## Contents

1 Overview ................................. 3
   1.1 Enrichment Statistics of Sum-of-weights Scores .................................. 3
   1.2 SaddleSum software ......................... 4

2 Standalone Program ....................... 5
   2.1 Downloading .............................. 5
   2.2 Building and installing executables ......................................................... 5
   2.3 Obtaining Term Datasets ..................... 5
   2.4 Using SaddleSum ............................. 6
   2.5 Examples .................................. 7
   2.6 Manpages ................................ 7
      saddlesum ................................ 7
      saddlesum-show-etd ......................... 10
   2.7 Structure of Extended Term Databases .................................................... 11
      Header .................................... 11
      Genes Database ............................. 11
      Term Namespaces ............................ 12

3 Web Service ............................... 14
   3.1 Input Form ............................... 14
   3.2 Output Page .............................. 15
      Term relationships network .................. 15
      Significant terms ........................... 15
      Query and database summary ................. 17
   3.3 Examples ................................ 18

4 Cytoscape Plugin .......................... 19
   4.1 Downloading and Installing ................ 19
   4.2 Building from Source ........................ 19
   4.3 Using CytoSaddleSum ....................... 19
      Starting plugin from Cytoscape ............. 19
      Configuration .............................. 20
      Setting up a query ......................... 21
      Working with the results .................. 22
      Example ................................... 26

5 Additional Information .................... 27
   5.1 License ................................ 27
   5.2 Reference ............................... 27
   5.3 Credits ................................. 27
   5.4 Acknowledgments ......................... 27
   5.5 News and Updates ......................... 28
1 Overview

Term enrichment analysis facilitates biological interpretation by assigning to experimentally/computationally obtained data annotation associated with terms from controlled vocabularies. This process usually involves obtaining statistical significance for each vocabulary term and using the most significant terms to describe a given set of biological entities, often associated with weights. Many existing enrichment methods require selections of (arbitrary number of) the most significant entities and/or do not account for weights of entities. Others either mandate extensive simulations to obtain statistics or make assumptions about data distribution that need not be justifiable. In addition, most methods have difficulty assigning correct statistical significance to terms with few entities.

Implementing the well-known Lugananni-Rice formula, we have developed a novel approach, called SaddleSum, that is free from all these undesirable constraints. It approximates the distribution of sum of weights asymptotically by saddlepoint method to arrive at analytically and computationally tractable form.

We evaluated SaddleSum against several existing methods and demonstrated its ability to adapt equally well to distributions with widely different properties. With entity weights properly taken into account, SaddleSum is internally consistent and stable with respect to the choice of number of most significant entities selected. Making few assumptions on the input data, the proposed method is universal and can thus be applied to areas beyond analysis of microarrays, such as deep sequencing, quantitative proteomics and in silico network simulations. SaddleSum also provides a term-size dependent score distribution function that gives rise to accurate statistical significance even for terms with few entities. As a consequence, SaddleSum enables researchers to place confidence in its significance assignments to small terms that are often biologically most specific.

1.1 Enrichment Statistics of Sum-of-weights Scores

Suppose that you have obtained some measurements for a number of genes or proteins in an organism. These can be gene expression log-ratios or values computed through some simulation. The problem now is to biologically interpret your experiment, that is, to link it to known terms and concepts.

One way to do so is to obtain a controlled vocabulary that covers the domain of your investigation and serves as basis for annotation of the genes from your organism. This means that every term in that vocabulary is associated with one or more genes; the term describes its associated genes. Your original problem is now transformed to the question of finding what terms best describe your overall measurements.

Let us further assume that gene measurements, which we will call weights, indicate relative importance of genes. If that is not so, for example if all highly expressed genes should be treated equally, you can transform your values so that this assumption holds and continue from there. A natural way to express the importance of each vocabulary term is to compute its score as the sum of weights of all genes mapping to that term.

Unfortunately, the score just calculated is not a good way to compare terms. Different terms may contain different numbers of genes and hence cover different parts of your measurement set. Thus, it may be possible that difference in scores of two terms is only due to their coverage. Normalizing by the number of genes may help but does not go far enough; there is still a question of the scale of the scores and their distribution.

The standard approach to this issue is statistical: we want to know the probability of a given score occurring by chance with respect to some null distribution. This probability is known as P-value and can be used as the statistical significance of the term. Note that there is a null distribution here associated with each term.
In our case, it is reasonable to assume that the scores are built by summing \( m \) independently and identically distributed weights, where \( m \) is the number of genes mapping to the term. If the term P-value is sufficiently small, say 0.01, we can conclude that it is very unlikely that term score has arisen by chance. Therefore, we are able to interpret our dataset using that term.

*SaddleSum* is a tool that automates the above procedure. It requires a term database and a collection of gene labels associated with weights. It outputs the most statistically significant terms together with their estimated statistical significance. To compute P-values, *SaddleSum* first estimates the distribution of weights using all the supplied weights. Then using this estimated distribution, it approximates the P-value of a score using the asymptotic saddlepoint approach developed by Lugananni and Rice. This estimate is extremely accurate, even for terms with very small number of associated genes or for very small P-values.

To correct for the fact that a term database usually has many terms that enhance the chance of random matches, it is necessary to perform *multiple hypothesis testing correction*. *SaddleSum* uses Bonferroni correction for multiple hypothesis testing, where raw term P-values are multiplied with the effective database size to give E-values. We define the effective database size as the number of terms in the term database that map to at least \( k \) genes corresponding to selected nodes, where \( k \) is the minimum term size. This parameter is calculated at the time of the query. By default, the calculated value is used to compute E-values but you can override it by changing the corresponding form entry. For example, raw P-values can be displayed by setting the effective database size to 1.

### 1.2 SaddleSum software

*SaddleSum* can be used through three interfaces: command line (standalone version), web and as a Cytoscape (http://www.cytoscape.org/) plugin.

The standalone version is written completely in the C programming language and has no dependencies beyond the gcc compiler and GNU make (although it is likely that it can be adapted to compile using other development tools). It provides a command line interface and uses term databases in a simple tab-delimited format (GMT) as well as in SaddleSum’s own ETD (Extended Term Database) format.

The web version provides a wrapper for the standalone version via a web form. The server script that interacts with the standalone version is written in Python programming language and is a part of *qmbpmn-tools* package, which also includes *ITM Probe*, a framework for analysis of information flow in interaction networks based on random walks. The *qmbpmn-tools* package depends on a number of external Python libraries and on the Graphviz suite for visualizing graphs.

Cytoscape is an open source platform for complex network analysis and visualization written in Java programming language. Apart for a rich set of graph visualization tools, it provides an interface for externally written plugins that provide additional functionality such as network analysis algorithms, database import and functional enrichment analysis. Cytoscape users are therefore able to combine algorithms and data from different sources to perform complex network-based analyses.

*CytoSaddleSum* is a Cytoscape plugin that enables *SaddleSum* queries from Cytoscape platforms. It can interact either with a locally installed command-line program directly, or through our web server. The results are presented as a Cytoscape network view showing term relationships. Term statistics, written to node attributes in the relationship network, can be manipulated by the user.
2 Standalone Program

The standalone version of SaddleSum is written in the C programming language and requires no additional libraries. It offers a standard UNIX command line interface and allows users to specify multiple term databases in the GMT format used by GSEA as well as in SaddleSum’s own ETD (Extended Term Database) format (see Structure of Extended Term Databases below).

2.1 Downloading


2.2 Building and installing executables

After downloading a source code, extract it to a chosen directory and enter the saddlesum-<version> directory. The standard process of:

```
./configure
make
make install
```

should install the binaries into your executable directory. To clean the build, type `make clean`. This requires gcc and GNU make.

We have successfully built the source on Linux systems, Mac OS X with gcc and on Windows XP using MinGW. It may be possible to build it on other platforms but we have not attempted to do so.

2.3 Obtaining Term Datasets

Term datasets corresponding to the three Gene Ontology (GO) categories and KEGG pathways are available for a number of species in both ETD and GMT format (gzipped). ETD datasets are recommended but GMT versions of the same data is provided for compatibility with the earlier version. The FTP link is ftp://ftp.ncbi.nih.gov/pub/qmbpmn/SaddleSum/term_datasets. The most recently built datasets are available under in the current subdirectory.

The naming of the GMT files follows the convention:

```
<database>-<species>.gmt.gz
```

The <database> field is either GO-<category> or KEGG, where category is one of mf (molecular function), bp (biological process) or cc (cellular component). The species field uses KEGG three-letter convention. The currently available species are:
<table>
<thead>
<tr>
<th>KEGG code</th>
<th>Species</th>
</tr>
</thead>
<tbody>
<tr>
<td>ath</td>
<td>Arabidopsis thaliana</td>
</tr>
<tr>
<td>cel</td>
<td>Caenorhabditis elegans</td>
</tr>
<tr>
<td>dre</td>
<td>Danio rerio</td>
</tr>
<tr>
<td>dme</td>
<td>Drosophila melanogaster</td>
</tr>
<tr>
<td>hsa</td>
<td>Homo sapiens</td>
</tr>
<tr>
<td>mmu</td>
<td>Mus musculus</td>
</tr>
<tr>
<td>pfa</td>
<td>Plasmodium falciparum</td>
</tr>
<tr>
<td>rno</td>
<td>Rattus norvegicus</td>
</tr>
<tr>
<td>sce</td>
<td>Saccharomyces cerevisiae</td>
</tr>
</tbody>
</table>

For example `GO-bp-pfa.gmt.gz` contains the GO biological process database for *Plasmodium falciparum*.

Each ETD file contains all provided term datasets for a single species (the three GO namespaces plus KEGG). The naming of the ETD files follows the convention:

`<species>.etd.gz`

The species field again uses the KEGG three letter codes shown above.

## 2.4 Using SaddleSum

Standalone *SaddleSum* has a simple command line interface:

```
saddlesum [options] <weights_file> [::<namespace>]<term_db> [::<namespace>:<term_db> ...]
```

For full list of options, type:

```
saddlesum -h
```

or see the *saddlesum* page in this manual.

The `<weights_file>` argument can either be a filename path or -, which means that the weights are to be taken from the command line. The weights are supplied in the two-column tab-delimited format, where the first column gives entity (gene) IDs, while the second gives their weights as floating point numbers. No comment lines or multiple columns are allowed.

Entity IDs depend on the term database being searched. Each ETD database contains, in addition to the term datasets, an abbreviated version of the NCBI Gene database for its species. Thus, when using an ETD database, SaddleSum is able to interpret entity IDs provided as Gene IDs, as gene symbols and as gene aliases. If provided symbol is ambiguous, SaddleSum reports a warning or an error, depending on whether the ambiguity can be resolved.

The GMT term databases supplied on our FTP site use NCBI Gene IDs to label genes and hence the weights supplied to `saddlesum` program must also use NCBI Gene IDs to supply weights. We have decided on using NCBI Gene IDs for this purpose because many genes have several widely-used names, while GMT format does not allow specifying aliases.

The database arguments are in the format `<namespace>:<term_db>` where `<namespace>` is an arbitrary namespace label and `<term_db>` is the path to the term database. The first database can be specified without the `<namespace>` label, which indicates that the `<term_db>` is an ETD file. If `<namespace>`
is present, a GMT dataset is assumed. All subsequent arguments must specify the namespace part and the term datasets must be in GMT format. In this way, it is possible to combine an ETD database with several GMT ones. The recommended use for GMT databases is to create customized databases containing associations that cannot be found in GO or KEGG.

Namespace labels for GMT databases can be repeated. Significant terms under the same namespace are sorted and printed together, with each results table headed by the `<namespace>` label. Term datasets within an ETD database already contain their namespace labels, which can be inspected using the `saddlesum-show-etd` utility.

### 2.5 Examples

The `examples/` subdirectory contains several examples of weights from microarrays (log2 ratios) and term databases. All the examples were taken from the NCBI GEO database and have the extension `.tab`. The database files are in GMT format and have extension `.gmt`. Copy `saddlesum` executable to your path and try the following examples:

```
>saddlesum GDS3184_GSM253560_up_geneid.tab test:GO-bp-rno.gmt
>saddlesum GDS3184_GSM253560_down_geneid.tab test:GO-bp-rno.gmt
>saddlesum GDS2338_GSM102668_up_geneid.tab test:KEGG-sce.gmt
>saddlesum GDS2338_GSM102668_down_geneid.tab test:KEGG-sce.gmt
>saddlesum GDS2352_GSM89756_up_geneid.tab test:GO-cc-hsa.gmt
>saddlesum GDS2352_GSM89756_down_geneid.tab test:GO-cc-hsa.gmt
```

One can also experiment with options, such as setting the cutoffs and similar. Note that the down-regulated weights were obtained by flipping the sign on the original log2 ratios and setting all weights smaller than zero to zero (please see our paper for the explanation).

### 2.6 Manpages

`saddlesum`

**SYNOPSIS**

The standalone `saddlesum` interface takes the following general form:

```
>saddlesum [options] <weights_file> [<namespace>:]<term_db> [<namespace>:<term_db> ...]
```

**OPTIONS**

**Arguments**

`<weights_file>`

Tab-delimited file with entity ids and weights. If `<weights_file>` is specified as `-`, use standard input.
<namespace>:<term_db>

The text up to the first column denotes the namespace label while the rest of the argument denotes a path to a term database. For the first database, the namespace component can be omitted, in which case the file is assumed to be in ETD format. All subsequent databases should be in GMT format and have namespace specified. This allows combining multiple databases to obtain joint results. Significant terms for each namespace are treated separately.

Generic options
- **h**
  Print a description of all command line options.
- **v**
  Print the version number and exit. The version number always matches that of the qmbpmn-tools.

Statistical options
- **m**<min_term_size>
  Set the minimum number of entities for a term to be considered (default = 2). Only entities with supplied weights count towards the term size.
- **e**<Evalue_cutoff>
  Set the largest E-value for a term to be considered significant (default = 1e-02).

  **Note:** Since saddlesum uses an algorithm that quickly rejects those terms that cannot have an E-value smaller than <Evalue_cutoff>, the choice of <Evalue_cutoff> can significantly affect the running time of the program.

- **n**<effective_db_size>
  Set the effective term database size for applying Bonferroni correction (i.e. calculating E-values) to P-values output by the algorithm (default = the total number of terms considered). Setting <effective_db_size> to 1.0 will result in the original P-values being returned.

- **s**<statistics_type>
  Set statistical method used to evaluate P-values. The argument must be one of the following:
  - **wsum**(default) use SaddleSum statistics based on sum-of-weights score for each term
  - **hgem** use one-sided Fisher’s Exact test.

  **Note:** Option -s hgem implies option -d (whether specified or not). Also, one of -r or -w options must be used to specify the cutoff for selecting weights.

- **a**
  When this option is specified, all weights that can be mapped to valid entities are used as statistical background. Otherwise, only those weights that both map to a valid entity and a vocabulary term in the term database are used.

- **x**<namespace>
  Exclude <namespace> from ETD database. Each ETD database may contain multiple namespaces. This option allows specific namespaces to be totally excluded from consideration. It affects the effec-
tive database and hence the term E-values. More than one namespace can be excluded by using this option multiple times.

Note: Use saddlesum-show-etd program to discover the names of all namespaces present in an ETD file.

-T <term_id>
Compute statistics only for the term with ID <term_id> and display the list of entities associated with that term, together with their weights. All other statistical and weight processing options can be specified in conjunction with -T but -e has no effect.

Note: If specifying -m results in the term to be excluded from computation of P-value, no statistics will be computed and displayed.

Weight processing options
-t <weight_transformation>
Apply a transformation to each of the provided weights prior to other applying other processing options (see below) and calculating enrichment statistics. The argument must be one of the following:

flip flip the sign of each weight
abs take the absolute value of each weight.

When this option is omitted, no transformation will be performed.

-r <rank_cutoff>
Set all weights ranked lower than <rank_cutoff> to 0. If there are several weights tied at <rank_cutoff>, keep all of them.

-w <weight_cutoff>
Set all weights smaller than <weight_cutoff> to 0.

Note: Only one of the -r and -w options can be set at the same time.

-d
Discretize weights. Set all weights greater than 0 to 1 and all those smaller than 0 to 0.

Note: The weight processing options are applied in this order: -t, then -r or -w and finally -d.

Output options
-O <output_file>
Output results to <output_file> instead of to the standard output.

-F <output_format>
Select output format. The argument must be one of the following:

txt (default) print results as formatted (pretty) text.

tab print results as a tab-delimited file. Different sections are separated by heading lines starting with # character.
-W
Print warnings about entity identifiers from the <weights_file>.

-U
Print ids from <weights_file> that are not present in the term databases.

---

**Note:** Options -W and -U apply only to text output. Tab-delimited output always contains sections containing warnings and unknown ids.

---

**saddlesum-show-etd**

**SYNOPSIS**

saddlesum-show-etd [options] <term_db>

**OPTIONS**

Arguments
<term_db>
A database in ETD format.

Generic options
-\nPrint a description of all command line options.
-\nPrint the version number and exit. The version number always matches that of the qmbpmn-tools.

Output options
-\nOnly list the namespaces present in the database, one per line. Otherwise, a more detailed info table is output.

**Note:** Specify -Ftab to obtain only the list of namespaces, without any other text.

-\ <output_file>
Output results to <output_file> instead of to the standard output.

-\ <output_format>
Select output format. The argument must be one of the following:

- txt *(default)* print results as formatted (pretty) text.
- tab print results as a tab-delimited file. Different sections are separated by heading lines starting with # character.
2.7 Structure of Extended Term Databases

Extended Term Databases (ETDs) are binary files that contain term databases used by SaddleSum. They are created through Python scripts from qmbpmn-tools. Here, we describe the structure of the binary format.

Each ETD consists of a header, genes database and one or more namespaces. The header contains the overall information about the database. Genes database is derived from NCBI Gene and serves to enable parsing of gene label aliases. Each namespace is a separate term database, containing terms, their relationships and mapping to genes.

Note: In the listing that follow we use C-like pseudocode to indicate the types of the fields. These resemble C declarations with a difference that array sizes are usually not fixed but need to be read from the file. The type uint32 here means 32-bit little-endian integer.

Header

The header part is written as following (comments are on the right):

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>char start_separator[8]</td>
<td>- always ‘EXTERMDB’</td>
</tr>
<tr>
<td>uint32 version_magic</td>
<td>- set to 1644632861 for the current version</td>
</tr>
<tr>
<td>uint32 db_name buflen</td>
<td></td>
</tr>
<tr>
<td>char db_name[db_name buflen]</td>
<td>- Full database name (description)</td>
</tr>
<tr>
<td>uint32 num_namespaces</td>
<td></td>
</tr>
<tr>
<td>uint32 namespaces buflen</td>
<td></td>
</tr>
<tr>
<td>char namespaces_buf[namespaces buflen]</td>
<td>- Names of all namespaces</td>
</tr>
</tbody>
</table>

Genes Database

The genes database is derived from NCBI Gene records for a given species. It contains a list of genes, their descriptions, their symbols and lists of conflicts. There are two types of conflicts. Type 1 conflict occurs when the canonical gene symbol for a gene is an alias of another gene. These are always resolved to mean the gene that has the conflicting label as the canonical symbol. Type 2 conflict occurs when a label is an alias of multiple genes but is not set as a canonical symbol for any gene. Such conflicts cannot be resolved.

The genes database structure is:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>char start_separator[8]</td>
<td>- always ‘NCBIGENE’</td>
</tr>
<tr>
<td>uint32 version_magic</td>
<td>- set to 1200900292 for the current version</td>
</tr>
<tr>
<td>uint32 metadata buflen</td>
<td></td>
</tr>
<tr>
<td>char metadata[metadata buflen]</td>
<td>- gene_info_file, NCBIGENE_URL_FMT</td>
</tr>
<tr>
<td>uint32 gene_info_checksum</td>
<td>- CRC32 checksum of gene_info_file</td>
</tr>
<tr>
<td>uint32 tax_id</td>
<td>- NCBI Taxonomy ID</td>
</tr>
<tr>
<td>uint32 N</td>
<td>- number of genes</td>
</tr>
<tr>
<td>uint32 gene_ids[N]</td>
<td>- assumes every NCBI Gene ID is uint32</td>
</tr>
<tr>
<td>uint32 offsets[N+1]</td>
<td>- offsets into gene_info_file</td>
</tr>
<tr>
<td>uint32 symbols_counts[N]</td>
<td>- counts of symbol lists for each gene</td>
</tr>
<tr>
<td>uint32 symbols buflen</td>
<td></td>
</tr>
<tr>
<td>char symbols_buf[symbols_len]</td>
<td>- N lists of symbols. Each symbol is terminated by ‘\0’</td>
</tr>
</tbody>
</table>
Each list contains one primary identifier and 0 or more synonyms.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uint32 descl_buflen</td>
<td>- gene descriptions</td>
</tr>
<tr>
<td>char descl[descl_buflen]</td>
<td></td>
</tr>
<tr>
<td>uint32 num_conf1</td>
<td>- number of conflicts (type 1)</td>
</tr>
<tr>
<td>uint32 conf1_counts[num_conf1]</td>
<td>- counts of conflict lists (type 1)</td>
</tr>
<tr>
<td>uint32 conf1_buflen</td>
<td>- A list of conflict lists (of type 1). Each symbol terminated by '\0'. The first item in each list conflicts with all the others.</td>
</tr>
<tr>
<td>char conf1_buf[conf1_buflen]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uint32 num_conf2</td>
<td>- number of conflicts (type 2)</td>
</tr>
<tr>
<td>uint32 conf2_counts[num_conf2]</td>
<td>- counts of conflict lists (type 2)</td>
</tr>
<tr>
<td>uint32 conf2_buflen</td>
<td>- A list of conflict lists (of type 2). Each symbol terminated by '\0'. The first item in each list conflicts with all the others.</td>
</tr>
<tr>
<td>char conf2_buf[conf2_buflen]</td>
<td></td>
</tr>
</tbody>
</table>

**Term Namespaces**

Each namespace is a separate term database. It contains terms (IDs and descriptions) their relationships and maps to genes. KEGG and Gene Ontology databases differ according to their metadata. The structure is:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>char start_separator[8]</td>
<td>- always 'TERMDBNS'</td>
</tr>
<tr>
<td>uint32 version_magic</td>
<td>- changes for each different version</td>
</tr>
<tr>
<td></td>
<td>Currently 2264738403 for KEGG, 2187050528 for Gene Ontology.</td>
</tr>
<tr>
<td>uint32 num_edgetypes</td>
<td>- number of term relationship types</td>
</tr>
<tr>
<td>uint32 edgetype_buflen</td>
<td></td>
</tr>
<tr>
<td>char edgetype_names_buf[edgetype_buflen]</td>
<td>- names for term relationships. Each name terminated by '\0'.</td>
</tr>
<tr>
<td>uint32 M</td>
<td>- number of terms</td>
</tr>
<tr>
<td>uint32 slim_flags[M]</td>
<td>- whether the terms are in reduced dataset (not currently used)</td>
</tr>
<tr>
<td>uint32 num_hits[M]</td>
<td>- counts of hits for each term</td>
</tr>
<tr>
<td>uint32 hits[M][num_hits]</td>
<td>- hits to genes. These map each term index k to an array of gene indices, of length num_hits[k]</td>
</tr>
<tr>
<td>uint32 termid_buflen</td>
<td>- term IDs. Each item terminated by '\0'.</td>
</tr>
<tr>
<td>char termid_buf[termid_buflen]</td>
<td></td>
</tr>
<tr>
<td>uint32 desc_buflen</td>
<td>- term descriptions, Each item terminated by '\0'.</td>
</tr>
<tr>
<td>char desc_buf[desc_buflen]</td>
<td></td>
</tr>
<tr>
<td>uint32 num_parents[M]</td>
<td>- numbers of parents of each term</td>
</tr>
<tr>
<td>uint32 parents[M][num_parents]</td>
<td></td>
</tr>
<tr>
<td>uint32 edgetypes[M][num_parents]</td>
<td>- relationships to parents (edgetype indices)</td>
</tr>
<tr>
<td>uint32 metadata_buflen</td>
<td>- metadata (provided by the caller). List of strings, each string terminated by '\0'.</td>
</tr>
<tr>
<td>char metadata[metadata_buflen]</td>
<td></td>
</tr>
</tbody>
</table>
KEGG namespaces use three items in metadata field: organism prefix, leaf URL format (URL format for the page describing a leaf term) and higher URL format (URL format for terms higher in the hierarchy). Gene Ontology namespaces have a single item; a url format for Amigo website. All URL formats must contain a single \%s formatting specifier.
3 Web Service

SaddleSum web service allows users to input their gene weights through a web form and receive results as HTML pages. The user interface consists of the input form, the output page and the term details page.

Note: SaddleSum web service runs standalone executable on the server and hence many of its options directly correspond to the command line options described on the man page for the saddlesum program available as an Appendix. We will here concentrate mostly on the aspects unique to the web service.

3.1 Input Form

Use the input form to set up SaddleSum queries. It consists of four sections: term database and weights, statistical parameters, weight processing parameters and output parameters.

![SaddleSum Query Form](image)

The only mandatory parameters are located in the Term Database and Weights section. You must choose an appropriate database from the dropdown box and enter gene labels with their weights. Since all databases
that can be selected are in ETD format (see *Obtaining Term Datasets*), you may enter gene labels as NCBI Gene IDs, as canonical gene names or as aliases. There are two ways to enter weights: through the text box or by uploading a text file. The format is the same in both cases: it consists of several lines where each line should contain a gene label string, followed by spaces or a TAB, followed by a floating point weight value.

The **Statistical Parameters** determine which database terms are retrieved as significant. These are E-value cutoff, minimum term size, effective database size and statistical method. See *Enrichment Statistics of Sum-of-weights Scores* for an explanation of *SaddleSum* statistics and the `saddlesum` man page for the detailed description of these options. Note that apart from the default Lugannani-Rice statistics, it is also possible to select One-sided Fisher’s Exact test statistics, which are based on the hypergeometric distribution. In that case, you must select a cutoff under the weight processing parameters.

The **Weight Processing Parameters** can be used to easily perform the most common transformations of weights and set weight cutoffs. Again, the `saddlesum` man page contains all the details about the allowed values.

The look of the *SaddleSum* output is controlled through the **Output Parameters** section. You can select the output format (**HTML**, **text** or **tab**), the format of the term relationships graph image (see below) and its color scheme. If you choose **text** or **tab** as output format, *SaddleSum* will not produce a term relationship graph.

To run *SaddleSum* press the **QUERY** button.

### 3.2 Output Page

To display query results *SaddleSum* produces an output page showing an image of the term relationships network, the tables listing significant terms, and a query and database summary.

**Term relationships network**

is placed at the top of the results. It contains terms as nodes linked by their relationships supplied by the term database they belong to. Of the databases currently by the web service, Gene Ontology has term relationships as a directed acyclic graph, while KEGG has a tree hierarchy. Only the significant terms according to the E-value cutoff plus their hierarchical parents are shown. Term significance (log E-value) is indicated by node colors. Clicking a term node opens a new window or tab in the browser that show the full description of the corresponding term.

**Significant terms**

are listed in the tables below the term graph. Each namespace with significant terms has its own table. A namespace table shows for each significant term its database identifier, description, the number of associations with genes, score and E-value.

Clicking on term identifier opens a new window or tab in the browser showing the full term description, while clicking on the last three columns brings up a statistical summary of the term and a full list of all genes associated with it together with their weights.
Query and database summary

is placed at the bottom of the results page. It echoes most of the input parameters, such as the name of the selected database, statistical parameters and weight transformation. In addition, it shows the statistics collected during the execution of the query that depend on the actual submitted gene identifiers and weights. These include the numbers of submitted and valid gene identifiers, total and used terms, and unused gene identifiers. Finally, you can toggle a list of unrecognized identifiers, if there were any.
3.3 Examples

The page https://www.ncbi.nlm.nih.gov/CBBresearch/Yu/mn/enrich/examples.html contains links to three examples of collections of weights that could be used with SaddleSum. The links lead to SaddleSum query forms filled with the log2 ratios of gene expressions, taken from selected microarrays from the NCBI GEO database. These are the same weights available as examples for the command-line version (Examples). Once you reach the query forms for an example, you only need to press the QUERY button to run it and obtain the results.
4 Cytoscape Plugin

CytoSaddleSum is a Cytoscape plugin that provides interface to SaddleSum functionality. It works by querying SaddleSum either locally (using the standalone version) or remotely, through an HTTP request to a web server. The results of a functional enrichment query are shown as a term relationship network, where nodes represent terms linked by edges showing term relationships. The statistics and other query details for each term are written as node attributes. This allows easy integration of SaddleSum into a network-based data analysis workflow.

CytoSaddleSum was developed for Cytoscape version 2.8. All the source code written at the NCBI is released into public domain.

4.1 Downloading and Installing

It is possible to download CytoSaddleSum either as a JAR file ready for installation as a Cytoscape plugin or as zipped source code archive. Both can be found on the NCBI FTP site (ftp://ftp.ncbi.nih.gov/pub/qmbpmn/CytoSaddleSum/). Releases of CytoSaddleSum share version numbers with other SaddleSum interfaces, starting with 1.4.

To install CytoSaddleSum, copy the JAR file you have downloaded to the plugins subdirectory of your Cytoscape distribution. If the installation location was correct, you would see the SaddleSum entry in the Cytoscape Plugins menu.

4.2 Building from Source

The source code was mostly developed using NetBeans (http://www.netbeans.org/) and built using Apache Ant (http://ant.apache.org/). To build the JAR file, you first need to unzip the source distribution file, put cytoscape.jar onto your CLASSPATH and copy the file build.xml.git in the root of the distribution to build.xml. Then type:

    ant package

To build the JAR named CytoSaddleSum.jar in the root distribution directory.

4.3 Using CytoSaddleSum

Starting plugin from Cytoscape

Start CytoSaddleSum by choosing the Cytoscape menu entry Plugins → CytoSaddleSum → Query Form. After reading the configuration file (see below), CytoSaddleSum attempts to retrieve the list of available term databases that can be selected for a query. After that, it creates a query form and inserts it into the Cytoscape Control Panel (on the left of the Cytoscape window). You may need to resize the Control Panel to see the entire SaddleSum query form.
The query form has a look and functionality similar to the query form of SaddleSum web interface. Three action buttons can be found at its bottom:

- QUERY - to start running a SaddleSum query (see Setting up a query);
- RESET - to reset the form;
- CONFIG - to change CytoSaddleSum configuration (see Configuration).
- LOAD - to load CytoSaddleSum results saved as attributes in a term relationship graph (see Saving and restoring results).

Configuration

Configuration of CytoSaddleSum is handled though a separate dialog.

There are two possible configurations:

**Web Query**  This option will configure CytoSaddleSum to query the SaddleSum web service over HTTP. Choosing this method means that you are not required to download, compile or configure the command-line version of SaddleSum. You may notice some latency when starting and running CytoSaddleSum and you will not be able to use custom GMT formatted term databases.
**Local Query** This option will configure *CytoSaddleSum* to run a local program. Choosing this method means that you need to download and compile the command-line version of *SaddleSum*. Queries will be generally faster and it will be possible to use custom GMT databases as well as ETD databases downloaded from the *SaddleSum* FTP site.

### Configuring for web queries

Configuring *CytoSaddleSum* to perform a web query requires clicking the “Web query” radio button and entering the URL for the *SaddleSum* web services into the “Web query URL” box.

**Note:** When creating the initial configuration, *CytoSaddleSum* will automatically set the URL of the default *SaddleSum* web service in the “Web query URL” box and this need not to be changed unless the server address changes.

### Configuring for local queries

Configuring *CytoSaddleSum* to perform a local query requires more work than configuring it to perform a web query. First, you need to download, compile and install the standalone *SaddleSum* (see *Standalone Program*) and, optionally, some ETD databases that you may use. In the configuration dialog you need to click the “Local query” radio button and set the paths for:

- The *SaddleSum* binary (for example `/usr/local/bin/saddlesum`)
- The directory holding downloaded ETD files.

**Note:** *CytoSaddleSum* assumes that the second binary from the standalone package (`saddlesum-show-etd`) can be located in the same directory as `saddlesum`.

**Note:** Downloading ETD files is optional but if none are present in the specified directory, only local queries using custom GMT term databases will be enabled.

### Setting up a query

*CytoSaddleSum* operates on the currently selected Cytoscape network. Before running a query, you need to have a network with at least one floating point node attribute that can serve as node weight. The always-present `canonicalName` attribute provides node identifiers for the query, while the weight attribute can be selected through a dropdown box on the query form. Select all the nodes in the network that you wish to be used for the query (see Warning below).

After selecting the network and the nodes within it, select an appropriate term database for your query. This may be a species-specific ETD database from the dropdown list (containing Gene Ontology and KEGG pathways) or a custom database in GMT format. GMT format, used by the GSEA tool is described at [http://www.broadinstitute.org/cancer/software/gsea/wiki/index.php/Data_formats](http://www.broadinstitute.org/cancer/software/gsea/wiki/index.php/Data_formats)
Next, choose statistical and weight processing parameters for the query. We recommend using the default values. Short description of each parameter is available through tooltips, while full descriptions can be found on the man page for the command-line saddlesum program.

For queries using the default Lugannani–Rice statistics with sum-of-weights score, the weights may take any floating-point value: the score for a term is the sum of the weights for each node mapping to it. The other option for statistics is One-sided Fisher’s Exact test or equivalently, hypergeometric distribution. This gives the statistics reported by most other functional enrichment tools. In order to use these statistics, you have to select a cutoff, either By Rank (e.g. select nodes with top 50 weights) or By weight (e.g. select all nodes with weight greater than 0.05). The value of the cutoff parameter (rank (e.g. 50) or weight (e.g. 0.05) should be entered in the “Cutoff Value” box.

**Warning:** Selection of nodes has different meaning in CytoSaddleSum than in some other enrichment tools available in Cytoscape. Here, the node selection effectively determines the statistical background for the enrichment query. All selected nodes are considered, even those without a value for the weight attribute. Such nodes are given the weight of 0. The “significant” nodes for the purpose of the Fisher’s exact test are chosen using the cutoff parameters. This means that some nodes not of interest can be totally excluded from the enrichment analysis.

To start a query, press the QUERY button. A progress dialog will appear and, after a while, the query results (or an error message) will be shown.

### Working with the results

For each query, CytoSaddleSum results consist of a term relationship graph and a set of tables inserted into the Cytoscape “Results Panel” (located on the right of the Cytoscape main window when docked). To distinguish different queries, each query has a distinct name consisting of the prefix SSUM followed by a three-digit number (for example SSUM019). This number is incremented every time a query is run up to 999 and then reset to 0.
Note: The summary tables and the term relationship graph for a query are linked. Destroying a term relationship graph will also remove the corresponding summary tables from the Results Panel.

Summary tables

At the completion of each query a new tab (labeled with the query name described above) is embedded into Results Panel. This tab contains four tabs titled “Results”, “Summary”, “Warnings” and “Unknown IDs”. The Results tab contains tables of significant terms found by SaddleSum, separated by namespace (e.g. Biological Process, KEGG Pathway ...). Each row shows term ID, term description, total number of entities (genes) associated with that term, term score (sum of the weights of entities mapped to that term) and E-value (statistical significance; expected number of terms that would score no worse than that term under null model). If there are no results, the Results tab will be empty and titled “No Results”.

The Summary tab contains the query and database summary, including statistics on database size, number of submitted weights and similar. The Warnings and Unknown IDs tabs show information about the node IDs that were either not recognized within the term database (Unknown IDs) or were recognized as potential conflicts (Warnings). A conflict occurs when the submitted node ID could be interpreted as an alias to more than one entity (gene) within the database. Most conflicts can be resolved as one name takes precedence over others but in some instances the labeling is totally ambiguous.

Term relationship network

The term relationship network shows graphically the significant database terms retrieved by the query and the relationships between them. Major term databases such as Gene Ontology and KEGG are do not only
consist of mappings of genes to terms, but also contain hierarchical relationships between them. The SaddleSum ETD format databases contain such relationships and thus SaddleSum is able to display a graph of significant terms, as well as their list.

Each term relationship network consists of terms as nodes and their relationships as directed links. Each node is associated with Cytoscape node attributes corresponding to result table columns. Results from ETD databases contain not only the significant terms as nodes, but also the terms higher in the hierarchy, up to the root of the hierarchy. These non-significant terms do not have all attributes set. GMT databases do not contain relationships between terms and hence the resulting term network contains only nodes without any links. The term network is empty if SaddleSum query produces no results.

CytoSaddleSum applies a custom visual style to term relationship networks. Term IDs serve as node labels, while term descriptions serve as tooltips. Statistical significance (E-value) for each node is indicated by color, with significance increasing from light to dark blue. The color scheme is discretized based on the logarithm of E-value and the colors belong to the Blues8 series of Brewer Colors (see Acknowledgments).

Built-in Cytoscape plugins and utilities allow easy access to additional resources related to terms. For example, to browse a detailed description of a Gene Ontology term, right-click a node in a term relationship network. In the popup menu that appears choose LinkOut → Ontology → Gene Ontology (Quick GO by ID). A full description of the term will appear in your web browser. For KEGG pathways terms (lowest in the hierarchy), use LinkOut → KEGG → All Species (Use KEGG ID).

Selecting nodes mapped to significant terms

Double clicking on a row in a namespace table in the “Results” tab will select the corresponding term node in the term relationship graph. If the original graph used for the query is present, it will be placed in focus and all the nodes mapped to that term will be selected. You can then easily examine the weights and other attributes of these nodes through Data Panel.

Saving and restoring results

Each CytoSaddleSum query stores its results by setting node and network attributes in the term relationship graph. This network and its attributes can be saved through Cytoscape and then be reloaded in a different session. To restore the corresponding results panel, select a term relationship network and hit the “LOAD” button on the query form.

Note: The “LOAD” button will be enabled only if the selected network contains saved results and these results are not already shown in Results Panel.

Warning: Since all SaddleSum results are placed into node and network attributes, it is possible to arbitrarily edit them and change them. This can cause unpredictable effects to the functioning of CytoSaddleSum. Make sure you have the original results saved before any editing so you can restore them if problems occur.
Exporting results to text files

The results stored in a term relationship network can be exported to text files though the Export menu. Use File → Export → SaddleSum Results as TXT File... to export in plain text format or File → Export → SaddleSum Results as TAB File... to export in tab-delimited format. The result formats are identical to the ones used by the standalone saddlesum program. The text format is human-readable but does not contain the information necessary to generate the term relationship network. The tab-delimited format may not be as easy to read but contains the entire set of results, including the nodes and edges of the term relationship network.

Importing results from TAB files

The TAB files produced by the standalone saddlesum program or through Export functionality can be used directly to recreate the term relationship network and the results panel in Cytoscape. Use the menu item File → Import → Import SaddleSum Results from TAB File... to select and load the results. A term relationship network will appear, together with a corresponding results panel. CytoSaddleSum will assign a new label to the term relationship network and the results panel, as if the results originated from a direct SaddleSum query.

Note: The results loaded in this way will not be connected to any “original” graph. Hence, you will not be able to select the nodes mapped to terms by clicking on the rows in the namespace table of the panel corresponding to the imported results.

Troubleshooting missing results

Sometimes a query produces no significant results without apparent cause. This list may help in that case:

1. Check that a database for the appropriate species is selected. An indicator for this problem would be too many Unknown IDs.

2. Ensure that node labels correspond to database gene labels. ETD databases contain list of aliases for each gene and hence are able to interpret non-canonical node IDs. To use GMT databases, your node IDs must correspond to those used by term mappings.

3. Ensure that the correct network and the correct weight attribute is selected when pressing the QUERY button.

4. Check if all important nodes within the network are selected - only the weights of selected nodes will be considered.

5. Check that E-value cutoff is sensible. Sometimes it is necessary to enter a larger cutoff (such as 1 or even 10) for some terms to appear in results.

6. Ensure that you have selected a cutoff when using One-sided Fisher’s Exact test statistics.

7. Check other input parameters as they may significantly influence results as well - use defaults if unsure.
Example

Here is a step-by-step example to help you get started with CytoSaddleSum. It involves the microarray log-ratios from an experiment in yeast investigating caffeine and rapamycin effect on various wild type strains. The same example is available for the standalone and web service version of SaddleSum.

**Note:** This tutorial assumes basic familiarity with Cytoscape and that the CytoSaddleSum plugin is already properly installed.


2. Create a new directory and extract the archive in it.


4. Activate CytoSaddleSum from plugins menu. You may need to enlarge the Control Panel to see the entire query form. Loading database info from a remote server may take a while.

5. Using the Cytoscape import dialog accessible from File → Import → Network (All file types)..., import the yeast PPI network from the extracted yeast-ppi.sif file.

6. Select the new network and import the canonicalName and weights attributes from the extracted canonicalName.NA and weights.NA files. You can use VizMapper to use the canonicalName attribute as node label rather than ID.

7. Make sure that the imported network is selected and select all nodes within it.

8. Activate the CytoSaddleSum query form and select GO + KEGG: Saccharomyces cerevisiae as the term database.

9. Hit the QUERY button. After a while you will see a new network created and a new tab in the Results Panel.

10. You can browse the result tables on the right. In this case many terms are significant, mostly having to do with the ribosome. Zoom into the term relationship network and observe the term hierarchy. If you right-click on any node, you can obtain more information about the corresponding term using LinkOut.

11. Click on any terms to highlight the corresponding nodes in the original graph.

12. This finishes the step-by-step tutorial. You can now explore the rest of the CytoSaddleSum interface and perhaps rerun the query with different statistical and weight options. One interesting idea is to select only the nodes with non-null weights using Cytoscape filters. What will be the effect on term E-values?
5 Additional Information

5.1 License

All code for SaddleSum and CytoSaddleSum written at the NCBI is released into Public Domain. The licenses of external components are indicated in the source packages.

5.2 Reference

If you found SaddleSum useful, please cite:


5.3 Credits

- Aleksandar Stojmirovic and Yi-Kuo Yu for designed the study, conducted the research and write the paper;
- Aleksandar Stojmirovic wrote most of the code for the web and the standalone version and a part of the Cytoscape plugin;
- Alexander Bliskovsky wrote most of the Cytoscape plugin as well as the code to import KEGG datasets and to download and maintain all term databases.

5.4 Acknowledgments

Standalone SaddleSum uses C code from the Cephes library by Stephen L. Moshier and a hashtable library by Christopher Clark.

SaddleSum web server is written in the Python (http://www.python.org/) programming language and relies on several open-source components:

- Numpy and Scipy (http://www.scipy.org/) libraries of scientific tools for Python,
- Graphviz (http://www.graphviz.org/) graph visualization software,
- Jinja2 (http://jinja.pocoo.org/2/) template engine.

The web browser client code uses various Javascript routines from the NCBI and elsewhere. ECMAScript code and widgets for the SVG ‘Network Navigator’ were taken from Carto:Net (http://www.carto.net/)

All discrete network image color schemes were taken from the www.ColorBrewer.org (http://www.ColorBrewer.org) site by Cynthia A. Brewer, Geography, Pennsylvania State University.

CytoSaddleSum uses Apache HttpComponents (http://hc.apache.org/) library for HTTP requests.

We are grateful to Zvezdana Stojmirovic for help with the graphic design of the SaddleSum web pages and interfaces.

5.5 News and Updates

- **02-Mar-2017.** The documentation was changed to correct a mistake.

- **14-Dec-2011.** Updated CytoSaddleSum to version 1.5. The new version allows users to export results into files in plain text or TAB format and to import them from TAB files. The web form now allows results to be received in TAB format as well. Standalone SaddleSum 1.5.0 is identical to SaddleSum 1.4.0 except for the documentation.


