Documentation

GeneSCF v1.1: Gene Set Clustering based on Functional annotation

Santhilal Subhash
Last revised 20 January 2016
Chapter 1

1. Overview of GeneSCF

Gene Set Clustering based on Functional annotation (GeneSCF) is a command line tool that uses database information from different sources: KEGG, REACTOME, Gene Ontology, and Network of Cancer Genes 4.0 (NCG) to find the enrichment of a set of user-provided gene list or target genes.

1.1 Gene Set Clustering based on Functional annotation

The statistical ranking used in this tool is a simple overlap significance Fisher's exact test employed with other multiple testing correction methods. Genes from the user-supplied list are clustered using the number of genes shared between the list of input genes and the genes in the biological functions from the database. The default database in GeneSCF contains functional information for Homo sapiens (Human) from KEGG, Reactome, Gene Ontology, and NCG, which can also be used for more species or organisms up to 4,000 using the prepare_database module in GeneSCF.

1.1.1 Gene Ontology

Gene ontology analysis is performed using the GO database, which has three sub-categories: Biological Process, Cellular Component, and Molecular Function. This tool can rank the biological functions from GO by integrating all three categories or individually based on users' interest (see section 1.3). GO supports more than 30 organisms.

1.1.2 Pathway analysis

Pathways are used from the KEGG database and can be ranked using users' set of genes and support more than 4,000 species. Reactome is limited to organisms, only supporting the database from Homo sapiens.

1.1.3 Cancer enrichment

For cancer enrichment analysis, the database containing 2,000 genes with 66 different cancer types are used from the Network of Cancer Genes 4.0 (NCG) database. NCG supports only genes from Homo sapiens.
Chapter 2

2. Installation and prerequisites

2.1 Installation
No Installation required. Please follow three simple steps below,
1. download the tool from http://genescf.kandurilab.org/downloads.php
2. Extract using TAR (tar -zxvf geneSCF-master-vx.x.tar.gz) and
3. Execute geneSCF.

2.2 System requirements

<table>
<thead>
<tr>
<th>Component</th>
<th>Minimum requirement/Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>512 MB (1 GB recommended)</td>
</tr>
<tr>
<td>System</td>
<td>Linux</td>
</tr>
<tr>
<td>PERL</td>
<td>&gt;=3.0 (Runs without any problem)</td>
</tr>
<tr>
<td>R-cran (Optional)</td>
<td>&gt;=3.0, Only if needed addition plots</td>
</tr>
<tr>
<td>ggplot2 R-package</td>
<td>Only if needed addition plots</td>
</tr>
</tbody>
</table>

2.3 Test basic UNIX commands
Before starting the analysis, please test these basic UNIX supportive commands for GeneSCF on the terminal.

**Commands:** awk, cat, gzip, wget, rm, mkdir, sort, date, sed, paste, join, grep, curl, echo, unzip, tar.
# Chapter 3

## 3. Detailed GeneSCF usage

### 3.1 General usage

```
```

<table>
<thead>
<tr>
<th>Arguments / Parameters</th>
<th>Options / Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ -m=</td>
<td>--mode= ]</td>
</tr>
<tr>
<td>[ -i=</td>
<td>--infile= ]</td>
</tr>
<tr>
<td>[ -t=</td>
<td>--gtype= ]</td>
</tr>
<tr>
<td>[ -db=</td>
<td>--database= ]</td>
</tr>
<tr>
<td>[ -o=</td>
<td>--outpath= ]</td>
</tr>
<tr>
<td>[ -bg=</td>
<td>--background= ]</td>
</tr>
<tr>
<td>[ -org=</td>
<td>--organism= ]</td>
</tr>
</tbody>
</table>
Arguments / Parameters | Options / Description
--- | ---
\[-db= | --database=\] | Options: [GO_all|GO_BP|GO_MF|GO_CC|KEGG|REACTOME|NCG]
\[-org= | --organism=\] | Options: [see, org_codes_help]

3.2 Other modules

./prepare_database -db=[GO_all|GO_BP|GO_MF|GO_CC|KEGG|REACTOME|NCG]
-org=[see, org_codes_help]

Arguments / Parameters | Options / Description
--- | ---
\[-db= | --database=\] | Options: [GO_all|GO_BP|GO_MF|GO_CC|KEGG|REACTOME]
\[-org= | --organism=\] | Options: [see, org_codes_help]

3.3 Organisms and codes

Multi-organism supported databases

<table>
<thead>
<tr>
<th>organism codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEGG</td>
</tr>
<tr>
<td>GeneSCF-master-vx-x/organ_codes_help OR <a href="http://rest.kegg.jp/list/organism">http://rest.kegg.jp/list/organism</a> (Column 2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>organism codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>GO (GO_BP,GO_CC,GO_MF,GO_all)</td>
</tr>
<tr>
<td>GeneSCF-master-vx-x/organ_codes_help OR <a href="http://www.geneontology.org/gene-associations/go_annotation_metadata.all.json">http://www.geneontology.org/gene-associations/go_annotation_metadata.all.json</a> (“id:”)</td>
</tr>
</tbody>
</table>

Single-organism supported databases

<table>
<thead>
<tr>
<th>organism code (Homo sapiens or Human)</th>
</tr>
</thead>
<tbody>
<tr>
<td>REACTOME or NCG</td>
</tr>
<tr>
<td>Hs</td>
</tr>
</tbody>
</table>
Chapter 4

4. Running test datasets (Tutorial)

GeneSCF comes with test datasets from two studies conducted in organism Human. One with list differentially expressed genes between healthy individuals and Chronic Lymphocytic Leukemia (CLL) patients. The second study with two gene lists containing p53 bound genes from 0 hours and 12 hours respectively (For detailed information about datasets refer GeneSCF article).

In this tutorial the task is to implement all type of databases from GeneSCF on these three different gene lists.

4.1 p53 bound genes

4.1.1 Gene Ontology run

>>>./geneSCF -m=update -i=test/H0.list -o=test/output/ -t=sym -db=GO_MF -bg=20000 --plot=yes -org=goa_human

The above run will search the gene list against Gene Ontology database (-db) and its sub-category Molecular Function (GO_MF) to find the enrichment. In this example, number of background genes (-bg) is 20,000 and organism (-org) Human goa_human (more organism codes refer 'org_codes_help' folder in GeneSCF). Selection of --mode parameter will determines whether GeneSCF should do enrichment run using stored database for Human on GeneSCF (-m=normal) or to use real-time database by directly connecting to GO_MF (-m=update). If 'update' mode used for already stored database on GeneSCF, the old GO database for human will be updated automatically to the recent information from GO. Since the list of input genes are from HUGO gene symbols the parameter for --type or -t is represented as 'sym'.

>>>./geneSCF -m=normal -i=test/H12.list -o=test/output/ -t=sym -db=GO_MF -bg=20000 --plot=yes -org=goa_human
This command is similar to the previous run except that we are using different list of genes and also importantly we are running in 'normal' mode. The reason for using 'normal' mode on second run is because once the 'update' mode is used for particular organism and database will be stored in GeneSCF. In this case the GO database for Human would have updated on first run and there is no need to use 'update' mode on consecutive runs for same organism and same database to reduce GeneSCF analysis time.

4.1.2 KEGG run
Similarly use the same dataset to see enriched pathways from KEGG for Human (hsa). For more organism codes refer 'org_codes_help' folder in GeneSCF.

```bash
>>>./geneSCF -m=update -i=test/H0.list -o=test/output/ -t=sym -db=KEGG -bg=20000 --plot=yes -org=hsa
>>>./geneSCF -m=normal -i=test/H12.list -o=test/output/ -t=sym -db=KEGG -bg=20000 --plot=yes -org=hsa
```

4.2 CLL differentially expressed genes (DEGs)

4.2.1 NCG run (cancer enrichment)

```bash
>>>./geneSCF -m=normal -i=test/TumorNormal_fc2.list -o=test/output/ -t=sym -db=NCG -bg=20000 --plot=yes -org=Hs
```

Above run searched the CLL DEGs against Network of Cancer Genes (NCG) database to find whether the obtained genes has any enrichment in different cancer types. NCG does not support 'update' mode.

4.2.2 KEGG run

```bash
>>>./geneSCF -m=normal -i=test/TumorNormal_fc2.list -o=test/output/ -t=sym -db=KEGG -bg=20000 --plot=yes -org=hsa
```
Above run searches the CLL DEGs against KEGG pathway database to find molecular pathways which are affected in CLL. The 'normal' mode is used because of the previous run from first dataset (section 3.1.2).

4.3 Implementing prepare_database module

This module prepares the required database prior to enrichment run. For example, the enrichment analysis in GeneSCF can be done in two different ways.

4.3.1 Method 1 (prior database preparation)

Modified example from section 3.1.1,

```bash
>>> ./prepare_database -db=GO_MF -org=goa_human
>>>./geneSCF -m=normal -i=test/H0.list -o=test/output/ -t=sym -db=GO_MF -bg=20000 --plot=yes -org=goa_human
```

In this method user will first prepare the required database (GO_MF) for an organism (goa_human) and then run the enrichment in 'normal' mode. This method will be useful for storing multiple databases for multiple organisms locally on GeneSCF for future enrichment analysis. And this can reduce time by using 'normal' mode on consecutive batch jobs on multiple databases.

4.3.2 Method 2 (Use real-time database)

Example from section 3.1.1,

```bash
>>>./geneSCF -m=update -i=test/H0.list -o=test/output/ -t=sym -db=GO_MF -bg=20000 --plot=yes -org=goa_human
```

In this method the GeneSCF uses in-built 'prepare_database' module to connect remote repository and utilize updated database for enrichment analysis (when specified -m='update'). This is one step run for Method 1, this is useful when user want to use updated database on each run.
Chapter 5

5. Interpreting GeneSCF output

The output from GeneSCF contains simple tab-separated (.tsv) file containing enriched functions and corresponding group of genes with some statistical tests. The file contains 12 columns and it is explained in detail below.

5.1 Description of output

<table>
<thead>
<tr>
<th># Column</th>
<th>Column names</th>
<th>Column description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Column 1</td>
<td>Genes</td>
<td>Matched user genes list for corresponding function</td>
</tr>
<tr>
<td>Column 2</td>
<td>Process~name</td>
<td>Name of matching function / Cancer type</td>
</tr>
<tr>
<td>Column 3</td>
<td>num_of_genes</td>
<td>Number of hits in the functional database from user gene list</td>
</tr>
<tr>
<td>Column 4</td>
<td>gene_group</td>
<td>Total number of genes involved in corresponding functions</td>
</tr>
<tr>
<td>Column 5</td>
<td>Percentage%</td>
<td>Percentage of functional genes covered by user gene list</td>
</tr>
<tr>
<td>Column 6</td>
<td>P-value</td>
<td>Probability of enrichment using Fisher's exact test</td>
</tr>
<tr>
<td>Column 7-12</td>
<td>False Discovery Rate method</td>
<td>Multiple testing correction methods</td>
</tr>
</tbody>
</table>

5.2 Visualizing the results

When specified –plot parameter, GeneSCF plots top 20 enriched functions ranked using log-transformed P-value. This requires installation of R and 'ggplot2' package (See section 2.2). The bubble plot below shows the enrichment of CLL DEGs (section 3.2.1) in Leukemia and Non-hodgkin lymphoma with p-value < 0.05 significance level. The size of bubble represents the percentage of functional genes (in this case Cancer genes) covered (Column 5 from the TSV file).
5.3 Statistical methodology

GeneSCF uses commonly used overlap statistics of Fisher's exact test and for multiple testing corrections it uses different methods. Please refer GeneSCF article for detailed methodology.
Bibliography


