Application Tutorial of Blood Transcription Modules

In this tutorial, we demonstrate how Blood Transcription Modules (BTMs) can be used as an alternative to pathways, with several pathway analysis methods. The Part I of this tutorial will describe the use of BTMs as gene sets in enrichment analysis via the interface of GSEA program. The Part II will describe how to convert gene expression to module activity, to do enrichment test on a gene list and to do antibody correlation analysis via our supplied Python program. While Part I does not require special computational skills, Part II is intended for more advanced bioinformatics users. This tutorial is accompanied by a download package, which includes several files:

```
BTM_for_GSEA_20131008.gmt
monocytes_vs_bcells.txt
gene_ab_correlation.rnk
btm_tool.py
btm_example_data.py
MCV4_D3v0_probesets.txt
```

All data files are tab delimited, UNIX text files. Users can import them into spreadsheet programs for editing. Human gene identifiers are official gene symbols in uppercase.

Part I. Enrichment test using BTMs in GSEA program

The GSEA (Gene Set Enrichment Analysis) software can be freely downloaded from Broad Institute website (http://www.broadinstitute.org/gsea/). We will test for the enrichment of BTMs (as gene sets) in 2-class microarray data comparison (monocytes vs B cells), and using a pre-ranked method (antibody correlation).

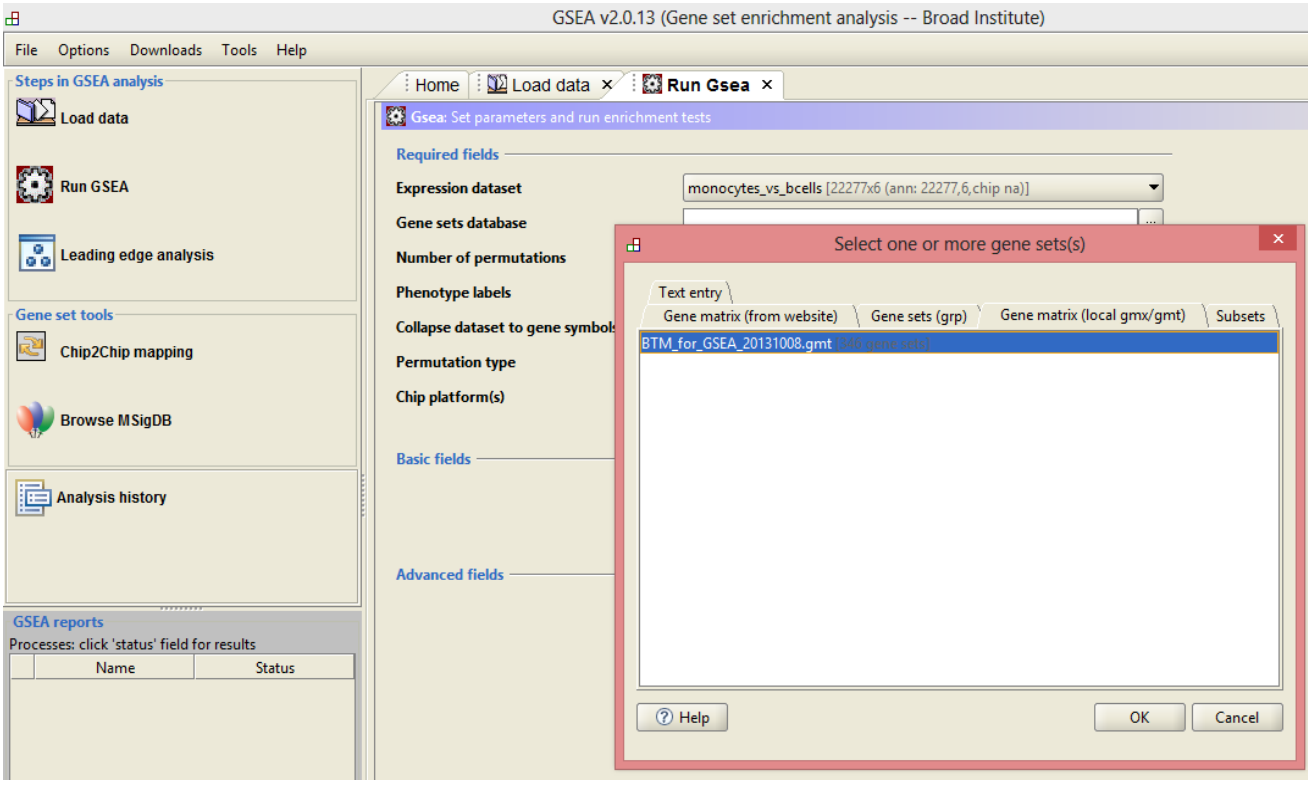

Figure T1. Using BTMs as locally supplied gene set to GSEA.

1. Launch GSEA interface, and click on "Load data". "Browse for files" to load `monocytes_vs_bcells.txt` and `BTM_for_GSEA_20131008.gmt`. The .txt file contains Affymetrix microarray data of 6 samples, and the .gmt file is the BTM modules in GSEA format.

2. Click "Run Gsea". To set the parameters in the "Run Gsea" interface, select "monocytes_vs_bcells" as Expression dataset. For "Gene sets database", click the button on the right to get the pop-up window as in **Figure T1**. Under the "Gene matrix (local gmx/gmt)" tab, find and select `BTM_for_GSEA_20131008.gmt`. This designates our BTM modules as the gene set for the analysis.

3. For "Phenotype labels", get to the pop-up window and fill in sample/class labels as in **Figure T2**.

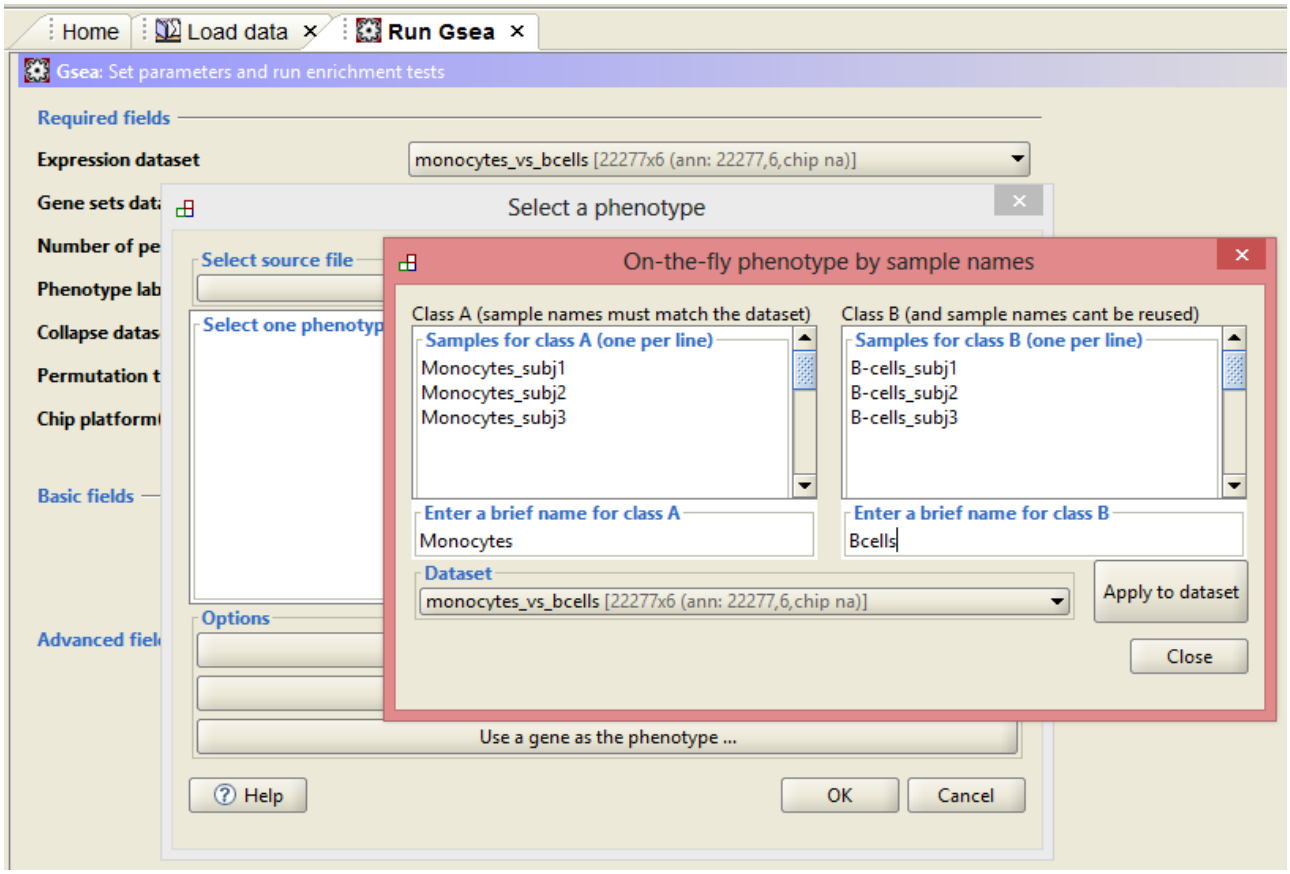

Figure T2. Phenotype labels.

4. Set "Permutation type" to "gene_set", "Chip platforms" to "HG_U133_Plus_2.chip". Because the microarray data in `monocytes_vs_bcells.txt` are supplied as probeset signals, the program will collapse the data to gene level.

5. You can define your own settings for the "Basic fields", except that the "Min size: excludes smaller sets" should be set to "10". The minimal size of a BTM module is 10 genes. **Figure T3** is the screen shot of parameters ready to run.

6. Click "Run". After it finishes, clicking the "Success" in the status field on the left will evoke the result in your web browser (which goes through the "index.html" file in the result directory). This completes a regular GSEA run, where the BTM modules were tested for their association to two sample groups (**Figure T4** shows an example output).

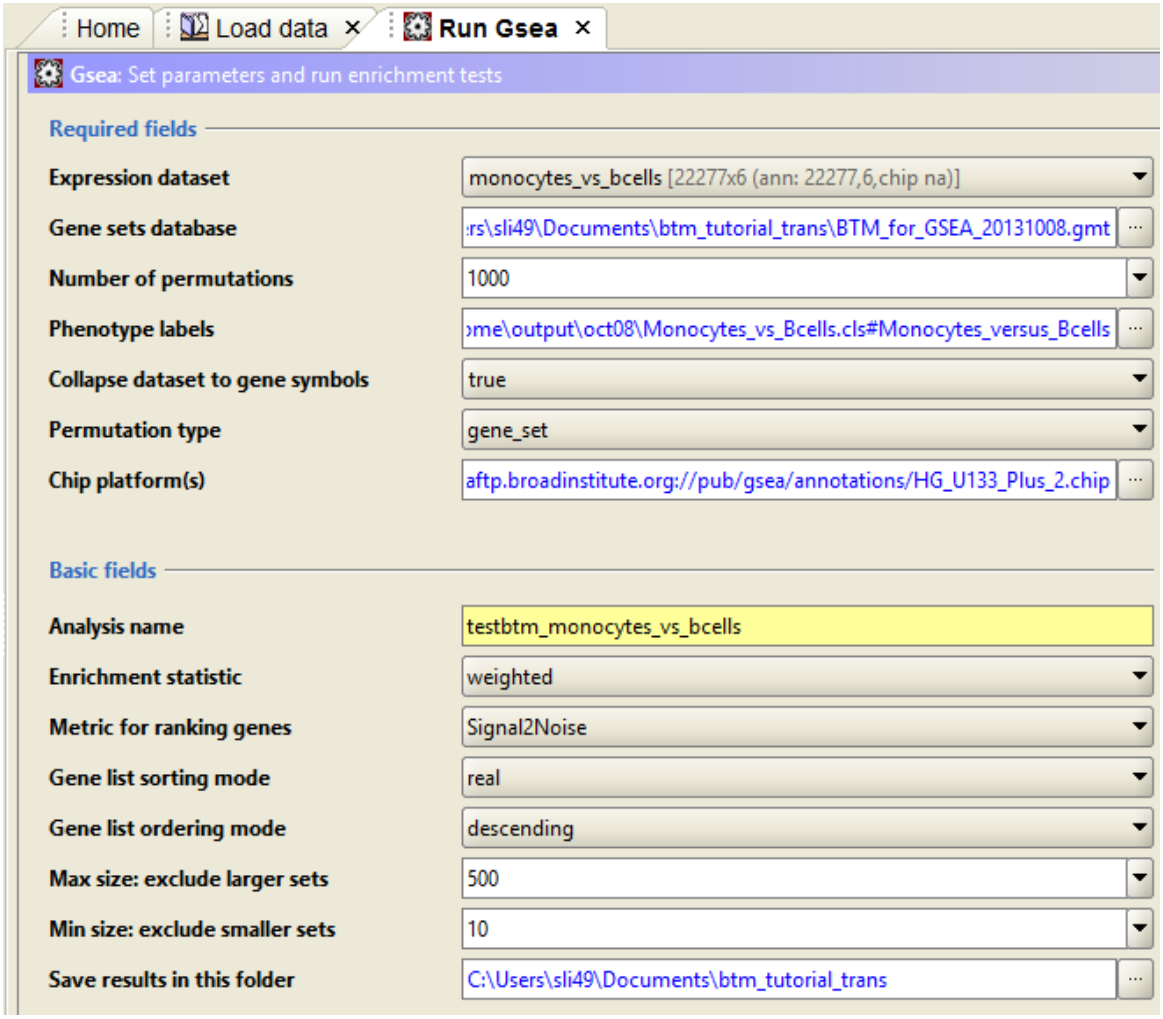

Figure T3. Screen shot of parameters, ready to run GSEA.

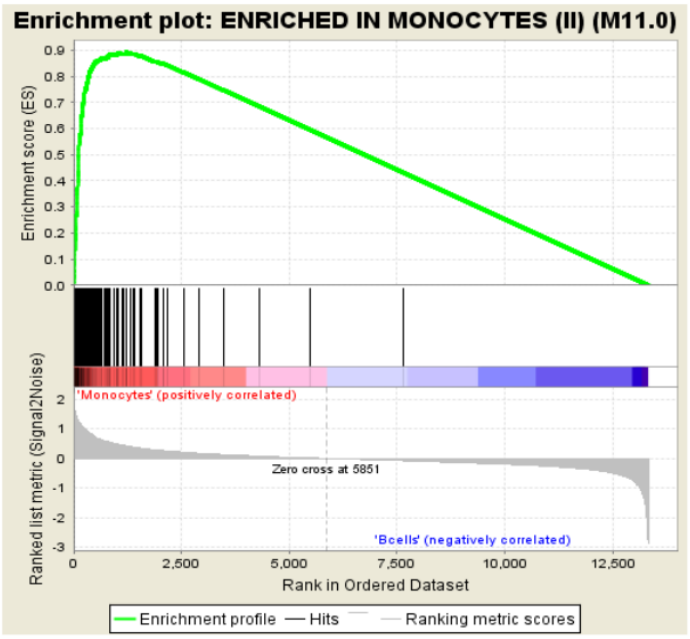

Figure T4. Example GSEA result. The red-blue bar represents the gene ranks by a designated statistical method (Signal2Noise in Figure T3). The vertical bars show the positions of member genes from our module M11.0. This module is highly enriched for the monocyte phenotype in the input microarray data.

7. The GSEA program also has an option to substitute its gene ranking method by your own. E.g. you can rank genes by paired t-test or Pearson correlation. The "GseaPreranked" method can be found under the "Tools" menu, as in **Figure T5**.

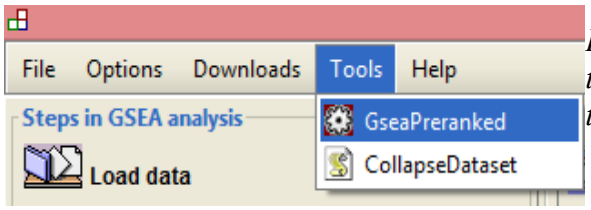

Figure T5. Pre-ranked method in GSEA to use the result from your own statistical test.

8. Load a new data file, `gene_ab_correlation.rnk`, similarly to Step 1. This is a gene list prepared separately, ranked by their Pearson correlation to antibody data.

9. In the "Run Gsea on a Pre-Ranked gene list" window, specify the parameters as in **Figure T6**. Now the "Ranked List" is our supplied data, which is already at gene level, without the need to collapse dataset.

10. Click "Run" to complete this analysis. This tested the enrichment of BTM modules based on the strength of their member genes correlated with antibody data, an alternative to the module activity method in Part II.

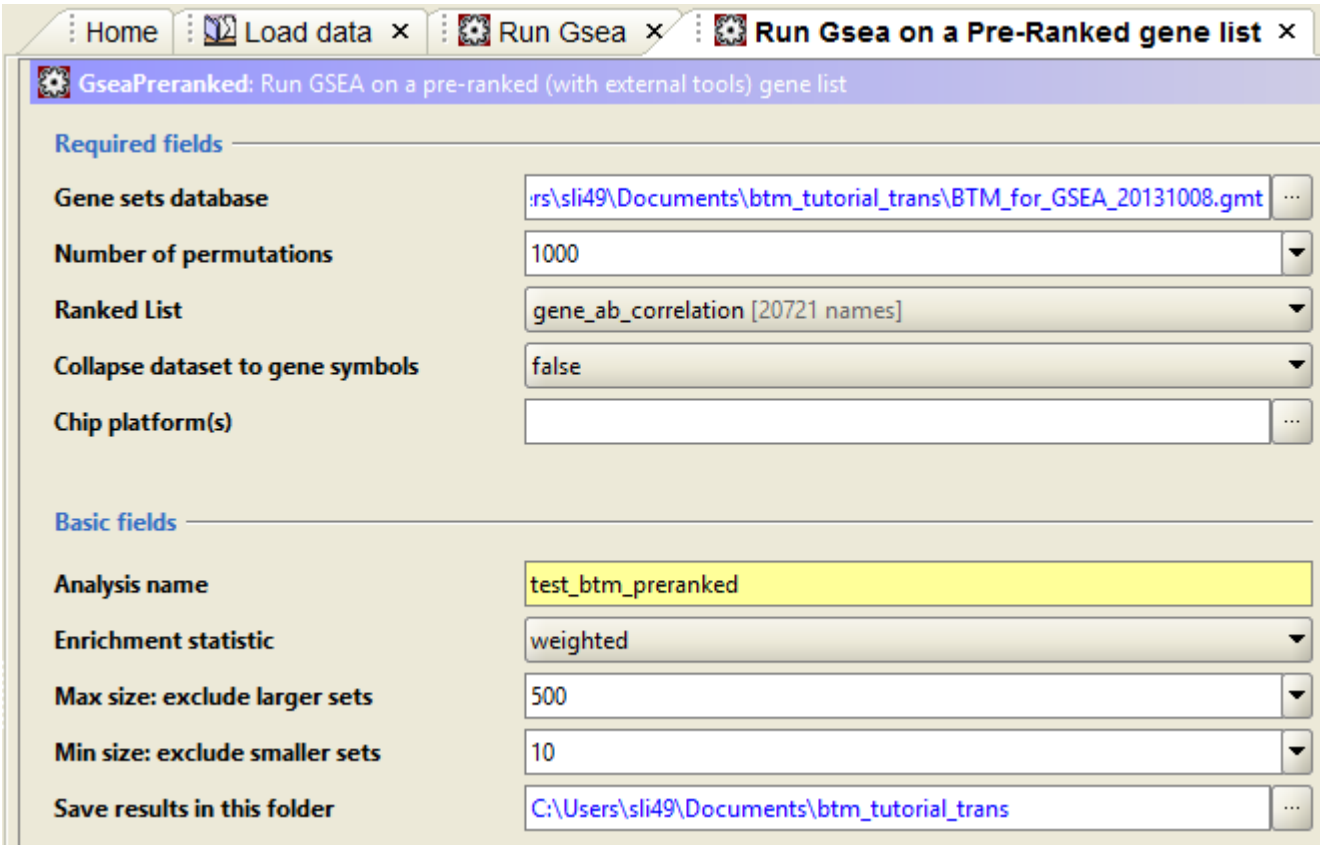

Figure T6. Screen shot of parameters, ready to run pre-ranked method in GSEA.

Part II. Using btm_tool program in Python command line

This part demonstrates the use of our supplied Python program, `btm_tool.py`, via command lines. This requires *Python 2.x* and its *Numpy/Scipy (ver 0.10+)* libraries. *Python 2.x* is shipped with Mac and Linux systems. Instruction of the library installation can be found at http://scipy.org/install.html. Windows users may follow the instruction there to do a bundled installation.

1. Evoke Python in the same directory where the downloaded files were unpacked, which should include `btm_tool.py` and others. In Python interpreter environment, do

```
>>> from btm_tool import *
```

This makes BTM data and a few functions available for your use.

2. The supplied file `MCV4_D3v0_probesets.txt` contains Affymetrix probeset level data from the MCV4 study. To convert it to gene level data, do

```
>>> probeset_to_genetable('MCV4_D3v0_probesets.txt',
affy_probeset_dict, 'MCV4_D3v0_genes.txt')
```

The output file is `MCV4_D3v0_genes.txt`. This program supports only Affymetrix platforms. For other platforms, you will need to prepare your own gene level data for the next step.

3. To **convert gene level data to BTM module activity scores**:

```
>>> genetable_to_activityscores('MCV4_D3v0_genes.txt',
'MCV4_D3v0_BTMactivity.txt')
```

The output file is `MCV4_D3v0_BTMactivity.txt`. The module activity scores are computed as the mean value of member genes. You can use these activity scores to perform further statistical test of your choice.

4. A common bioinformatics task is to test the over-representation in a list of genes. To do this with BTM modules, we first need to get a gene list of interest.

```
>>> genelist = [x.split('\t')[0] for x in
open('gene_ab_correlation.rnk').readlines()[20: 220]]
```

This is a trick to pull 200 genes from the `gene_ab_correlation.rnk` file we used earlier. You may want to get a more interesting genelist from your own data.

5. To **do enrichment test on this gene list using BTMs**:

```
>>> enrichment_test(genelist, 'my_enrichment_test.txt')
```

This performs Fisher Exact Test on this gene list and each BTM module. The output file, `my_enrichment_test.txt`, contains enrichment p-values of each module in tab-delimited format. You can import it into a spreadsheet program for further formatting and editing.

6. To **do antibody correlation analysis using BTM framework**:

```
>>> do_antibody_correlation('MCV4_D3v0_genes.txt',
mcv4_log2_antibody, 'my_correlation_test.txt')
```

This takes some time because the correlation significance is estimated by permutations of both sample labels and gene memberships. The input data are gene level expression file `MCV4_D3v0_genes.txt` and `mcv4_log2_antibody`, which is pre-loaded example antibody concentration. Of course, the antibody

data, in matched sample order, will have to be supplied by users in a real analysis. The output file, `my_correlation_test.txt`, contains the p-values of each module in tab-delimited format. You can import it into a spreadsheet program for further formatting and editing. If the plot library *matplotlib* is installed, a probability distribution figure will also be produced by this function.

This `btm_tool` program is provided as demonstration code. Users should feel free to modify and incorporate it into their own analysis.