## Contents

1 Preface .......................................................................................................................... 1

2 GLANET Presentation on YouTube
   2.1 Content: .................................................................................................................. 3

3 Indices and tables .......................................................................................................... 31
Preface

GLANET is Genomic Loci ANnotation and Enrichment Tool.
2.1 Content:

2.1.1 GLANET Overview

GLANET

GLANET is Genomic Loci ANnotation and Enrichment Tool.

Genomic studies identify genomic loci of interest representing genetic variations, transcription factor binding, or histone modification through next generation sequencing (NGS) technologies. Interpreting these loci requires evaluating them within the context of known genomic and epigenomic annotations. We present GLANET as a comprehensive annotation and enrichment analysis tool. Input query to GLANET is a set of genomic intervals of any length. GLANET annotates and performs enrichment analysis on these loci with a rich library of genomic elements. GLANET annotation library includes intervals of genomic elements such as:

- regions defined on and in the neighbourhood of coding regions
- ENCODE-derived potential regulatory regions that encompass binding sites for multiple transcription factors, DNaseI hypersensitive sites, modification regions for multiple histones across a wide variety of cell types
- gene sets derived from KEGG pathways

In order to identify the genomic elements enriched in the query set, GLANET implements a sampling-based enrichment test that accounts for systematic biases such as GC content and mappability inherent to NGS technologies. When the input intervals are derived from an NGS experiment, these biases constrain regions of the genome that can contribute to interval generation. Few of the existing tools account for one or both of these biases at coarse-grain level but not at fine-grain level as GLANET does.

We developed GLANET both as an annotation and enrichment tool with several useful built-in analysis capabilities. Users can easily annotate their input intervals with the genomic elements defined in the annotation library and expand the GLANET library by adding user-defined libraries and/or pre-defined gene sets. When the input is a SNP list, users
can evaluate whether the SNPs reside in transcription factor binding regions and, if so, whether they are located in the actual transcription factor binding motifs obtainable via either the reference or the SNP allele and whether the variation potentially impacts the binding of TFs, either by enhancing or disrupting binding motifs. GLANET enables joint enrichment analysis for transcription factor binding and KEGG pathways. With this option, users can evaluate whether the input set is enriched concurrently with binding sites of TFs and the genes within a KEGG pathway. This joint enrichment analysis provides a detailed functional interpretation of the input loci.

In order to assess the statistical power and Type-I error control of GLANET, we designed data-driven computational experiments using large collections of ENCODE ChIP-seq and RNA-seq data. These experiments indicated that while GLANET enrichment test often performs conservatively in terms of Type-I error, it has high statistical power.

GLANET can be run using its GUI or on command line.

GLANET Flowchart

GLANET Features

- Users can query SNPs or varying length genomic intervals for annotation and/or enrichment analysis. GLANET supports commonly used input formats such as BED, GFF3, 0-based or 1-based coordinates, and reference SNP (RS) IDs for SNPs.
- GLANET implements a null model that accounts for systematic biases such as mappability and GC content inherent to NGS in order to evaluate whether the input intervals overlap significantly with the genomic elements in the GLANET annotation library.
- GLANET’s sampling-based enrichment analysis also accounts for given interval’s length and the chromosome it is located on in addition to the systematic biases such as mappability and GC content.
- GLANET interprets gene sets in three different ways which are exon-based, regulation-based and all-based manner. Exon-based gene set takes exons of genes, regulation-based takes introns, upstream and downstream proximal regions of genes and lastly all-based takes all the defined regions in exon-based and regulation-based, plus upstream and downstream distal regions of genes of each gene set into account.
• GLANET enables user to load user defined gene sets and/or user defined library and to expand annotation library.

• GLANET provides Regulatory Sequence Analysis using RSAT's matrix scan web service for all of the annotated TFs when the input consists of SNPs only.

• GLANET has assessed its Type-I error and power by designed data-driven computational experiments on two cell lines, GM12878 and K562, which showed that it has a well-controlled Type-I error rate and high statistical power.

GLANET Output

After a successful GLANET execution

• Annotation results can be found under
  ~path/to/GLANET Folder/Output/givenJobName/Annotation/

• Enrichment results can be found under
  ~path/to/GLANET Folder/Output/givenJobName/Enrichment/

• Regulatory Sequence Analysis results can be found under
  ~path/to/GLANET Folder/Output/givenJobName/RegulatorySequenceAnalysis/

• GLANET log file (GLANET.log) can be found under the same directory where GLANET.jar is located. If you are running GLANET from the source code, GLANET log file will be created under Glanet project directory.

2.1.2 GLANET Installation and System Requirements

GLANET Installation

You can find a demo screencast for GLANET Installation on youtube in the link below: https://www.youtube.com/watch?v=d-tRriMCmjY&t=130s

1. Java

   Install latest Java SE from here

2. Perl

   Important: If you have installed Perl before but it is not a version of Strawberry Perl, we strongly suggest that you, first, uninstall the Perl you have and then follow the steps specified below. This is very important because required modules have to be downloaded from Strawberry Perl’s database.

   For Windows users, Strawberry Perl can be downloaded from here

   For Mac OS X users, the operating system comes with the installed Perl. If you want to update or install Perl, open Terminal.app and write the command below:

   ```bash
   $ sudo curl -L http://xrl.us/installperlosx | bash
   ```

   For Linux users, Perl is probably installed in your operating system. If you want to update or install Perl, open a Terminal and write the command below. After installing perl, you may also need to install parser library for Perl. You may run the commands below seperately:

   ```bash
   $ sudo curl -L http://xrl.us/installperlnix | bash
   $ sudo apt-get build-dep libxml-parser-perl
   ```
After you have installed Perl, you need to install the required modules.

First install cpanminus, which will allow other modules to be installed easily. Open **Terminal (or Command Prompt in Windows)** and write the command below:

```
$ cpan App:cpanminus
```

Now, install Getopt/Long.pm module. Note that if any of the modules below is already installed in your computer, you will be notified:

```
$ cpanm Getopt::Long
```

**Important:** If one of your modules is not installed successfully, then you may run the command with sudo, if you have Linux/Mac OS X operating system. For Windows, you may want to run command prompt as administrator instead of running the command with sudo. For Linux and Mac OS X operating systems, you may install a module with sudo as following:

```
$ sudo cpanm Getopt::Long
```

**Important:** If you are still having issues for installing a module, try to add them using -f option, which will take longer but it will try to force the module to be installed. Note that if this step still does not work, we suggest you to try installing the modules another time. There might be some problems with the server at that time. For example, if Getopt/Long.pm is still not installed, you may want to write:

```
$ cpan -f Getopt::Long
```

Install LWP/UserAgent.pm module:

```
$ cpanm LWP::UserAgent
```

Install HTTP/Request/Common.pm module:

```
$ cpanm HTTP::Request::Common
```

Install HTTP/Headers.pm module:

```
$ cpanm HTTP::Headers
```

Install XML/XPath.pm module:

```
$ cpanm XML::XPath
```

Install XML/XPath/XMLParser.pm module:

```
$ cpanm XML::XPath::XMLParser
```

Install JSON module:

```
$ cpanm JSON
```

3. Download executable GLANET.jar from executable-label

4. Download Data.zip from data-label and extract it as **Data** under a directory you name it, which will become your **GLANET Folder**.

    e.g.: ~path/to/GLANET Folder/

Data.zip contains the all the necessary data for Annotation. The important point is that this **GLANET Folder** directory must be the parent directory of extracted **Data** directory.
 Once you have followed all these steps, you should be ready to run GLANET properly.

**GLANET System Requirements**

1. You can download and run GLANET in any operating system (Windows, Mac OS X, Linux).
2. Your computer should have at least 8 GB memory. Otherwise, you may not be able to use all the functionalities of GLANET.
3. Java SE 8 (or higher) should be installed in your computer in order to execute GLANET. We suggest you to use the latest Java SE update.
4. Strawberry Perl should be installed in your computer.
5. During execution, GLANET makes calls to NCBI E-utilities and RSAT web service, therefore GLANET must be run an a computer with an internet connection.

### 2.1.3 Download

**Source Code**

GLANET is an open source code project. You can browse and download GLANET source code from [github](https://github.com) and run it as a Java project.

**Executable**

You can download GLANET executable jar from [here](https://github.com) and run it.

**Data**

If you want to use all features of GLANET, please download Data.zip from the link below.

<table>
<thead>
<tr>
<th>Explanation</th>
<th>Data Files(zip)</th>
<th>Data Files(tar.gz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download GLANET data</td>
<td>Data.zip</td>
<td>Data.tar.gz</td>
</tr>
<tr>
<td>Download GLANET data without GC and Mappability matching</td>
<td>Data_woGCM.zip</td>
<td>Data_woGCM.tar.gz</td>
</tr>
</tbody>
</table>

### 2.1.4 GLANET Tutorial

You can run GLANET using its graphical user interface (GUI) or command line arguments.

By double clicking the GLANET.jar you can open GLANET’s GUI. In this case, GUI will be opened with default amount of memory allocated for JVM which is 256MB. In order to allocate specific amount of memory for GLANET.jar and run it through its GUI, one should write the following basic command on Terminal (Linux or Mac OS X) or on Command Prompt (Windows)*:

```bash
$ java -Xms8G -Xmx8G jar -path/to/GLANET.jar
```

Note that with this command above, 8GM of memory is allocated for GLANET.jar. Depending on the number of intervals in the sets and number of samplings you want to achieve you may need to increase the memory accordingly.
In order to run GLANET.jar through command line arguments, in addition to the memory allocation parameter above, you must provide other arguments which are explained below.

Throughout the guide, we will use ~path/to/GLANET.jar to indicate your absolute path to GLANET.jar and ~path/to/GLANET Folder/ to indicate your absolute path to GLANET Folder.

GLANET Graphical User Interface

Fig. 2.1: GLANET GUI Upper Part

GLANET provides Annotation by enabling user to annotate given genomic intervals w.r.t. ENCODE DNA regulatory elements, RefSeq genes, predefined gene sets such as KEGG Pathways, user defined gene sets and user defined library. Other features of GLANET includes Enrichment Analysis and Regulatory Sequence Analysis.

1. **Input File Name**: (Mandatory)
You have to provide input file which contains user given genomic intervals. Assume that Data.zip is extracted under a directory, let’s call it GLANET Folder. Then sample input data can be reached from

~path/to/GLANET Folder/Data/demo_input_data/

2. Input Format: (Mandatory)

GLANET supports input formats such as dbSNP IDs, BED, GFF3, 1-based coordinates (End Inclusive) and 0-based coordinates (End Inclusive). For the user given input file, Input Format has to be selected accordingly.

• dbSNP IDs
  – Sample input data for dbSNP IDs can be reached at

~path/to/GLANET Folder/Data/demo_input_data/CVD_rsIDs_Mediation.txt
  – Input Format must be selected as dbSNP IDs.

• BED
  – Sample input data for BED can be reached at

~path/to/GLANET Folder/Data/demo_input_data/CVD_Mediation_0BasedStart_EndExclusive_GRCh37_p13_coordinates.bed
  – Input Format must be selected as BED.

• GFF3
  – Sample input data for GFF3 can be reached at

~path/to/GLANET Folder/Data/demo_input_data/test_input_data_GFF3_format.gff3
  – Input Format must be selected as GFF3.

• 1-based coordinates (End Inclusive)
  – Sample input data for 1-based coordinates (End Inclusive) can be reached at

~path/to/GLANET Folder/Data/demo_input_data/OCD_GWAS_chrNumber_1Based_GRCh37_p13_Coordinates.txt
  – Input Format must be selected as 1-based coordinates (End Inclusive).

• 0-based coordinates (End Inclusive)
  – Sample input data for 0-based coordinates (End Inclusive) can be reached at

~path/to/GLANET Folder/Data/demo_input_data/test_interval_data_K562_Usf2_0based_GRCh37_p13_Coordinates.txt
– Input Format must be selected as 0-based coordinates (End Inclusive).

3. **Assembly**: (Mandatory for BED, GFF3, 1-based coordinates (End Inclusive) or 0-based coordinates (End Inclusive))

   GLANET supports two assemblies.
   - **GRCh38**
     - GRCh38 is also known as hg38
   - **GRCH37.p13 (hg19)**
     - GRCh37.p13 is also known as hg19

   If BED, GFF3, 1-based coordinates (End Inclusive) or 0-based coordinates (End Inclusive) is chosen as Input Format, then Assembly has to be set as either GRCh38 or GRCH37.p13.

   In case of dbSNP IDs, there is no need to set Assembly.

4. **GLANET Folder**: (Mandatory)

   Set the GLANET Folder (e.g.: ~path/to/GLANET Folder). Please note that GLANET Folder has to be the parent of Data Folder.

   ~path/to/GLANET Folder/Data/

5. **Annotation, Overlap Definition, Number of Bases**: (Mandatory)

   For Annotation, set the number of bases for overlap definition. e.g: Setting number of bases as 3 means that two intervals are accepted as overlapped if and only if these intervals have at least 3 overlapping bases. Default is 1 in order to handle the case where the SNPs are given as input data.

6. **Annotation, Annotation Options**: (At least one of the Annotation Option has to be checked)
   - **DNase Annotation (CellLine Based)**
     - Check this check box, if you want to annotate given intervals w.r.t. ENCODE provided DNaseI hypersensitive sites.
   - **Histone Annotation (CellLine Based)**
     - Check this check box, if you want to annotate given intervals w.r.t. ENCODE provided Histone modifications sites.
   - **Transcription Factor (TF) Annotation (CellLine Based)**
     - Check this check box, if you want to annotate given intervals w.r.t. ENCODE provided Transcription Factors (TFs) binding sites.
   - **Gene Annotation**
     - Check this check box, if you want to annotate given intervals w.r.t. clade: Mammal, genome: Human, assembly: Feb. 2009 (GRCh37/hg19), RefSeq Genes.
   - **Gene Ontology Annotation**
     - Check Biological Process GO check box, if you want to annotate given intervals w.r.t. Gene Ontology Biological Process terms in exon-based, regulation-based and all-based manner.
     - Check Molecular Function GO check box, if you want to annotate given intervals w.r.t. Gene Ontology Molecular Function terms in exon-based, regulation-based and all-based manner.
– Check Cellular Component GO check box, if you want to annotate given intervals w.r.t. Gene Ontology Cellular Component terms in exon-based, regulation-based and all-based manner.

• **KEGG Pathway Annotation**

  – Check this check box, if you want to annotate given intervals w.r.t. KEGG Pathways in exon-based, regulation-based and all-based manner.

• **TF and KEGG Pathway Annotation**

  – Check this check box, if you want to annotate given intervals concurrently w.r.t. Transcription Factors binding sites and KEGG Pathways in exon-based, regulation-based and all-based manner.

  This is a joint annotation which means that given interval, TF and KEGG Pathway intervals overlap concurrently. Here, same TF with different cell lines are pooled.

  In other words, joint annotation requires trio overlaps.

• **TF and KEGG Pathway Annotation (CellLine Based)**

  – Check this check box, if you want to annotate given intervals concurrently w.r.t. Transcription Factors binding sites (CellLine Based) and KEGG Pathways in exon-based, regulation-based and all-based manner.

  This is a joint annotation which means that given interval, TF and KEGG Pathway intervals overlap concurrently.

  In other words, joint annotation requires trio overlaps.

• **User Defined Gene Set Annotation**

  – Check this check box, if you want to annotate given intervals w.r.t. User Defined Gene Set in exon-based, regulation-based and all-based manner.

  – User Defined Gene Set Input File (Mandatory if User Defined Gene Set Annotation check box is checked.)

  – User Defined Gene Set Input File lists each gene in a gene set in one line, in row-based manner. Namely, there will be n lines/rows for a gene set that consists of n genes.

    * User Defined Gene Set Input File contains tab delimited GeneSetID and Gene Information per line.

    e.g.: GO:0031424 LCE6A

    * Sample User Defined Gene Set Input File can be reached at

    ~path/to/GLANET Folder/Data/demo_input_data/UserDefinedGeneSet/GO/GOTerm_GeneSymbol_EvidenceCode_Ontology.txt

  – **Gene Information Type (Mandatory if User Defined Gene Set Annotation check box is checked.)**

    * GLANET supports three Gene Information Types:

    * GENE ID (e.g.: 84561)

    * GENE SYMBOL (e.g.: SLC12A8)

    * RNA NUCLEOTIDE ACCESSION (e.g.: NM_024628)
* Choose the appropriate Gene Information Type. (e.g: Choose GENE SYMBOL if you have loaded GOTerm_GeneSymbol_EvidenceCode_Ontology.txt as User Defined Gene Set Input File.)

- **Name (Optional)**
  - You can give a name for the User Defined Gene Set
    - e.g.: GO
  - Default Name is NoName.

- **Description File (Optional)**
  - Description File contains tab delimited GeneSetID and description of Gene Set per line.
    - e.g.: GO:0000001 mitochondrion inheritance
  - Sample Description File can be reached at
    ~path/to/GLANET Folder/Data/demo_input_data/UserDefinedGeneSet/GO/GO_ids2terms.txt

* **User Defined Library Annotation**
  - Check this check box, if you want to annotate given intervals w.r.t. User Defined Library.
  - **User Defined Library Input File (Mandatory if User Defined Library Annotation check box is checked.)**
    - In this input file, you list the file/s that you want to add into library.
    - User Defined Library Input File contains tab delimited, 4 columns, ~path/to/file, ElementType, ElementName and optional window-size value for considering window around summit in case of TF Data per line:
      
      ```
      G:\GLANET_DATA\ENCODE\transcription_factors\spp.optimal.
      → wgEncodeBroadHistoneGm12878CtcfStdAlnRep0_VS_
      → wgEncodeBroadHistoneGm12878ControlStdAlnRep0.narrowPeak TF →
      → CTCF_GM12878
      ```
    - With an header line at the top, in each row of this input file, there are 4 columns separated by tab.
    - **Header Line describes the 4 columns in this input file.*
      
      ![1. Column: FilePath_FileName] [2. Column: ElementType] [3. Column: ElementName] [4. Column: Optional Column for considering window around summit in case of TF Data]
    - 1. *column:* Provide the path to the file including file name, these files can be of type bed, narrowPeak, pk or any text file having genomic intervals with their chr name, start position and end position separated by tab character in each row.
    - 2. *column:* Supply the element type e.g.: TF for transcription factors or HISTONE for histone modifications (TF or HISTONE are just examples, you name it),
3. **column**: Provide the specific name of this element in each file. Important point is that each file must consist of same element's genomic intervals. e.g.: CTCF_GM12878, H3K27ME3_GM12878.

4. **column**: Provide this column for considering only a window around summit in bps for narrowPeak files. Fourth column is optional.

* Sample User Defined Library Input File can be reached at

```
~path/to/GLANET Folder/Data/demo_input_data/UserDefinedLibrary/
UserDefinedLibraryInputFile.txt
```

* The files referred in **UserDefinedLibraryInputFile.txt** are provided under

```
~path/to/GLANET Folder/Data/demo_input_data/UserDefinedLibrary/
TranscriptionFactors/
```

* Please note that all the files that will be used in User Defined Library Annotation have to reside on your local machine.

* Sample files are provided under ~path/to/GLANET Folder/Data/demo_input_data/UserDefinedLibrary/TranscriptionFactors directory.

* Please update the absolute path of these files accordingly in UserDefinedLibraryInputFile.txt.

* Therefore, please change the ~path/to/file column in **UserDefinedLibraryInputFile.txt** accordingly:

```
G:\GLANET_DATA\ENCODE\transcription_factors\spp.optimal.
→wgEncodeBroadHistoneGm12878CtcfStdAlnRep0_VS_
→wgEncodeBroadHistoneGm12878ControlStdAlnRep0.narrowPeak
```

User Defined Library Data Format (Mandatory if **User Defined Library Annotation** check box is checked.) Important point is that all the file/s listed in User Defined Library Input File must have same data format. GLANET supports four Data Formats:

* 0-based coordinates (End Exclusive)
* 0-based coordinates (End Inclusive)
* 1-based coordinates (End Exclusive)
* 1-based coordinates (End Inclusive)

Choose the appropriate the Data Format. Choose 0-based coordinates (End Exclusive) if you have selected

```
~path/to/GLANET Folder/Data/demo_input_data/UserDefinedLibrary/
UserDefinedLibraryInputFile.txt
```

as User Defined Library Input File.

Choose 0-based coordinates (End Exclusive) for bed, narrowPeak and pk files.

7. **Enrichment**: (Optional)

   • **Perform Enrichment**
Fig. 2.2: GLANET GUI Lower Part
Choose **Perform Enrichment** or **Perform Enrichment without Annotation** from the related combo box if you want to achieve Enrichment Analysis. GLANET will accomplish Enrichment Analysis for the element types checked in Annotation Options. When you choose **Perform Enrichment without Annotation**, GLANET will not do any Annotation analysis for the given input but only Enrichment analysis for them.

- **Perform Enrichment With Z-Scores** (Default)
- **Perform Enrichment Without Z-Scores**

GLANET can be run with **with z-score** and **without z-score** modes. These two modes lead to different memory consumptions.

- When the **with z-score** calculation is requested, GLANET keeps test statistic calculated for each of the \( B \) random sampling and for each element \( N \) in the memory. Accumulation of test statistics is accomplished for each chromosome one at a time and therefore does not require storing interval trees of all chromosomes in the memory concurrently. In this setting, memory consumption is \( O(M + B*N) \), where \( B \) is the number of total samplings, \( N \) is the number of genomic elements being tested and \( M \) is the size of one interval tree constructed for each chromosome.

- Running GLANET **without z-score** calculates the test statistic for each sampling and requires storing the interval trees for all chromosomes at the same time. Then, **without z-score** checks whether each sampling has test statistic greater than or equal to original (observed) test statistic. If yes, it increments the number of such samplings by one. As a result, although **without z-score** calculation increases memory consumption by requiring all of the interval trees for all chromosomes in memory at the same time, it decreases the memory consumption by not keeping test statistics for each sampling for each element but just the number of samplings (that have test statistic greater than or equal to the original test statistic) per element. Thus, its memory consumption is proportional to \( O(P*N) \), where \( P \) is the size of total number of interval trees.

- This trade-off between running GLANET in **with z-score** mode or in **without z-score** mode can be solved by considering the number of total samplings \( B \) times number of elements \( N \) being analysed. When \( B*N \) is in hundreds of thousands, running GLANET with **with z-score** option could be too memory consuming; therefore, we recommend using **without z-score** option.

**Generate Random Data Mode**
- With GC (default)
- With Mappability
- With GC and Mappability
- Without GC and Mappability

Choose the **Generate Random Data Mode** which can be **With GC**, **With Mappability**, **With GC and Mappability** or **Without GC and Mappability**. Default is **With GC**.

**Multiple Testing**
- Benjamini Hochberg FDR (Default)
- Bonferroni Correction

Select the **Multiple Testing procedure** which can be either **Bonferroni Correction** or **Benjamini Hochberg FDR**. In fact, GLANET performs both of the Multiple Testing procedures but results are sorted w.r.t. the selected Multiple Testing procedure. Default is **Benjamini Hochberg FDR**.

**False Discovery Rate (FDR)**
– Default False Discovery Rate (FDR) is 0.05.

• **Bonferroni Correction Significance Level**
  – Default Bonferroni Correction Significance Level is 0.05.

• **Number of Samplings**
  – Choose the number of samplings among 5000, 10000, 50000 and 100000 choices.
  – Start with smaller number of samplings, and increase number of samplings depending on your computer’s performance.

• **Number of samplings In Each Run**
  – Choose the number of samplings in each run among 1000, 5000 and 10000 choices.
  
  e.g.: Do not forget that increasing the number of runs increases the GLANET execution time.

  – If your system performance is high in terms of CPU and RAM, prefer minimum number of runs.
  – You may have 10000 samplings, by achieving 10000 samplings in each run, which makes 10000/10000 = 1 run at total.
  – Or you may have 10000 samplings, by achieving 1000 samplings in each run, which makes 10000/1000 = 10 runs at total.

8. **Regulatory Sequence Analysis:** (Optional)
   - Please note that Regulatory Sequence Analysis is enabled if you have checked at least one of the following Annotation Options such as TF, TF and KEGG Pathway or TF and KEGG Pathway (CellLine based).
   - If you want to perform Regulatory Sequence Analysis, you must check the Regulatory Sequence Analysis using RSAT check box.
   - Regulatory Sequence Analysis requires Annotation Output drop down box selected as Write Overlaps, Each One In Separate File (Element Based) or Write Overlaps, All In One File (Element Type Based).
   - Please notice that Regulatory Sequence Analysis is carried out for all of the annotated Transcription Factors.
   - Regulatory Sequence Analysis makes use of RSAT web services.

9. **Job Name:** (Optional)
   - Please give a job name, then a directory named with this job name will be created under ~path/to/GLANET Folder/Output/JobName/
   - Choose shorter job name so that all the sub folders to be created under ~path/to/GLANET Folder/Output/JobName/ directory will not exceed the allowable length.
   - Default is Job Name is NoName.

**Log File:**

GLANET run from GUI (by double clicking the GLANET.jar) will be logged in a log file. Log file will be saved under the same directory where the GLANET.jar resides.

**GLANET Command-Line Interface and Command-Line Arguments**

In the following table, command-line arguments and their prerequisite arguments, if any, are specified. A command-line argument is required if and only if its precondition argument(s) is specified. You must set at most one parameter per argument. For example, if you set both -f0 and -bed parameters for Input File Format argument, then the program
GLANET Documentation, Release 0.9.0

Details of the command-line arguments with sample runs are specified below. Note that parameter “-c” indicates that GLANET will run in command-line mode, not with GUI.

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameter</th>
<th>Optional/Required</th>
<th>Precondition</th>
<th>Default Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-line</td>
<td>-c</td>
<td>Optional</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Input File Name</td>
<td>-i</td>
<td>Required</td>
<td>-c</td>
<td>None (&quot;path/to/file&quot;)</td>
</tr>
<tr>
<td>Assembly</td>
<td>-grch37</td>
<td>Required</td>
<td>-c</td>
<td>-grch37</td>
</tr>
<tr>
<td>GLANET Folder</td>
<td>-g</td>
<td>Required</td>
<td>-c</td>
<td>None (&quot;path/to/file&quot;)</td>
</tr>
<tr>
<td>Output Folder</td>
<td>-o</td>
<td>Optional</td>
<td>-c</td>
<td>GlanetFolder/Output/</td>
</tr>
<tr>
<td>Input File Format</td>
<td>-f1</td>
<td>Required</td>
<td>-c</td>
<td>-fbed</td>
</tr>
<tr>
<td></td>
<td>-f0</td>
<td>Required</td>
<td>-c</td>
<td>-fbed</td>
</tr>
<tr>
<td></td>
<td>-fbed</td>
<td>Required</td>
<td>-c</td>
<td>-fbed</td>
</tr>
<tr>
<td></td>
<td>-fgff</td>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-fdbsnp</td>
<td>Optional</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Association Measure Type</td>
<td>-noob</td>
<td>Required</td>
<td>-c</td>
<td>-noob</td>
</tr>
<tr>
<td></td>
<td>-eoo</td>
<td>Required</td>
<td>-c</td>
<td>-noob</td>
</tr>
<tr>
<td>Number of Bases</td>
<td>-nb</td>
<td>Required</td>
<td>-eoo</td>
<td>1 (Integer)</td>
</tr>
<tr>
<td>Annotation</td>
<td>-dnase</td>
<td>Optional</td>
<td>-c</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>-histone</td>
<td>Optional</td>
<td>-c</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>-tf</td>
<td>Optional</td>
<td>-c</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>-gene</td>
<td>Optional</td>
<td>-c</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>-gobp</td>
<td>Optional</td>
<td>-c</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>-gomf</td>
<td>Optional</td>
<td>-c</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>-gocc</td>
<td>Optional</td>
<td>-c</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>-kegg</td>
<td>Optional</td>
<td>-c</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>-tfkegg</td>
<td>Optional</td>
<td>-c</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>-tfcellkegg</td>
<td>Optional</td>
<td>-c</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>-udgs</td>
<td>Optional</td>
<td>-c</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>-udl</td>
<td>Optional</td>
<td>-c</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>-udgsinput</td>
<td>Required</td>
<td>-udgs</td>
<td>None (&quot;path/to/file&quot;)</td>
</tr>
<tr>
<td></td>
<td>-geneid</td>
<td>Required</td>
<td>-udgs</td>
<td>-genesym</td>
</tr>
<tr>
<td></td>
<td>-genesym</td>
<td>Required</td>
<td>-udgs</td>
<td>-genesym</td>
</tr>
<tr>
<td></td>
<td>-generna</td>
<td>Optional</td>
<td>-udgs</td>
<td>&quot;NoName&quot; (String)</td>
</tr>
<tr>
<td></td>
<td>-udgsname</td>
<td>Optional</td>
<td>-udgs</td>
<td>None (&quot;path/to/file&quot;)</td>
</tr>
<tr>
<td></td>
<td>-udgslfile</td>
<td>Optional</td>
<td>-udgs</td>
<td>None (&quot;path/to/file&quot;)</td>
</tr>
<tr>
<td></td>
<td>-udlinput</td>
<td>Required</td>
<td>-udl</td>
<td>None (&quot;path/to/file&quot;)</td>
</tr>
<tr>
<td></td>
<td>-udldf0exc</td>
<td>Required</td>
<td>-udl</td>
<td>-udldf0exc</td>
</tr>
<tr>
<td></td>
<td>-udldf0inc</td>
<td>Required</td>
<td>-udl</td>
<td>-udldf0exc</td>
</tr>
<tr>
<td></td>
<td>-udldf1exc</td>
<td>Required</td>
<td>-udl</td>
<td>-udldf1exc</td>
</tr>
<tr>
<td></td>
<td>-udldf1inc</td>
<td>Required</td>
<td>-udl</td>
<td>-udldf1inc</td>
</tr>
<tr>
<td></td>
<td>-aos</td>
<td>Required</td>
<td>-c</td>
<td>-aos</td>
</tr>
<tr>
<td></td>
<td>-aoo</td>
<td>Required</td>
<td>-c</td>
<td>-aos</td>
</tr>
<tr>
<td></td>
<td>-aon</td>
<td>Required</td>
<td>-c</td>
<td>-aos</td>
</tr>
<tr>
<td>Enrichment</td>
<td>-e or -ewoa</td>
<td>Optional</td>
<td>-dnase or -histone or -gene or -kegg or -tfkegg or -udgs or -udl or -tfcellkegg</td>
<td>None</td>
</tr>
</tbody>
</table>

Continued on next page
### Table 2.1 – continued from previous page

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameter</th>
<th>Optional/Required</th>
<th>Precondition</th>
<th>Default Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-wzs</td>
<td>Required</td>
<td>-e</td>
<td>-wzs</td>
</tr>
<tr>
<td></td>
<td>-wozs</td>
<td>Required</td>
<td>-e</td>
<td>-wozs</td>
</tr>
<tr>
<td></td>
<td>-wgc</td>
<td>Required</td>
<td>-e</td>
<td>-wgc</td>
</tr>
<tr>
<td></td>
<td>-wm</td>
<td>Required</td>
<td>-e</td>
<td>-wm</td>
</tr>
<tr>
<td></td>
<td>-wgcm</td>
<td>Required</td>
<td>-e</td>
<td>-wgcm</td>
</tr>
<tr>
<td></td>
<td>-wogcm</td>
<td>Required</td>
<td>-e</td>
<td>-wogcm</td>
</tr>
<tr>
<td></td>
<td>-wif</td>
<td>Required</td>
<td>-e</td>
<td>-wif</td>
</tr>
<tr>
<td></td>
<td>-woif</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-bh</td>
<td>Required</td>
<td>-e</td>
<td>-bh</td>
</tr>
<tr>
<td></td>
<td>-bonf</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-fdr</td>
<td>Required</td>
<td>-e</td>
<td>0.05 (Float)</td>
</tr>
<tr>
<td></td>
<td>-sl</td>
<td>Required</td>
<td>-e</td>
<td>0.05 (Float)</td>
</tr>
<tr>
<td></td>
<td>-s</td>
<td>Required</td>
<td>-e</td>
<td>10000 (Integer)</td>
</tr>
<tr>
<td></td>
<td>-se</td>
<td>Required</td>
<td>-e</td>
<td>5000 (Integer)</td>
</tr>
<tr>
<td>RSA</td>
<td>-rsa</td>
<td>Optional</td>
<td>-tf or -tfkegg or -tfcellkegg</td>
<td>None</td>
</tr>
<tr>
<td>Job Name</td>
<td>-j</td>
<td>Optional</td>
<td>-c</td>
<td>“NoName” (String)</td>
</tr>
<tr>
<td>Number of Threads</td>
<td>-t</td>
<td>Optional</td>
<td>-c</td>
<td>80% of all processors</td>
</tr>
<tr>
<td>Log File</td>
<td>-l</td>
<td>Optional</td>
<td>-nl</td>
<td></td>
</tr>
</tbody>
</table>

### Command-Line Parameters Descriptions

There are several parameters that are either required or optional to make GLANET run in Terminal or in Command Prompt. Whether a parameter is required or not will be specified as we describe it. The order of parameters is not fixed. One may set the parameters in any order. Some parameters may require some other parameters to be set as preconditions, which will also be indicated. You can see the preconditions of a parameter as shown in *GLANET Command-Line Interface and Command-Line Arguments*

- **-c**

To enable GLANET to run in Terminal or Command Prompt, it must be indicated with `-c` option. If there is no such option specified, program will run with its graphical user interface.

- **-i**

Required if `-c` is set. Absolute input file location must be specified just after `-i` option.

- **-grch37**

Required if `-c` is set. This option specifies assembly of input data as GRCh37.p13. If you do not set anything, `-grch37` is set as default.
-grch38

**Required** if \(-c\) is set. This option specifies assembly of the input data as GRCh38. If you do not set anything, \(-grch37\) is set as default.

-g

**Required** if \(-c\) is set. Glanet folder location must be specified just after writing \(-g\). Do not forget that GLANET folder must have the Data folder as its sub folder.

-o

Specifies the output folder where the results will be written under. The folder location must be specified after \(-o\). If the folder does not exists, GLANET creates one.

-f1

**Required** if \(-c\) is set. One of the input data format options (\(-f1, -f0, -fb, -fgff, -fbsnp\)) must be specified. This option specifies that input file contains 1-based coordinates (End Inclusive) per line.

-f0

**Required** if \(-c\) is set. This option specifies that input file contains 0-based coordinates (End Inclusive) per line. See also \(-f1\).

-fbed

**Required** if \(-c\) is set. This option specifies that input file format is BED. See also \(-f1\).

-fgff

**Required** if \(-c\) is set. This option specifies that input file format is GFF3. See also \(-f1\).

-fdbsnp

**Required** if \(-c\) is set. This option specifies that input file contains dbSNP IDs per line. See also \(-f1\).

-noob

**Required** if \(-c\) is set. This option specifies that association measure type is Number of Overlapping Bases, it can be 0 or more. See also \(-eoo\).
**-eoo**

**Required if** `-c` **is set.** This option specifies that association measure type is Existence of Overlap, it is either 0 or 1. See also `-noob`.

**-nb**

**Required if** `-eoo` **is set.** This option sets the number of bases that must overlap in order to accept two intervals as overlapping intervals. A positive integer value must be specified as the parameter. If you do not set anything, default option is `-nb 1`.

**-dnase**

This option enables DNase Hypersensitive Sites (Cell Line Based) annotation.

**-histone**

This option enables Histone Modifications Sites (Cell Line Based) annotation.

**-gene**

This option enables clade: Mammal, genome: Human, assembly: Feb. 2009 (GRCh37/hg19), RefSeq Genes annotation.

**-tf**

This option enables Transcription Factors Binding Sites (Cell Line Based) annotation.

**-gobp**

This option enables Biological Process Gene Ontology annotation.

**-gomf**

This option enables Molecular Function Gene Ontology annotation.

**-gocc**

This option enables Cellular Component Gene Ontology annotation.

**-kegg**

This option enables KEGG Pathway annotation.
-tfkegg

This option enables joint Transcription Factor Binding Sites and KEGG Pathway annotation.

-tfcellkegg

This option enables joint Transcription Factor Binding Sites (Cell Line Based) and KEGG Pathway annotation.

-udgs

This option enables user defined gene set annotation.

-udgsinput

**Required** if -udgs is set. This option specifies user defined gene set input file. Absolute input file location must be specified as the parameter.

-genesym

**Required** if -udgs is set. This option specifies gene information type as “Gene Symbol”. One of the gene information type (-geneid, -genesym, -generna) must be specified. If you do not set any of these options, default option is -genesym.

-geneid

**Required** if -udgs is set. This option specifies gene information type as “Gene Id”. See also -genesym.

-generna

**Required** if -udgs is set. This option specifies gene information type as “RNA Nucleotide Accession”. See also -genesym.

-udgsname

This option gives a name for the user defined gene set.

-udgsdfile

This option specifies the user defined gene set description file location. Absolute file location must be specified as the parameter.

-udl

This option enables User Defined Library Annotation.

2.1. **Content:**
-udlinput

Required if -udl is set. This option specifies User Defined Library Input File. Absolute input file location must be specified as the parameter.

-udldf0exc

Required if -udl is set. This option specifies User Defined Library Data Format as “0-based coordinates (End exclusive)”. One of the data format (-udldf0exc, -udldf0inc, -udldf1exc, -udldf1inc) must be specified. If you do not set any of these options, default option is -udldf0exc.

-udldf0inc

Required if -udl is set. This option specifies User Defined Library Data Format as “0-based coordinates (End inclusive)”. See also -udldf0exc.

-udldf1exc

Required if -udl is set. This option specifies User Defined Library Data Format as “1-based coordinates (End exclusive)”. See also -udldf0inc.

-udldf1inc

Required if -udl is set. This option specifies User Defined Library Data Format as “1-based coordinates (End inclusive)”. See also -udldf0inc.

-aos

If this option is specified, GLANET outputs annotation results in element based separate files. See also -aoo.

-aoo

If this option is specified, GLANET outputs annotation results for all elements of the same element type in one file. See also -aos.

-aon

If this option is specified, GLANET does not output annotation results at all.

-e

If this option is specified, GLANET performs enrichment. Enrichment operation requires at least one annotation option (-dnase, -histone, -tf, -kegg, -tfkegg, -tfcellkegg, -udgs, -udl) to be set.
If this option is specified, GLANET performs enrichment without annotation. Enrichment operation requires at least one annotation option (-dnase, -histone, -tf, -kegg, -tfkegg, -tfcellkegg, -udgs, -udl) to be set.

If this option is specified, GLANET performs enrichment with z-scores.

If this option is specified, GLANET performs enrichment without z-scores. -wozs is only available for -eoo.

Required if -e or -ewoa are set. This option generates random data with GC. You must either set -wgc, -wm, -wgcm or -wogcm to specify generating random data mode. If you do not set anything, default option is -wgcm.

Required if -e or -ewoa are set. This option generates random data with Mappability. You must either set -wgc, -wm, -wgcm or -wogcm to specify generating random data mode. If you do not set anything, default option is -wgcm.

Required if -e or -ewoa are set. This option generates random data with GC and Mapability. You must either set -wgc, -wm, -wgcm or -wogcm to specify generating random data mode. If you do not set anything, default option is -wgcm.

Required if -e or -ewoa are set. This option generates random data without GC and Mapability. You must either set -wgc, -wm, -wgcm or -wogcm to specify generating random data mode. If you do not set anything, default option is -wgcm.

If this option is specified, GLANET performs random interval generation using random interval selected from corresponding isochore family pool. If you do not set anything, default option is -wif. See also -woif.

If this option is specified, GLANET performs random interval generation without isochore family pool. If you do not set anything, default option is -wif. See also -wif.
-bh

Required if -e or -ewoa are set. This option sets Multiple Testing correction as “Benjamini-Hochberg FDR”. You must either set -bh or -bonf to specify multiple testing correction option. If you do not set anything, default option is -bh.

-bonf

Required if -e or -ewoa are set. This option sets Multiple Testing correction as “Bonferroni Correction”. See also -bh.

-fdr

Required if -e or -ewoa are set. This option followed by a float value sets False Discovery Rate for Benjamini-Hochberg multiple testing. Default option is -fdr 0.05.

-sl

Required if -e or -ewoa are set. This option followed by a float value sets Significance Level for Bonferroni Correction multiple testing. Default option is -sl 0.05.

-s

Required if -e or -ewoa are set. This option followed by an integer value sets the total number of samplings in Enrichment. Default option is -s 10000.

-se

Required if -e or -ewoa are set. This option followed by an integer value sets number of samplings in each run. Default option is -se 5000.

-rsa

If this option is set, GLANET performs Regulatory Sequence Analysis using RSAT. -rsa requires that given intervals consist of SNPs and at least one of the following annotations to be set (-tf, -tfkegg -tfcellkegg) as preconditions. -rsa requires -aos or -aoo selected as GLANET annotation results output option. Otherwise Regulatory Sequence Analysis is not performed.

-j

It sets Job Name that GLANET is about to perform. It creates an output folder with the specified Job Name. Results will be collected under this folder. Job Name must be specified as the parameter. If you do not set anything, default option is -j NoName.
-t

This option followed by an integer value sets the number of threads allocated for the GLANET run. If the option is not specified, the default value is the 80% of the all processors available in the system. If the number of threads specified exceeds the maximum available threads in your system, maximum available threads will be used.

-l

If this option is set, GLANET Run is written to a log file. Log file will be saved under the directory where the GLANET command-line run is achieved.

-nl

If this option is set, GLANET Run is not written to a log file.

**GLANET DEMO for Annotation and Enrichment Analysis of OCD GWAS SNPs**

**GLANET Command Line Arguments:**

```
$ java -Xms16G -Xmx16G jar "path/to/GLANET.jar" -c -g "path/to/GLANET Folder/" -i "path/to/GLANET Folder/Data/demo_input_data/OCD_GWAS_chrNumber_1Based_GRCh37_p13_Coordinates.txt" -f1 -dnase -histone -tf -tfkegg -e -s 10000 -se 10000 -l -j "DEMO_OCD_GWAS_SNPs"
```

You can find a demo screencast for Annotation and Enrichment Analysis of OCD GWAS SNPs on youtube in the link below:

https://www.youtube.com/watch?v=e6tM71c6oII

This run took 19 minutes on Intel(R) Core i7-3630QM CPU, 2.40 GHz with 16GB RAM.

**GLANET DEMO for Annotation and Enrichment Analysis of srf.hg19.bed**

**GLANET Command Line Arguments:**

```
$ java -Xms4G -Xmx4G jar "path/to/GLANET.jar" -c -g "path/to/GLANET Folder/" -i "path/to/GLANET Folder/Data/demo_input_data/GAT_Comparison_Data/srf.hg19.bed" -fbed -tf -e -s 10000 -se 10000 -l -j "DEMO_SRF"
```

You can find a demo screencast for Annotation and Enrichment Analysis of srf.hg19.bed on youtube in the link below:

https://www.youtube.com/watch?v=PW2Oj-MzYkg

This run took 8 minutes on Intel(R) Core i7-3630QM CPU, 2.40 GHz with 16GB RAM.

**GLANET DEMO for Regulatory Sequence Analysis of OCD GWAS SNPs**

**GLANET Command Line Arguments:**

```
$ java -Xms4G -Xmx4G jar "path/to/GLANET.jar" -c -g "path/to/GLANET Folder/" -i "path/to/GLANET Folder/Data/demo_input_data/OCD_GWAS_SIGNIFICANT_SNP_RSIDs.txt" -fdbsnp -tf -rsa -l -j "DEMO_OCD_GWAS_SNPs_RSA"
```
You can find a demo screencast for Regulatory Sequence Analysis feature on youtube in the link below:
https://www.youtube.com/watch?v=AP63LgsJBIY

This run took 54 minutes on Intel(R) Core i7-3630QM CPU, 2.40 GHz with 16GB RAM.

**GLANET DEMO for User Defined Library Feature**

You can add your own library for annotation and enrichment analysis.

**In order to make a demo for User Defined Library Feature of GLANET using GUI:**

1. Open GLANET GUI allocated with 16GB amount of memory using command line arguments below:

   ```
   $ java -Xms16G -Xmx16G -jar "path/to/GLANET.jar"
   ```

2. Load “OCD_GWAS_chrNumber_1Based_GRCh37_p13_Coordinates.txt” under “path/to/GLANET Folder/Data/demo_input_data/” as Input File Name.

3. Choose “1-based coordinates (End Inclusive)” as Input Format.

4. Set GLANET Folder (“path/to/GLANET Folder/”) where GLANET Folder must be parent directory of Data directory.

5. Set Output Folder as you wish.

6. Check “User Defined Library Annotation” check box in the User Defined Library panel.


8. Choose “0-based coordinates (End Exclusive)” as User Defined Library Data Format.

9. Check “Perform Enrichment” Check Box.

10. Enter a “Job Name” as you wish. e.g.: DEMO_UDL

11. Leave the other options set as default.

12. Then click on Run button.

You can find a demo screencast for User Defined Library feature of GLANET on youtube in the link below:
https://www.youtube.com/watch?v=yJg_l2eEqB8

This run took 6 minutes on Intel(R) Core i7-3630QM CPU, 2.40 GHz with 16GB RAM.

**In order to make a demo for User Defined Library feature of GLANET using Command Line Arguments:**

```bash
$ java -Xms16G -Xmx16G -jar "path/to/GLANET.jar" -c -g "path/to/GLANET Folder/" -i "path/to/GLANET Folder/Data/demo_input_data/OCD_GWAS_chrNumber_1Based_GRCh37_p13_Coordinates.txt" -grch37 -f1 -noob -udl -udlinput "path/to/GLANET Folder/Data/demo_input_data/UserDefinedLibrary/UserDefinedLibraryInputFile.txt" -udldf0exc -e -wzs -wqcm -bh -s 10000 -se 10000 -i -j "DEMO_UDL"
```
GLANET Documentation, Release 0.9.0

GLANET DEMO for User Defined Gene Set Feature

You can add your own user defined gene sets for annotation and enrichment analysis.

In order to make a demo for User Defined Gene Set Feature of GLANET using GUI:

1. Open GLANET GUI allocated with 16GB amount of memory using command line arguments below:
   
   ```
   $ java -Xms16G -Xmx16G -jar "path/to/GLANET.jar"
   ```

2. Load “OCD_GWAS_chrNumber_1Based_GRCh37_p13_Coordinates.txt” under “path/to/GLANET Folder/Data/demo_input_data/” as Input File Name.
3. Choose “1-based coordinates (End Inclusive)” as Input Format.
4. Set GLANET Folder (“path/to/GLANET Folder/”) where GLANET Folder must be parent directory of Data directory.
5. Set Output Folder as you wish.
6. Check “User Defined Gene Set Annotation” check box in the User Defined Gene Set panel.
8. Set “Gene Symbol” as Gene Information Type.
9. Set Name as you wish, e.g.: GO, since User Defined Gene Set Input File consists of GO Terms.
10. Load “GO_ids2terms.txt” under “path/to/GLANET Folder/Data/demo_input_data/UserDefinedGeneSet/GO/” as Description File.
11. Check “Perform Enrichment” Check Box.
12. Enter a Job Name as you wish. e.g.: DEMO_UDGS
13. Let the other options set as default.
14. Then click on Run button.

![Fig. 2.4: GLANET User Defined Gene Set](image)

You can find a demo screencast for User Defined Gene Set feature on youtube in the link below:

https://www.youtube.com/watch?v=ErUMziglK4A

This run took 38 minutes on Intel(R) Core i7-3630QM CPU, 2.40 GHz with 16GB RAM.

In order to make a demo for User Defined Gene Set Feature of GLANET using Command Line Arguments:

2.1. Content: 27
GLANET Documentation, Release 0.9.0

GLANET DEMO for Annotation and Enrichment Analysis of OCD GWAS SNPs w.r.t. GO Terms

In order to make a demo for Annotation and Enrichment Analysis of OCD GWAS SNPs w.r.t. GO Terms using Command Line Arguments:

```
$ java -Xms16G -Xmx16G -jar "path/to/GLANET.jar" -c -g "path/to/GLANET Folder/" -i "path/to/GLANET Folder/Data/demo_input_data/OCD_GWAS_chrNumber_1Based_GRCh37_p13_Coordinates.txt" -grch37 -f1 -noob -udgs -udgsinput "path/to/GLANET Folder/Data/demo_input_data/UserDefinedGeneSet/GOTerm_GeneSymbol_EvidenceCode_Ontology.txt" -genresym -udgsname "GO" -udgsdfile "path/to/GLANET Folder/Data/demo_input_data/UserDefinedGeneSet/GO/GO_ids2terms.txt" -e -wgcm -s 10000 -se 10000 -l -j "DEMO_UDGS"
```

GLANET DEMO for Annotation and Enrichment Analysis of OCD GWAS SNPs w.r.t. KEGG Pathways

In order to make a demo for Annotation and Enrichment Analysis of OCD GWAS SNPs w.r.t. KEGG Pathways using Command Line Arguments:

```
$ java -Xms16G -Xmx16G -jar "path/to/GLANET.jar" -c -g "path/to/GLANET Folder/" -i "C:\Users\Burçak\Google Drive\Data\demo_input_data\OCD_GWAS_chrNumber_1Based_GRCh37_p13_Coordinates.txt" -grch37 -f1 -noob -kegg -e -wgcm -s 1000 -se 1000 -l -j "DEMO_Predefined_KEGGPathways"
```

Additional GLANET Command-Line Sample Runs

Example Annotation Run is as following (Number of threads specified explicitly):

```
$ java -Xms8G -Xmx8G jar "path/to/GLANET.jar" -c -g "path/to/GLANET Folder/" -i "path/to/GLANET Folder/Data/demo_input_data/CVD_all_1_based_coordinates.txt" -f1 -grch37 -eoo -dnase -histone -tf -kegg -tfkegg -l -j "SampleRun1" -t 16
```

Example Annotation and Enrichment Run is as following (Enrichment without ZScores):

```
$ java -Xms8G -Xmx8G jar "path/to/GLANET.jar" -c -g "path/to/GLANET Folder/" -i "path/to/GLANET Folder/Data/demo_input_data/OCD_GWAS_chrNumber_1Based_GRCh37_p13_Coordinates.txt" -f1 -dnase -histone -tfcellkegg -e -wozs -se 10000 -l -j "SampleRun2"
```

Example Annotation and Enrichment Run is as following (Enrichment without GC and Mapability):

```
$ java -Xms8G -Xmx8G jar "path/to/GLANET.jar" -c -g "path/to/GLANET Folder/" -i "path/to/GLANET Folder/Data/demo_input_data/OCD_GWAS_chrNumber_1Based_GRCh37_p13_Coordinates.txt" -f1 -dnase -histone -tfcellkegg -kegg -e -wgcm -se 10000 -l -j "SampleRun3"
```

Example Annotation (Number of Overlapping Bases is chosen as Association Measure Type) is as following:
2.1.5 Contact

If you need any help please contact me at burcak@ceng.metu.edu.tr

2.1.6 Citation


2.1.7 Licence

Copyright (c) <2015> <Burcak Otlu>
CHAPTER 3

Indices and tables

• genindex
• modindex
• search