

Blast2GO Command Line User Manual

Version 1.3.3 November 2016



BioBam Bioinformatics S.L.
Valencia, Spain

Contents

1	Introduction	1
1.1	Main characteristics	2
1.2	Main features	2
1.3	Developed by	2
2	Setup	3
2.1	System Requirements	3
2.2	Product Activation	3
2.3	Create a Properties File	3
2.4	Setting up a local Blast2GO GO Database	4
3	Command Line Parameters	6
3.1	Generic Export	8
4	Use Case Examples	9
4.1	Examples	9

1 Introduction

Support: clisupport@blast2go.com

Website: <https://www.blast2go.com>

A Functional Annotation Pipeline: The Blast2GO Command Line is a professional solution for flexible, high-performance and automatic functional annotation tasks. This Annotation Pipeline allows you to integrate and automate your functional annotation task in a flexible way. Generate high-quality results in a reproducible way directly integrated into your data analysis workflows. The Blast2GO Command Line (CLI) version allows to combine BLAST and InterPro XML results, to perform the mapping against the Gene Ontology database and to assign the most suitable functional labels to the un-characterized sequence dataset. The command line allows to run the GO analysis on a local database which can be easily setup from the command line. Additionally the CLI allows to perform GO-Slim and Annex, to generate over 25 different summary charts and a comprehensive PDF report of each analysis.

The Command Line is based on the Blast2GO methodology, first published in 2005, [Conesa et al., 2005] for the functional annotation and analysis of gene or protein sequences. The method uses local sequence alignments (BLAST) to find similar sequences (potential homologous) for one or several input sequences. The program extracts all Gene Ontology (GO) terms associated to each of the obtained hits and returns an evaluated GO annotation for all query sequence(s). Enzyme codes are obtained by mapping to equivalent GOs and InterPro motifs can directly be added to the BLAST based annotation. A basic annotation process with Blast2GO consists of 4 steps: blasting and interpro-scan, GO mapping and functional annotation. [Götz et al., 2008]

```
Blast2GO Command Line v0.9.0 build: 20131217

BLAST2GO

usage: blast2go_cli [-annex] [-annotation] [-creategodb <files>] [--createkeyfile] [-gograph
<graphs>] [-goslim <name>] [-help] [-loadannot <path>] [-loadblast <path>] [-loaddat <path>]
[-loadfasta <path>] [-loadips48 <paths>] [-loadips50 <paths>] [-mapping] [-nodat] [-noreport]
[-overwrite] [-projectfolder <path-to-folder>] [-projectname <name>] [-protein] [-saveannot]
[-saveseqtable] [-showlicenseinfo] [-statistics <charts>] [-tempfolder <path-to-folder>]

options:
  -annex                Run ANNEX
  -annotation           Run AN
  -creategodb <files>  server you specify. The
                        loaded automatically, you can
                        properties file.
  --createkeyfile       annotation information, the file
                        in the current folder
  -gograph <graphs>   s e.g. mf.bp.cc
                        m_generic.obo,
                        lim_yeast.obo,
                        m_candida.obo,
  -goslim <name>       protein switch if
  -help
  -loadannot <path>   file (default: false)
  -loadblast <path>   create pdf report (default: false)
  -loaddat <path>     overwrite project folder (default: false)
  -loadfasta <path>   Project folder (default: current folder)
  -loadips48 <path>   Project name (default: b2g.project)
  -loadips50 <path>   Interpret fasta file content as proteins, only makes sense wh
  -mapping              used together with -loadfasta option.
  -nodat               Saves the .annot file after finishing the analysis
  -noreport            Saves a tab separated text file that contains the presentati
  -overwrite           your data as it would be shown in the Blast2GO GUI version
  -projectfolder <path-to-folder> Show details about the currently available license.
  -projectname <name>  Comma separated list of desired statistical charts (try
  -protein             -statistics without options to get a list of all available
                        charts)
  -saveannot           Path to temporary folder (default: <homefolder>/blast2go_
  -saveseqtable       can be shared among different projects.
  -showlicenseinfo
  -statistics <charts>
  -tempfolder <path-to-folder>
```

1.1 Main characteristics

- **High Performance** The command-line version of Blast2GO allows you to analyse large datasets on your own computing servers without a graphical user interface.
- **Flexible** Easily integrate your functional annotation tasks within your custom analysis pipeline and run different analysis scenarios in parallel.
- **Automatic Data Generation** Generate all the statistics charts Blast2GO offers in an automatic fashion. This includes a summary report in PDF as well as different images and text file formats.
- **Reproducibility** Control the whole analysis with a simple configuration file. This allows you to set up different analysis strategies and reproduce the multiple scenario for one or various datasets.
- **Secure** Run BLAST, InterProScan and the Blast2GO annotation offline on your own servers according to your security requirements. Take 100% control of data sources and versions you use throughout the analysis.

1.2 Main features

- Perform Blast (Cloud/Local) directly from the Blast2GO Command Line
- Perform InterProScan from the Command line (online feature)
- Semi-automatic local GO database setup and update
- Run Blast2GO on your own servers and control all analysis steps from the command line
- Automate your functional annotation
- Reproduce your results in a consistent manner
- Handle tens of thousand of sequences
- Design advanced annotation strategies
- Integrate Blast2GO into your existing analysis pipeline
- Work offline with your own resources
- Create your own local Blast2GO database
- Fast import of BLAST and InterProScan results
- Automatically generate PDF Reports
- Save all your results to specific project folders
- Work consistent and effective once you defined the right settings for your analysis

1.3 Developed by

Blast2GO Command Line is developed and maintained by BioBam Bioinformatics which is internationally recognised for its expertise in functional annotation and genome analysis.

2 Setup

2.1 System Requirements

Blast2GO Command Line (CLI) is a Java application and can be run on Mac, Linux and Windows 64-bit systems. It is always necessary to have Java 64-bit (version 1.6 or higher preferably from Sun/Oracle) installed, at least 1GB of RAM is recommended. The Blast2GO Command Line needs a Blast2GO database (DB) to perform the mapping step. This DB can be generated with the CLI itself; however the previous installation and configuration of a MySQL server (GPL license) is necessary.

Note:

With MySQL server version higher than 5.6 it is necessary to change the default configuration: Please adapt your MySQL server configuration (probably `/etc/mysql/my.cnf`) and add this option.

```
[mysqld]
sql_mode = STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE, \
ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION
```

Note:

In general this program works offline, however the CloudBlast and InterProScan need a working internet connection. The creation of a local GO database can be done offline as well by downloading the necessary data files manually.

2.2 Product Activation

Blast2GO offers 2 types of product activation. Either bound to a specific hardware or via a floating license. Floating licenses are served by a RLM license server in the same network and the setup instructions are provided upon request.

This section describes how to activate the product on a specific hardware (either for evaluation or perpetual). You will need your Activation Key for these steps. The license can be obtained automatically from www.blast2go.com/cli-activation. To do so, a signature of the workstation has to be generated first. The command line parameter **-createkeyfile** will generate such a file named `information.lic`.

Note: On **MS Windows** all the following commands starting with `./blast2go_cli.run` must be replaced with `blast2go_cli.exe`.

Steps to activate the Blast2GO Command Line:

1. Generate the **information.lic** file by executing the following command:

```
./blast2go_cli.run -createkeyfile
```
2. Go to www.blast2go.com/cli-activation and upload **information.lic**, provide your Activation Key and your Email.
3. The Blast2GO support team will create a **license.lic** file which has to be placed in the same folder as the `blast2go_cli` executable.
4. You can check the details of a license file with the option **-showlicenseinfo**

You can now continue with the GO Mapping database installation(See section 2.4) and have a look at the example use cases (See section 4).

2.3 Create a Properties File

The Blast2GO Command Line needs a properties file, that contains all the information of the different parameters that can be changed for the analysis. The properties file can be created with this command:

```
./blast2go_cli.run -createproperties cli.prop
```

Once this file has been created it is possible to edit it with a text editor. An editor with syntax highlighting is recommended, since it allows to better distinguish between comments and parameters.

2.4 Setting up a local Blast2GO GO Database

Blast2GO CLI offers the possibility to install a Blast2GO database semi-automatically with the command **-creategodb**.

Important: Before running this command please check the following requirements which will also be described more in detail below:

1. **A working MySQL database installation**
2. **Available disk space** (approx. 210GB during the installation process)
3. **Obtain all necessary and up-to-date data files**
4. **Configure the cli.prop file for your setup** (db-version, db-user, etc)

MySQL database installation

A MySQL server, local or accessible over network is necessary. Furthermore the credentials for a user with sufficient permissions on the database server are needed and will be prompted in the terminal window. Since the command line also creates a Blast2GO default user to access the database, it is mandatory to provide a user with **grant privileges** (e.g. the root user, ALL is not sufficient!).

If you are not allowed or willing to provide the root password, you can create the database manually (The database name must match dbname provided in the properties).

It is also necessary to create the database user which can INSERT and SELECT on this database.

Available disk space

During the installation the program will download and extract various files, which occupy (September 2016) approx. 65GB (Choose file location with **-tempfolder** parameter). The final database installation needs approx 150GB of available disk space in the MySQL servers data folder (usually /var/lib/mysql on the server machine). The software itself does not check for sufficient disk space, so if the database installation seems to hang, please check your available disk space on the server.

Obtain the necessary data files

Before starting the automatic download, please check that the file URLs provided in the properties file are up-to-date. Next, start the database installation from the Blast2GO Command Line. Obviously, for this option you need a working Internet connection. If you can not connect to the Internet from the workstation you are running the command line you can download the necessary files directly from the indicated URLs and copy them to your temp directory (**-tempfolder**). You can now specify the path to these files in the properties file and start the database installation.

Configure the properties file for your setup

Before creating the Blast2GO database you need to configure your **cli.prop** file. The following parameters will be used during the installation.

```
Dbaces.dbname=b2g_sep15
Dbaces.dbhost=192.168.0.1 // also possible with a port 192.168.0.1:3306
Dbaces.dbuser=blast2go
Dbaces.dbpasswd=blast4it
```

The information provided in these 4 lines is used to create the Blast2GO database and to use it later on for the GO Mapping process. It is recommended to name the database according to the version of the **assocdb-data** file like in our example above (b2g_MMMyy).

The next section defines all files we need to import into the database:

```
Dbaces.assocbdbdata=http://archive.geneontology.org/latest-full/go_monthly-assocdb-data.gz
Dbaces.geneinfo=ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/gene_info.gz
Dbaces.gene2accession=ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/gene2accession.gz
Dbaces.idmapping=ftp://ftp.pir.georgetown.edu/databases/idmapping/idmapping.tb.gz
```

If you do not want to use online resources i.e. work offline (without URLs to third party websites) you can also point the URI to a local file e.g.:

```
Dbaces.assocbdbdata=file:///path/to/go_monthly-assocdb-data.gz
```

Note: The 3 slashes at the beginning of the URI are required.

Troubleshooting

The automatic installation is divided into 4 steps. Each step consists in downloading, unpacking and importing one file. If one step fails you can prevent redoing previous successful steps by indicating which steps you want to perform, e.g.:

1. You start the installation with

```
./blast2go_cli.run -properties cli.prop -creategodb
```

2. Step 1 completed successfully but the program fails in the 2nd step because of a network connection error.
3. You can now restart the installation starting with the 2nd step

```
./blast2go_cli.run -properties cli.prop -creategodb 2,3,4
```

Updating an existing database

To update an existing database we recommend to re-run the DB installation process with a new database name. Once the installation is complete, the database name can be changed in the corresponding properties files and the old database can be removed. E.g.: drop database b2g_MMMyy

3 Command Line Parameters

This section gives a quick guide on the parameters used in Blast2GO CLI. Some command examples will be given in the end of the detailed description.

1. Load or import data commands:

- loadannot** <path> Path to .annot file
- loadb2g** <path> Path to Blast2GO .b2g file
- loadblast** <path> Path to Blast .xml file (pre 2.2.31)
- loadblast31** <path> Path to Blast .xml/.json/.zip file (2.2.31+)
- loaddat** <path> Path to Blast2GO .dat file (legacy format - use loadb2g instead)
- loadfasta** <path> Path to fasta file. Activate -protein option when working with amino acids.
- loadips48** <path> Path to InterProScan 4.8 file or folder
- loadips50** <path> Path to InterProScan 5.0 file or folder

2. Analysis commands:

- annex** Run ANNEX to complement the Gene Ontology annotation based on existing molecular functions
- mapping** Run the Gene Ontology mapping
- annotation** Run the Blast2GO Annotation algorithm
- ecmapping** Map Annotated GOs to their Enzyme Codes (Included when -annotation is set)
- goslim** <path> Run goslim using an *.obo file. Possible subset obo files can either be downloaded from <http://geneontology.org/page/go-slim-and-subset-guide>, or customized by hand with OBO-Edit2.
- cloudblast** <cloudblastkey> Run CloudBlast via webservice. This requires a working internet connection and a valid CloudBlast key with a positive balance.
- cloudblastbalance** <cloudblastkey> Print the CloudBlast Computation Unit balance. This requires a working internet connection.
- extractfasta** <path> Extract features from a fasta reference to a fasta file (path). Needs configuration in the properties file.
- ips** <email> Run InterPro via webservice. This requires a working internet connection, a valid email address and that your data-set contains sequence data.
- localblast** <path> Run Blast against a local database. This requires a working internet connection in order to download the necessary Blast executable (Alternatively you can specify a binary folder manually and place the binary there). Also necessary is a correctly configured local blast database (properties file).

3. Save or export commands:

- saveannot** <path> Save the functional annotations (Gene Ontology terms and Enzymes) as .annot
- saveb2g** <path> Save the project as .b2g
- savedat** <path> Save the project as .dat (legacy format - use saveb2g instead)
- savelog** <path> Save the log in a specified file.
- savelorf** <path> Convert nucleotide sequences (FASTA format) into amino acid sequences (longest Open Reading Frame, FASTA format). This function may be used to prepare a FASTA file for a local InterProScan run.
- savereport** <path> Create .pdf report
- saveseqtable** <path> Save your data as it would be shown in the Blast2GO GUI version (tab separated)
- statistics** <charts> Provide a comma-separated list of desired statistical charts (try -statistics without options to get a list of all available charts). '-statistics all' will try to export all statistics that are available. The option -nameprefix will be ignored.
- exportgeneric** <path> Export sequence data in tabular format for post processing

4. Other Options:

- createproperties** <path> Path to where the default properties file should be created
- godbstat** Print a statistic about a GO database specified in the cli.prop. Useful to get a hint if the database installation has been successful.
- creategodb** <1,2,3,4> Create a Gene Ontology database on a specified MySQL server. All necessary files (4) will automatically be downloaded if not provided by the user. The corresponding URLs (uris) can be changed in the properties file. Please refer to the manual before.
- createkeyfile** Create a file which contains a unique ID for your computer. This file is necessary to issue license keys.
- help** Display this message
- nameprefix** <name> Prefix for any output files, if you do not specify any path for them (default: b2g_project)
- properties** <path> Path to properties file (mandatory)
- protein** Set this flag if the fasta file contains protein sequences. This option only makes sense together with the -loadfasta option.
- showlicenseinfo** Show details about the currently available license.
- tempfolder** <path> Path to temporary folder (default: Systems temp folder)
- useobo** <path> The obo file to use for annotation, some statistics and various file im- and exports. Download the latest version from http://data.biobam.com/b2g_res/obo_files/go_latest.obo.gz
- workspace** <path> Workspace folder, e.g. where the results will be saved if not specified (default: current folder)

Note:

If a path is specified for a save option (e.g. -saveannot), the options **workstation** and **nameprefix** will be ignored for this particular option (see Use Case Examples for detailed information).

3.1 Generic Export

The option **-exportgeneric** allows to export the data obtained for each sequence into a tabular text file. The resulting file contains one line per sequence and customizable columns. The corresponding settings in the properties file (**GenericExportParameters**) allow to decide how columns and items are separated and which data should be exported.

Column and item separators can be defined as: comma, semicolon, tabulator, whitespace or the pipe symbol (—). There are 54 different items available for each sequence. These items can be grouped as follows:

- General Sequence information such as the sequence name or its length.
seq_name, seq_desc, seq_data, seq_length
- Summarized Blast information like the total number of blast hits.
blast_hit_count, blast_min_eval, blast_sim_mean
- Specific information about each Blast hit such as its hit description, length, e-value or alignment length.
*blast_hits_desc, blast_hits_tax, blast_hits_eval, blast_hits_length, blast_hits_alignlength
blast_hits_pos, blast_hits_sim, blast_hits_hsphit, blast_hits_query, blast_hits_hspcount
blast_hits_frame, blast_hits_geneid, blast_hits_acc, blast_hits_score*
- Specific information about the top blast hit (highest bit score).
*blast_tophit_desc, blast_tophit_tax, blast_tophit_eval, blast_tophit_length
blast_tophit_alignlength, blast_tophit_pos, blast_tophit_sim, blast_tophit_hsphit
blast_tophit_query, blast_tophit_hspcount, blast_tophit_frame, blast_tophit_geneid
blast_tophit_acc, blast_tophit_score*
- All GO Mapping candidate GO terms and evidence codes.
*mapping_genename, mapping_tax, mapping_xref, mapping_xrefdb
mapping_goid, mapping_goname, mapping_gocategory*
- Annotated GO terms and enzyme codes.
annot_count, annot_goid, annot_goterm, annot_gocategory, enzyme_code, enzyme_name
- Detailed information about InterPro results like obtained domains, families as well as GO terms.
ips_acc, ips_type, ips_name, ips_sig, ips_goid, ips_goname, ips_gocategory

4 Use Case Examples

This section provides several example use cases for the Blast2GO Command Line. Please read the Setup chapter carefully and configure the GO Mapping database in your properties file.

Important things to consider:

- If you are using MS Windows all commands must be changed accordingly. Please replace `./blast2go_cli.run` with `blast2go_cli.exe`
- A properties file is always necessary, create it with:

```
./blast2go_cli.run -createproperties cli.prop
```

- GO Annotation, Enzyme Code Mapping, Statistics, GO Slim and various import and export functions make use of the obo file (-useobo). The CLI contains a default obo file. However, we recommend that to download the up-to-date version of the obo file. Note: The obo file should be from the same month as the GO Mapping database. http://data.biobam.com/b2g_res/obo_files The latest version can be found here: http://data.biobam.com/b2g_res/obo_files/go_latest.obo.gz Just provide this file additionally when executing a command:

```
./blast2go.run -useobo go_latest.obo.gz -properties cli.prop -annotation ...
```

- A public GO Mapping server is available for testing purpose. This database is not for production use and from 2013! To use this server for testing, please configure your properties file as follows:

```
// ** DataAccessParameters **
DataAccessParameters.ownDBName=b2g_sep13
DataAccessParameters.ownDBHost=publicdb.blast2go.com
DataAccessParameters.ownDBUser=blast2go
DataAccessParameters.ownDBPassword=blast4it
```

4.1 Examples

1. Load a DNA fasta file, add the corresponding BLAST results and perform GO Mapping and Annotation. Furthermore, we want to save the .b2g file and the PDF report at the current directory with the name "example".

```
./blast2go_cli.run -properties cli.prop -loadfasta \  
example_data/1000_plant.fasta -loadblast example_data/1000_plant_blastResult.xml \  
-mapping -annotation -saveb2g example.b2g -savereport example.pdf
```

2. This example requires a local Swissprot Database installation. Simply download and extract the file from: ftp.ncbi.nlm.nih.gov/blast/db/swissprot.tar.gz
Load nucleotide sequences, run local BLAST against the Swissprot database, GO Mapping and Annotation. We also create various statistics. Finally the whole project will be saved to the example_data folder in .b2g format with the chosen name prefix together with the log file. The LocalBlastAlgoParameters have to be configured:

```
// ** LocalBlastAlgoParameters **
LocalBlastAlgoParameters.blastProgram=blastx-fast
LocalBlastAlgoParameters.blastDbFile=/path/to/swissprot.pal
LocalBlastAlgoParameters.blastXML2ResultEnable=true
LocalBlastAlgoParameters.blastXML2Result=example_data/blast_xmls
```

Command Line:

```
./blast2go_cli.run -properties cli.prop -loadfasta example_data/15_plant.fasta \
-workspace example_data -nameprefix localblastSwissprot \
-localblast <path_to_blastx_executables> -mapping -annotation -statistics \
bspecdis,mbresmap,aannotscore -saveb2g -save log example_data/blast2go.log
```

3. Load nucleotide sequences, import BLAST results (.json or .xml2) from a zip file, run GO Mapping and Annotation. Save the whole project and its report as example.json.b2g.

```
./blast2go_cli.run -properties cli.prop -loadfasta example_data/15_plant.fasta \
-loadblast31 example_data/json/02X9PD4T01R-Alignment.json.zip -mapping \
-annotation -savereport example_data/example_json_report.pdf -saveb2g \
example_data/example_json.b2g
```

4. Load example.b2g from the second example and run InterProScan (online). We will save the project, as well as the InterProScan results.
The following InterProScanAlgoParameters have to be configured:

```
// ** InterProScanAlgoParameters **
InterProScanAlgoParameters.ipsXML2Result=example_data/ips_xmls
InterProScanAlgoParameters.ipsXML2ResultEnabled=true
```

Command Line:

```
./blast2go_cli.run -properties cli.prop -loadb2g example_data/example.b2g \
-ips <valid_email_address> -saveb2g example_data/example_withIPS.b2g
```

5. Convert sequences to proteins and save them as fasta file.

```
./blast2go_cli.run -properties cli.prop -useobo go_latest.obo -loadfasta \
example_data/15_plant.fasta -savelorf example_data/15_plant_protein
```

6. Load a .b2g file, apply plants GO Slim and save the results as .b2g, which will be saved with the default nameprefix "b2g.project" into the current directory.

```
./blast2go_cli.run -properties cli.prop -useobo go_latest.obo -loadb2g \
example.b2g -goslim example_data/goslim_plant.obo -saveb2g
```

- To run this example, we need the results from the previous example. Run CloudBlast, GO Mapping, Annotation on the protein sequences and save the results as .b2g and customized annotation format.

Please configure the following properties sections:

```
// ** CloudBlastAlgoParameters **
CloudBlastAlgoParameters.blastProgram=blastp-fast
CloudBlastAlgoParameters.blastDB=nr_alias_viridiplantae
CloudBlastAlgoParameters.blastXML2ResultEnable=true
CloudBlastAlgoParameters.blastXML2Result=example_data/blast_xmls
```

```
// ** ExportAnnotParameters **
ExportAnnotParameters.format=custom
ExportAnnotParameters.desc=true
ExportAnnotParameters.go=category_and_id_and_term
ExportAnnotParameters.goseparator=tabulator
ExportAnnotParameters.column=tabulator
ExportAnnotParameters.row=sequence
```

Command Line:

```
./blast2go_cli.run -properties cli.prop -loadfasta \
example_data/15_plant_protein.fasta -protein -cloudblast B2G-CloudBlastKey \
-mapping -annotation -saveb2g example_data/15_plant_protein.b2g \
-saveannot example_data/15_plant_annotation.txt
```

- Load a protein fasta file, add the corresponding BLAST results and execute GO Mapping and Annotation. All files (.b2g, .pdf, .annot and .txt) will be saved with the nameprefix “p53” in the “work_dir” in the current directory. Additionally, the data distribution pie chart and enzyme statistics will also be saved in the “work_dir” folder.

Command Line:

```
./blast2go_cli.run -properties cli.prop -loadfasta \
example_data/1000_seq_protein.fasta -protein -loadblast \
example_data/1000_plant_protein_blastResult.xml -mapping -annotation \
-workspace work_dir -nameprefix p53 -saveb2g -saveannot -savereport \
-saveseqtable -statistics gdatadispie,aecdis
```

- Load a fasta file, a BLAST result file and InterProScan 5.0 files, perform GO mapping, annotation and ANNEX. Then create all statistical charts. As a result we want to obtain the .b2g and the PDF report, which will be saved with the default nameprefix “b2g-project” into the current directory.

```
./blast2go_cli.run -properties cli.prop -loadfasta example_data/1000_plant.fasta \
-loadblast example_data/1000_plant_blastResult.xml -loadips50 \
example_data/1000_seq_protein_ips50.xml -mapping -annotation -annex \
-statistics all -saveb2g -savereport
```

- Load an example data-set and export user defined columns for each sequence.

Please configure the following properties section:

```
// ** GenericExportParameters **
GenericExportParameters.columnSeparator=tabulator

GenericExportParameters.itemSeparator=semicolon

GenericExportParameters.itemsToExport=seq_name,blast_hit_count, \
mapping_genename,mapping_xref,mapping_goid,annot_goid,enzyme_code
```

```
./blast2go_cli.run -loadb2g example_data/example.b2g \
-exportgeneric example_data/blast_top_hit.txt
```

Bibliography

- [Conesa et al., 2005] Conesa, A., Götz, S., García-Gómez, J. M., Terol, J., Talón, M., and Robles, M. (2005). Blast2go: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics*, 21(18):3674–3676.
- [Götz et al., 2008] Götz, S., Garcia-Gomez, J. M., Terol, J., Williams, T. D., Nagaraj, S. H., Nueda, M. J., Robles, M., Talon, M., Dopazo, J., and Conesa, A. (2008). High-throughput functional annotation and data mining with the blast2go suite. *Nucl. Acids Res.*, pages gkn176+.