Genevestigator Export Tools

Revision: 2.4 (07/12/2020)

1 Export Tools

1.1 Installation

The Genevestigator Export Tools consists of a set of shell scripts running under Unix/Linux and a small Java program to export the data from the Genevestigator server. The prerequisites are:

1. Java version 11 or higher
2. Bash Shell
3. the GNU `tar` archiving utility

To install the scripts and Java program proceed as follows:

1. Change directory to your installation folder and unpack the archive:
   `tar -xzf DataExportTool.tar.gz`
2. Set the execution permission on all Shell scripts:
   `chmod ug+x *.sh DataExportTool/bin/*.sh`
3. Optionally, change the group of the installation folder and its files if the scripts should be accessible by different users:
   `chgrp -R <group_name> <installation_folder>`
4. Change the environment settings in the file `DataExportTool/bin/setenv.sh`. Set `JAVA_HOME` pointing to the installation folder of a Java 11 version and configure the `SERVICE_URL`, `USERNAME` and `PASSWORD` variables pointing to the services of the Genevestigator server with a user (and password) having the permission to use the DataExport API. For instance, your `setenv.sh` file might look like:

   ```
   ...
   # java settings
   JAVA_OPTS="-Xmx4g"
   JAVA_HOME=/usr/java/openjdk-11
   # server
   SERVICE_URL="https://<hostname>:443/Server_V4/dispatch/";
   USERNAME="export";
   PASSWORD="<password>";
   ...
   ```

5. In your home directory, create the folder `GVLogs` where all scripts will save their logfiles:
   `mkdir $HOME/GVLogs`
6. Test the installation by exporting all platforms from Genevestigator:
   `<installationfolder>/exporttool.sh platforms platforms.csv`
   This will echo the commandline arguments and export all available platforms of the configured Genevestigator server into the textfile `platforms.csv`. Any error will be reported into the logfile `exporttool.log` in the `GVLogs` folder just created.

The Export Tools package contains several files and directories, among which the most important are:

Name	Description
`exporttool.sh`	Script to export (mostly) tree-structured experiment data
`experimentexport.sh`	Script to export experiment data in tabular form (e.g. GEO)
`computetool.sh`	Script to compute meta-profiles for anatomy, development, cell line, neoplasm, perturbations on stimulus, mutations or perform operations on expression matrices
`dataexport.sh`	Script to export all files for a platform
`export_wrapper.R`	R-Wrapper to the data export scripts (to be called from R)
`README`	Latest release information and short description of script usage
`bin/setenv.sh`	Environment settings for all Shell scripts
`doc/ExportFormats.pdf`	Definition of the export formats
`doc/ExportAPI.pdf`	Description of the Java Export API
`doc/ExportTools.pdf`	This document
`doc/ExportTools_R-wrapper.pdf`	Description of the R-wrapper
`javadoc/`	Java-style documentation of the Export API and Tools
`lib/`	Contains `.jar` files of the Export Tools and its dependencies
`src/`	Contains the Java sources of `DataExportAPI_V2.jar` and of `DataExportTool_V2.jar`

1.2 Usage

1.2.1 exporttool.sh

Synopsis

`exporttool.sh␣platforms␣`*destfile*

Description Exports the available platform descriptions from the server configured in bin\setenv.sh into specified destination file. Note that the exported data can be restricted by permissions.

Synopsis

`exporttool.sh␣(anatomytree|celltypetree|celllinetree|neoplasmtree|developmenttree|propertytree|stimulustree|mutationtree)␣`*destfile* `(-t␣`*platform*`␣|␣-e␣(`*experiment numbers*`)+␣|-f␣`*experiment numbers file*`)␣[-n]`

Description Exports the relationships between ontologies, samples and comparisons into the specified destination file. To export all annotations including properties use the commands of the script experimentexport.sh as defined in 1.2.4

Options

- anatomytree Exports the Anatomy-Chip relations.
- celltypetree Exports the Cell type-Chip relations.
- celllinetree Exports the Cell line-Chip relations.
- neoplasmtree Exports the Neoplasm-Chip relations.
- developmenttree Exports the Developmentstate-Chip relations.
- propertytree Exports the Property-Chip relations.
- stimulustree Exports the Stimulus-Control/Treatment Group-Chip relations.
- mutationtree Exports the Mutation-Control/Treatment Group-Chip relations.
- -t platform Exports the data for specified platform.
- -e experiment numbers Exports the data for specified list of experiment numbers.

-f file Exports the data for experiment numbers listed in specified file. Each line contains one experiment number.

-n Export the notes/description for the tree nodes.

Synopsis

`exporttool.sh␣(experimenttree|experiments)␣`*destfile*`␣(-t␣`*platform*`␣|␣-e␣(`*experiment numbers*`)+␣|␣-f␣`*experiment numbers file*`)`

Description Exports the experiment-replica-chip relationship and the description of the experiment into specified destination file.

Options

experimenttree Exports Experiment-Replica-Chip relationship.

experiments Exports the experiment description.

-t platform Exports the data for specified platform.

-e experiment numbers Exports the data for specified list of experiment numbers.

-f file Exports the data for experiment numbers listed in specified file. Each line contains one experiment number.

Synopsis

`exporttool.sh␣(applicationarea|globalstudytype|studydesign)␣`*destfile*`␣(-o␣`*organism*`␣|␣-t␣`*platform*`␣|␣-e␣(`*experiment numbers*`)+␣|␣-f␣`*experiment numbers file*`)␣[-n]`

Description Exports the experiment-level annotation into the specified destinaton file.

Options

applicationarea Exports the application area experiment ontology.

globalstudytype Exports the global study type experiment ontology.

studydesign Exports the study design experiment ontology.

-o Exports the experiment ontology only for experiments of specified organism. Without any option the experiment ontology for all experiments across all organisms will be exported.

-t platform Exports the experiment ontology for specified platform.

-e experiment numbers Exports the experiment ontology for specified list of experiment numbers.

-f file Exports the experiment ontology for experiment numbers listed in specified file. Each line contains one experiment number.

-n Export the notes/description of the ontology tree nodes.

Synopsis

`exporttool.sh␣expressionmatrix␣`*destfile*`␣(-t␣`*platform*`␣|␣-e␣(`*experiment numbers*`)+␣|␣-f␣`*experiment numbers file*`)␣[-p␣(`*transcript*`)+]␣[--datatype␣`*datatype*`]␣[-s]`

Description Exports the expression matrix into specified destination file.

Options

-t platform Exports the expression matrix for specified platform

-e experiment numbers Exports the expression matrix for specified list of experiment numbers.

-f file Exports the expression matrix for experiment numbers listed in specified file. Each line contains one experiment number.

-p (transcript)+ Extracts only the signal value for given transcripts.

-datatype datatype Extracts the specified datatype. By default the log2 expression values of the given platform will be exported. Valid datatypes are RMA, RMA_LOG, TPM, TPM_LOG, EXPECTED_COUNT

-s Export floating point values in scientific notation.

Synopsis

`exporttool.sh␣ortholog␣`*destfile*`␣[-o␣`*organism*`␣|␣--organisms␣(`*organism*`)+]␣[-f␣`*optionfile*`]␣[-g␣`*type*`]␣[-s]`

Description Exports the ortholgous genes and its measures including the sequence score (PAM) and the computed expression score into specified destination file. Note: If no further option is specified all orthologous genes for all available organisms will be exported.

Options

-o Exports only orthologous genes from all other organisms to specified organism.

-organisms Exports only orthologous genes available between list of specified organisms.

-f optionfile Exports the orthologs genes for given organisms, namingschemes and platforms specified in the file. Each line of the given csv-file must specify an organism.short_caption, namingscheme.caption and platform.short_caption used for the export.

-g type Exports the expression value for given genetic element: Gene, Protein, Transcript or Probeset. Default: Gene. Note that the supported type depends on the specified platform.

-s Export floating point values in scientific notation.

Synopsis

`exporttool.sh␣diffexpression␣(-e␣`*experiment number*`␣-c␣(`*comparison name*`|all)␣|␣-f␣`*comparisonfile*`␣[--allowCrossComparisons])␣--maxFDR␣`*float*
`␣--output-directory␣`*directory*`[-g␣`*type*`]␣`
`␣[-m␣`*namingschemes*`]␣[--minLogRatio␣`*float*`]␣[--maxLogRatio␣`*float*`]␣[-s]`

Description Computes and exports the differential expressed genes for predefined or custom comparison(s) of an experiment into files.

Options

-e experiment number Experiment number for which the comparison will be selected.

-c comparison name Name of a predefined comparison or "all" if the differential expression should be calculated for all comparisons of the specified experiment.

-f file Name of a comparison file containing exported or custom comparisons for which the differential expression should be computed. The samples of each comparison must belong to the same experiment unless the option `--allowCrossComparisons` is provided. All comparisons resp. experiments must belong to the same organism.

–allowCrossComparisons Allows the definition of comparisons with samples from different experiments. All samples in a comparison must belong to the same platform.

–maxFDR Maximal false discovery rate for the exported differential expressed genes.

-m Exports additionally the aliases of the genes for the specified namingschemes.

–minLogRatio Filters differential expressed genes below given threshold.

–maxLogRatio Filters differential expressed genes above given threshold.

–output-directory Export directory for the resulting files. For each comparison a file named `diffexpression_<nr>_<experiment.nbr>.csv` will be created.

-g type Use the expression value for given genetic element: Gene, Protein, Transcript or Probeset to compute the differential expression. Default: Gene. Note that the supported type depends on the specified platform of the experiment.

-s Export floating point values in scientific notation.

Exit codes

0 Command computed the differential expression for all specified comparisons and terminated successfully.

1 Command failed to compute the differential expression for all specified comparisons.

2 For multiple comparisons: the command successfully computed the differential expression for some of the comparisons but failed for others. The command will either log a warning if a comparison does not meet the condition to compute the differential expression (eg. at least 2 samples for each group of a comparison) or an exception if the command failed to compute differential expression (eg. server error).

3 Command was called with an unknown or illegal options.

1.2.2 dataexport.sh

Synopsis `dataexport.sh` *directory* *platform* `[-e (`*experiment* `)+]`

Description Exports all data files for all microarrays for given platform (platform.short_caption) from the server configured in bin\setenv.sh into given directory. Note that the exported data can be restricted by experiments.

Options

-e eperiment numbers List of experiments (experiment.nbr) separated by space. Only chips for given experiments will be retrieved.

1.2.3 computetool.sh

Synopsis
`computetool.sh anatomy` *destfile anatomytreefile expressionmatrixfile* `[-s]`
`computetool.sh celltype` *destfile celltypetreefile expressionmatrixfile* `[-s]`
`computetool.sh cellline` *destfile celllinetreefile expressionmatrixfile* `[-s]`
`computetool.sh neoplasm` *destfile neoplasmtreefile expressionmatrixfile* `[-s]`
`computetool.sh development` *destfile developmenttreefile expressionmatrixfile* `[-s]`
`computetool.sh stimulus` *destcontrolfile desttreatmentfile stimulustreefile expressionmatrixfile* `[-s]`
`computetool.sh mutation` *destcontrolfile desttreatmentfile mutationtreefile expressionmatrixfile* `[-s]`
`computetool.sh difference` *destfile expressionmatrixfile1 expressionmatrixfile2* `[-s]`
`computetool.sh ratio` *destfile expressionmatrixfile1 expressionmatrixfile2* `[-s]`

Description A tool to perform calculation on measure/signal matrix files and to compute the averaged expression matrix file for a given anatomy, stimulus, mutation, experiment or development data file and measure/chip signal matrix file.

Commands

anatomy Computes the averaged expression matrix file for each anatomical category of the anatomytreefile with given expressionmatrix file and saves the result in the new destination file.

celltype Computes the averaged expression matrix file for each celltype category of the celltypetreefile with given expressionmatrix file and saves the result in the new destination file.

cellline Computes the averaged expression matrix file for each cell line of the celllinetreefile with given expressionmatrix file and saves the result in the new destination file.

neoplasm Computes the averaged expression matrix file for each neoplasm category of the neoplasmtreefile with given expressionmatrix file and saves the result in the new destination file.

development Computes the averaged expression matrix file for each developmental state of the developmenttreefile with given expressionmatrixfile and saves the result in the new destination file.

stimulus Computes the averaged expression matrix file for the control set and treatment set of each stimuli of the stimulustreefile and saves the result into the destination files destcontrolfile desttreatmentfile.

mutation Computes the averaged expression matrix file for the control set and treatment set of each mutation of the mutationtreefile and saves the result into the files destcontrolfile desttreatmentfile.

difference Calculates the difference of two expression matrices expressionmatrixfile1 and expressionmatrixfile2 and saves the result into the new destination file.

ratio Calculates the ratio of two expression matrices expressionmatrixfile1 and expressionmatrixfile2 and saves the result in the new destination file.

-s Export floating point values in scientific notation.

1.2.4 experimentexport.sh

Synopsis

experimentexport.sh␣geoseries␣*destfile*␣(-t␣*platform*␣|␣-e␣(*experiment numbers*)+
␣|␣-f␣*experiment numbers file*)␣[-g␣*type*]␣[-y]␣[-s]␣[-x]

experimentexport.sh␣comparison␣*destfile*␣(-t␣*platform*␣|␣-e␣(*experiment numbers*)+
␣|␣-f␣*experiment numbers file*)␣[-y]␣[-x]

experimentexport.sh␣signal␣*destfile*␣(-t␣*platform*␣|␣-e␣(*experiment numbers*)+
␣|␣-f␣*experiment numbers file*)␣[-g␣*type*]␣[-m␣(*namingscheme.caption*)+␣]
␣[-b]␣[-s]␣[-d␣*datatype*␣]␣[-x]

experimentexport.sh␣annotation␣*destfile*␣(-t␣*platform*␣|␣-e␣(*experiment numbers*)+
␣|␣-f␣*experiment numbers file*)␣[-y]␣[-r]␣[--cluster_biosample]␣[--cell]␣[-x]

Description A tool to export experiment data in a tabular form. In contrast to the other export formats, these formats are somewhat redudant, but have the advantage that annotations are represented in a self-contained manner. As such, they are well-suited to export individual experiments to view them e.g. in Excel.

Commands

geoseries Exports the given experiment data (annotations and signal) in the Geo Matrix Series format. The signal data is exported as linear values.
If multiple experiments are given the specified <destfile> is used as prefix to create separate files with the naming pattern <destfile>_<experiment.nbr>.<extension>.

comparison Exports the comparisons (perturbations) of the given experiment(s). For each chip, it is indicated in which perturbations it occurs in a treatment or in a control group.
If the given experiments belong to different platforms a separate file per platform is created with the naming pattern <destfile>_<platform.short_caption>.<extension>.

signal Exports the signal data for the given experiment(s), as linear values. The main difference to the `expressionmatrix` command of the `exporttool.sh` script is that the chip columns are labeled by experiment number and the chip's repository name and the data is exported in linear values. Moreover, the -m option allows to label rows by aliases from the given namingschemes in addition to the measure identifier.
If the given experiments belong to different platforms a separate file per platform is created with the naming pattern <destfile>_<platform.short_caption>.<extension>.

annotation Exports the annotation data for the given experiment(s).
If the given experiments belong to different platforms a separate file per platform is created with the naming pattern <destfile>_<platform.short_caption>.<extension>.

Options

-t Exports the data for all experiments of specified platform

-e Exports the data for specified experiment number(s). By default experiment number(s) are Genevestigator specific experiment id(s) (e.g. HS-01202) unless option -x is selected.

-f file Exports the data for experiment numbers listed in specified file. Each line contains one experiment number.

-x Specified experiment number(s) are id(s) from other repositories (e.g. GSE20680)

-g Exports the signal values for measurements of given type: Gene, Probeset, Protein or Transcript. Default: Gene. Note that the supported type depends on the specified platform of the experiments.

-m Exports the aliases for the specified namingschemes.

-r Exports the comparisons/perturbations: For each chip, it is indicated in which perturbations it occurs in a treatment or in a control group.

-y Exports the hierarchy number (a numbering of ontology nodes) for ontology categories.

-b Exports only the best measures for specified type in option -g. If absent all measures for the platform will be exported.

-s Export floating point values in scientific notation.

-d Export values for the specified data type.

–cluster_biosample Only supported on aggregate platforms: Exports the cluster and biosample names of aggregates.

–cell Only supported on aggregate platforms: Exports the cell names of clusters of aggregates. For each cluster all cells will be exported along with its annotation.