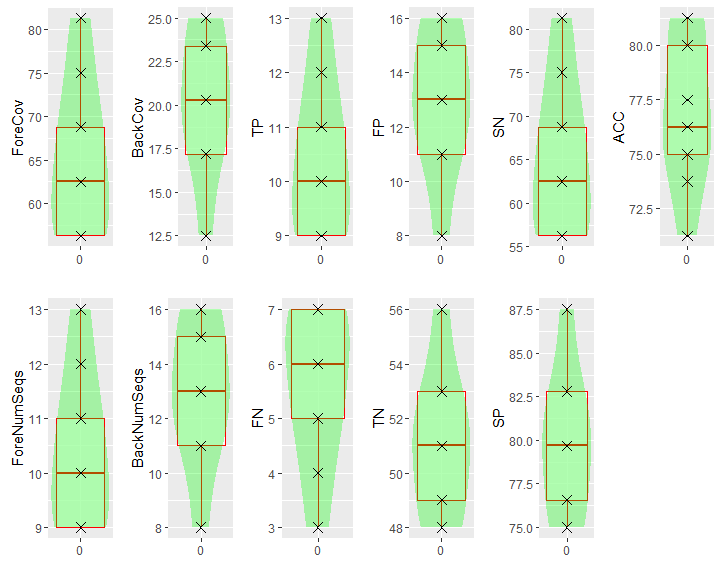
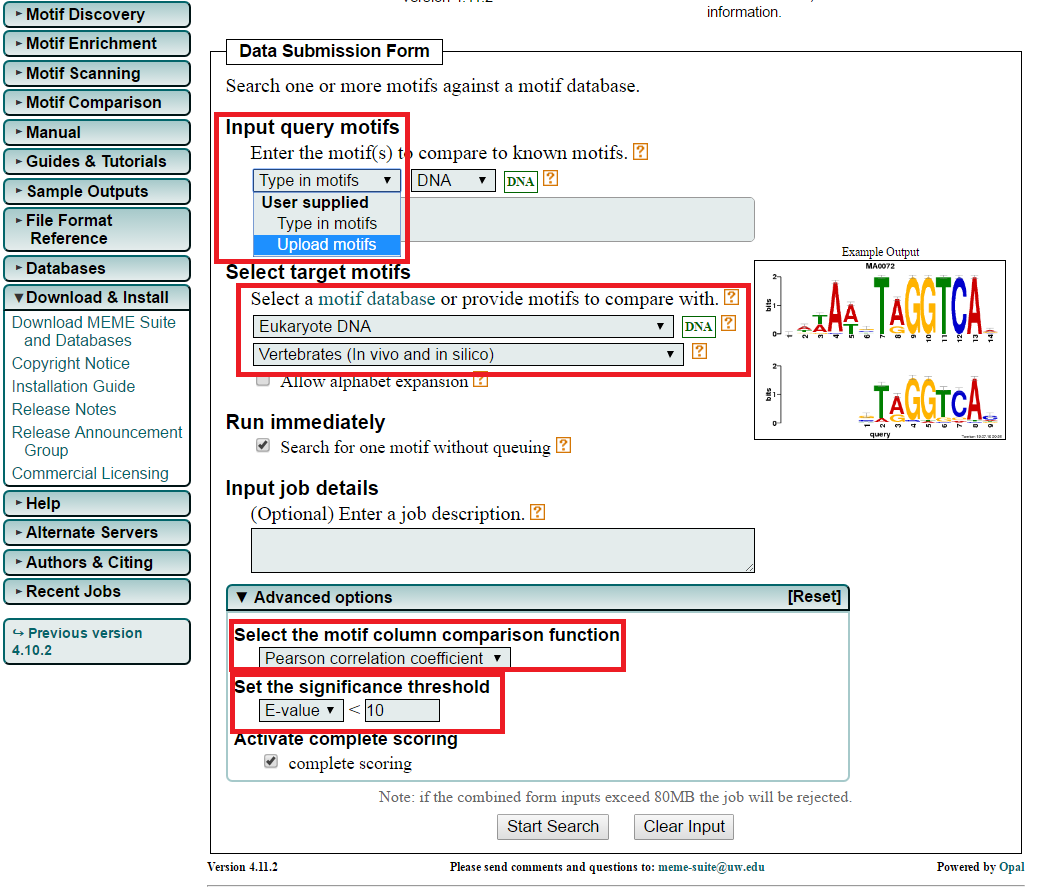


Figure . violin plot of the statistics

1. Match to known motifs

You can use Tomtom to match your motifs to motif databases. In the “Input query motifs”, choose “uploaded motifs”. Then select a motif database. In the advanced options, try different comparison functions and e-values. I often use Euclidian distance and e-value=1. Start your search.



1. Analysis of the genomic location

It is interesting to look at the distribution of the motifs, e.g. the distance from TSS. In this analysis, we are looking for some non-random distributed motifs.

An R script is available for generating the distribution violin plot (fig 3). It is at: <http://www.ohio.edu/people/yl079811/Motif_discovery_manual/TSS_box_violin_plot.py>

It takes the fimo output as input. You can use your selected motif pwm file and run fimo again at <http://meme-suite.org/tools/fimo> .

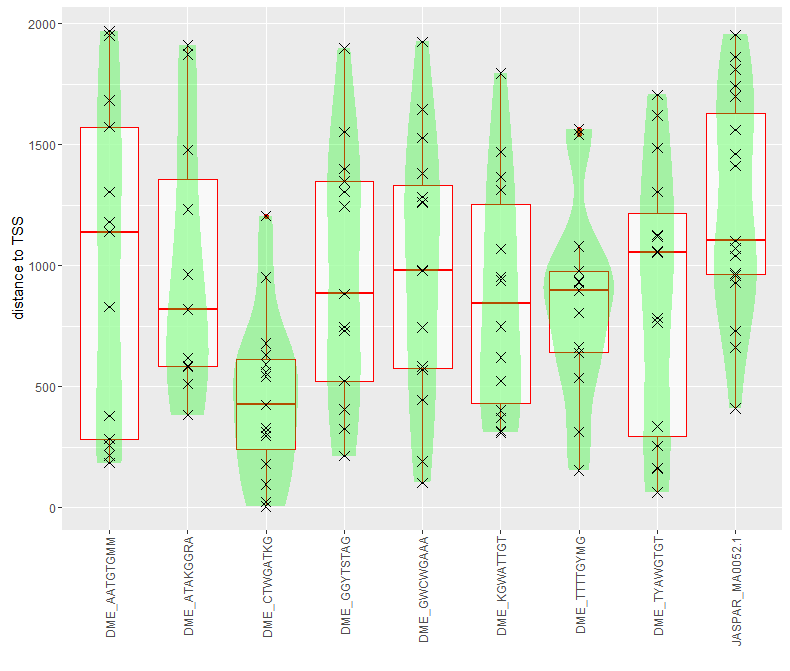


Figure . Violin plot of the distance to TSS

1. Sequence-motif visualization

Use MAST (<http://meme-suite.org/tools/mast>) to do sequence visualization. To report all occurrences, you need to set the threshold to be a large number, e.g. 99999.

