**Table of Contents:**

[1 Goal of the documentation 2](#_Toc94180219)

[2 Scope and Responsibilities 2](#_Toc94180220)

[3 Term/Definition/Abbreviation 2](#_Toc94180221)

[4 Procedure/ Workflow 2](#_Toc94180222)

[4.1 Software 2](#_Toc94180223)

[4.2 Security details 2](#_Toc94180224)

[4.3 Method procedure 3](#_Toc94180225)

[4.3.1 General Notes 3](#_Toc94180226)

[4.3.2 Downloading WormSource 3](#_Toc94180227)

[4.3.3 Finding new GEO datasets 4](#_Toc94180228)

[4.3.4 Transcriptomics File 5](#_Toc94180229)

[4.3.5 Assembling dataset 6](#_Toc94180230)

[4.3.6 Manual Data Analysis 7](#_Toc94180231)

[4.3.7 WormBase ID 7](#_Toc94180232)

[4.3.8 Fusion of Files 7](#_Toc94180233)

[4.3.9 TomCat pilot run 7](#_Toc94180234)

[4.3.10 Updating database 7](#_Toc94180235)

[5 Document History 7](#_Toc94180236)

[6 Publication bibliography 7](#_Toc94180237)

**Further documents**:

(evogen 2015) Manual for WormExp

(Yang et al. 2016) WormExp :WormExp: a web-based application for a Caenorhabditis elegans-specific gene expression enrichment analysis

Website <https://wormexp.zoologie.uni-kiel.de/wormexp/>

(Dozmorov 2016) GEOparse documentation

datafinder jupyter notebook file

1. **Goal of the documentation**

This documentation describes the procedure to update the web-based application WormExp.

1. **Scope and Responsibilities**

This documentation is only valid for AG Schulenburg.

|  |  |
| --- | --- |
| **Function** | **Responsibilities** |
| Employee/User | Updating WormExp  Is responsible for the correct use and update of the application. |

1. **Term/Definition/Abbreviation**

|  |  |
| --- | --- |
| GEO | Gene Expression Omnibus |

1. **Procedure/** **Workflow**

## Software

|  |  |  |  |
| --- | --- | --- | --- |
| Software/Website | Specification | Version | Source/Link |
| WormExp | web-based application for a taxon-specific gene set exploration and enrichment analysis | WormExp v1.0 | https://wormexp.zoologie.uni-kiel.de/wormexp/ |
| Python | Programming language | 3.8.11 | https://www.python.org/ |
| Anaconda | Python package distribution and management | 2020.11 | https://www.anaconda.com/ |
| GEOparse | Python library to access Gene Expression Omnibus Database (GEO) | 2.0.3 | https://geoparse.readthedocs.io/en/latest/GEOparse.html |
| Jupyter Notebook | Web-based environment for working with notebooks | 6.4.0 | https://jupyter-notebook.readthedocs.io/en/stable/index.html |
| Matplotlib | Python data visualization tool | 3.4.2 | https://matplotlib.org/ |
| Numpy | Core package for scientific computing with Python. | 1.20.3 | https://numpy.org/ |
| Pandas | Library for tabular data structures | 1.3.2 | https://pandas.pydata.org/docs/index.html# |
| Biopython | Library for biological computation written in Python | 1.78 | https://biopython.org/ |
| Java DK | The JDK is a development environment for building applications using the Java programming language. | 17.0 | https://www.oracle.com/java/ |
| Apache TomCat | Apache Tomcat software powers numerous large-scale, mission-critical web applications. | 8.5.70 | https://tomcat.apache.org/ |

## Security details

N/A

## Method procedure

### General Notes

This documentation describes the general procedure on how to update the database behind WormExp. The details in the procedure can be modified depending on the user’s operation system, experience, etc. The scripts and procedures used here were based on a Python (version 3.8.11) script in a virtual environment managed by anaconda (version 2020.11). The virtual environment contained several additional libraries: GEOparse (version 2.0.3), Jupyter (version 6.4.0), Matplotlib (version 3.4.2), Numpy (version 1.20.3), Pandas (version 1.3.2) and Biopython (version 1.78).

These libraries are essential for a correct function of the provided script to find new GEO datasets uploaded into Pubmed GEO. However, GEOparse is also available for R and the whole procedure can therefore be transferred to R if wished.

The data should be collected in a separate folder and new Excel file, and only merged with the existing database at the very end. Additionally, a pilot run should be included on a copy of the current database with new gene sets to make sure that everything works appropriately. The database will be tested locally via apache tomcat, but other software to locally host servers are applicable.

### GitHub

Description of GitHub repo + WormSource folder and everthing

The file contains a folder “WormSource”, which holds the complete database information. It contains another folder WormExp, which holds the java source files to run it. Furthermore, it contains several .txt files:

|  |  |
| --- | --- |
| File | Description |
| c\_elegans.WS283.geneIDs | contains Wormbase ID, gene name and gene ID for every gene |
| reference | contains dataset name and link to publication |
| Chemicalexposure-otherStress | contains datasets categorized to Chemicals/Stress |
| DAF Insulin food | contains datasets categorized to DAF/Insulin/food |
| Development-Dauer-Aging | contains datasets categorized to Development/Dauer/Aging |
| Kim Mounts | contains datasets categorized to Kim Mountains |
| Mutants | contains datasets categorized to Mutants |
| Other | contains datasets categorized to Other |
| Pathogen | contains datasets categorized to Microbes |
| Targets | contains datasets categorized to TF Targets |
| Tissue-specific | contains datasets categorized to Tissue |
| Epigenetics | contains datasets categorized to Tissue |
| WormExp\_info | has key information about the datasets, like number of genes, links to publications and methodology |
| ~~datafinder.ipynb~~ | ~~jupyter notebook that contains script and instruction to use GEOparse to find new datasets~~ |
| Wormbase\_version\_changes | contains information about changes in WormBase IDs the last years |

### Finding new GEO datasets

To find new GEO datasets, the jupyter notebook “datafinder” can be used. It contains scripts and instructions on how to use it. The script uses the API GEOparse and searches for datasets depending on the inserted query. During the start of this project an analysis was conducted to ensure the used query contains the correct datasets. The results are documented in the supporting pdf document and given to the project owner.

Adaptions to the query should only be made in respect to the publication date. As of 02/2022 the database contains datasets until 2018.

The script will create a separate Excel file called “GSE\_query\_results” in which further (manual) work will be conducted. This Excel file is the backbone of all further investigation and will be described in detail in the following section “Transcriptomics File”. It must be noted that in order to use the datafinder notebook, the software from section 4.1 has to be installed.

### Transcriptomics File

The datafinder script produces an Excel file that contains detailed information about every dataset found with GEOparse. It shows not only the exact title of the dataset, but also its geo\_accession number, publication date, contributors, etc. etc. See Figure 2 for an example.

Calendar

Description automatically generated with low confidence

Figure 2: Exempt from original Transcriptomics file

The main task here is to start sorting through the found datasets and finding out which datasets are useful for the updates. It has been decided to concentrate mostly on datasets that already possess an in-depth transcriptomics analysis done by the respective scientists. Datasets that only possess raw data can be ignored until otherwise stated. For a better overview, the transcriptomics file was transformed, and colors have been introduced (see Figure 3) to show which data has supplementary data available (blue), which datasets contain only raw data (yellow), and which datasets can be excluded (red). Exclusion of datasets was mainly due to no available paper or dataset was not focused on differential gene expression.

Table

Description automatically generated

Figure 3: Excerpt from the Transcriptomics file, after initial preparation

Additionally, new columns have been introduced. One of the main tasks in updating the database is to sort found datasets to categories mentioned in 4.3.2. If a dataset can be sorted to more than one category, the respective row is copied and added directly underneath (see section 4.3.5 for more detailed information about categories). Furthermore, columns “Category\_check”, “Dataset\_check”, and “Comment” were added for supervision. Those categories will only be switched to “done” when categories and assembled dataset (see 4.3.6) were checked by supervisors and no problems occurred.

### Categorizing datasets

Datasets are categorized according to the scientific research topic of interest. The choice should always be validated by a supervisor, but general rules are the following:

*Kim Mountains* is a specific category reserved for results from Kim et al., 2001. In the category *Mutants* order all differentially expressed genes that show up in mutants or upon RNA interference-silencing of a particular gene. Datasets that show exposure or feeding of various microorganism are categorized *Microbes*. *TF Targets* is for transcription factor targets inferred by knock-down/knock-out of the respective transcription factors. The category *Tissue* is for gene expressions in specific tissues. *Development/Dauer/Aging* includes differential expression in the various developmental stages and during aging. *DAF/Insulin/food* has differential expression in response to food, starvation, or insulin-like receptor activation/de-activation. The category *Chemicals/stress* incorporates exposure to chemical compounds or other stressors and *Other* includes all gene sets not categorized. A new category *Epigenetics* has been added. It includes all gene sets that came from chromatin studies or epigenetic markers. If a data set fits more than one category, it will be added to all of them.

### Assembling data sets

Next to the transcriptomics file, another file must be updated for the database. For the following dataset assembly, the main work will be conducted in the WormExp\_info file.

Chart, scatter chart

Description automatically generated

Figure 4: Excerpt from WormExp\_info file

All new and added datasets must be added in the same manner as can be seen in the existing file. The columns are described in detail in the following table:

|  |  |
| --- | --- |
| Column | Description |
| WormBaseVersion | contains information which WormBase Version was used to map Entrez IDs to gene IDs. If n.a. no information was given. More information in 4.3.7. |
| Categories | same as in transcriptomics file. If more than one category applies, dataset will be added for every category. |
| Gene Set name | Explanation for the assembled gene set. |
| number\_genes | number of genes collected for respective gene set |
| Refs | Reference to paper |
| Data From | gives information where the gene set was found in the paper |
| Selection\_criteria | Shows which selection criteria was applied when extracting the data set. More information in 4.3.6.1 |
| decided\_by | gives information if the selection\_criteria was given by the authors or if it was decided by the assembler |
| Rawdata | Information to GSE accession number |
| Additional | Column for additional information |
| Comment | Column for comments |

Assembling the dataset is the trickiest and most error-prone part in this work. This work cannot be streamlined, as every scientist analyzed their work differently and uploaded it in different places, and in various formats. However, most of the times a differential gene expression analysis is uploaded in a separate table and can be found in the supplementary of the respective paper. Depending on the authors, selection criteria are more or less strict. The assembler must decide in most cases which genes to extract. In 4.3.6.1 some guidelines for selection criteria have been decided.

For every GSE number an overall gene set name should be chosen that describes sufficiently the experiment conducted. Every gene set extracted from the respective experiment should be saved in a .txt. In this .txt file all gene sets belonging to this experiment should be saved. Assembled gene sets should not be added yet to the category files in WormSource! This fusion step will follow after every gene set has been checked over by supervision and no further changes are made.

#### Good Practice for Gene Set names

1. For mutations choose “in” instead of “by”
2. Chosen strains are in Upper Case (only if important for context)
3. Gene alleles in lower case -> only if different alleles were important in study
4. Add first authors in brackets if similar data set already exists (e.g. in mutants!)

#### Criteria for supplementary data filtering

As mentioned, every scientist employed their own significance selection criteria. If possible, selection criteria from the authors should be used. These criteria will be added in the column „selection\_criteria” and “decided\_by”.

Exceptions come into play if the selection criteria are not strict enough (e.g., p-value > 0.1 without any adjustments and without corrections). In general, p-fdr/padj < 0.01 and log-foldchange >= 2 (or <= -2) should be used.

### WormBase ID

WormBase IDs should only be mapped, when all categories and datasets have been checked by the supervisor. In this state of the project, it is recommended to make a copy of the current WormSource database and delete all data sets in the respective category files. This way, all gene sets can be collected, and a test run (see 4.3.9) with only the new data sets can be conducted. Afterwards and if no errors are found in the test run, the category files can be fused at once with the old database.

If a gene set has been curated, it is possible that WormBase IDs (WBGeneXXXXXXXX) are still missing. As there is no standard methodology of uploading differential gene expression analysis, some datasets only have gene IDs or gene names, whereas other already possess WormBase IDs. If WormBase IDs are provided, add WormBase Version to the file WormExp\_info (column WormBaseVersion) and this gene set can be directly added to respective category files (see 4.3.8). This information should be provided in the paper (but not always…). If no information is given, add n.a.

For all gene sets, that are not in WormBase ID format, use the following procedure. Go to <https://wormbase.org//tools/mine/simplemine.cgi>. It’s an online tool provided by WormBase to get WormBase ID for found genes. In Step 1 choose Caenorhabditis elegans. In Step 2 check “case insensitive input”, “download results as a tab-delimited file”, and “keep duplicate gene entries in results”. In Step 3 uncheck everything besides WormBase Gene ID.

It is helpful to use an Excel file to update the curated gene set from gene IDs to WormBase IDs. As “keep duplicate gene entries in results” was marked, the downloaded file from SimpleMine can have entries such as “Multiple entries found” or also “not found”. In the first case, keep both WormBase IDs found for this gene. In the latter case, delete the found gene. In any case, adjust the number of curated genes in the WormExp\_info file when deleting or adding multiple genes. Also add the respective WormBase Version that has been used for mapping the WormBase IDs.

If all gene sets have a WormBase ID mapping, they can be added to the category files of the database copy.

### Update database

If all category files have been updated, several other files have to be updated as well and the procedure is explained here.

#### c\_elegans.WS283.geneIDs

Go to <https://wormbase.org//tools/mine/simplemine.cgi>. In Step 1 choose Caenorhabditis elegans. In Step 2 check “case insensitive input”, “download results as a tab-delimited file”, and “keep duplicate gene entries in results”. In Step 3 uncheck everything besides WormBase Gene ID, Public Name and Sequence Name. In Step 4 click “Query all genes in this species” to get a full list of the current WormBase version.

Change this content to the same format as is currently used in the file c\_elegans.WS283.geneIDs (separated by comma instead of tab). Change the name of the file accordingly and add it to the WormSource folder.

#### Reference file

WormSource has an additional reference file that only contains the gene set name and its respective source (PubMed link, etc.). As this information is automatically curated in WormExp\_info, updating the reference file is fairly straightforward. Copy the columns “gene set names” and “Refs” and delete all duplicates of gene sets. Then the new table can be saved as tab delimited .txt file. This file should also be added in the database copy and can be fused later on with the pre-existing reference file.

#### Category files

Fuse all newly collected gene sets with the pre-existing category files by simply adding them under the entries. Make sure that no empty lines are in the file! Otherwise, an error will occur when booting the server.

#### WormExp folder

Three main files have to be changes for the Front End changes of the database: dat.properties, web.properties, and wormexp.properties.

Text

Description automatically generated

In dat.properties, add the name of idfile, if a new wormbase version has been added to the database. Additionally, if a new data set category is added, add it to the bottom of dataset in the same format as all the other data sets.

Text

Description automatically generated

In web.properties some texts of the front end can be changed if so wished. Additionally, add a new data set category, if wished. Keep the format as indicated by other categories.

Text

Description automatically generated

Text

Description automatically generated

In wormexp.properties more changes on the displayed text can be made. Important changes are the update of the current version as well as the Wormbase version.

Graphical user interface, text, application, chat or text message

Description automatically generatedAnother important and necessary update is the change of downloadable data sets from the website. Zip all updated category files plus WormExp.info and replace the file in WormExp -> upload. This changes the downloadable content here:

### Test Runs

#### Apache TomCat

To test if all changes were implemented correctly, test runs will be applied. In order to run the server locally and check all changes, Java should be installed, as well as apache tomcat.

The following steps describe how to locally run the server to conduct test runs:

1. Make sure, that the file test.jar is in both versions. Run Windows Powershell and go into the project directory that contains test.jar.
2. Run: “java -classpath test.jar com.dem.test.HiServer”
3. Go to “.\Apache Software Foundation\Tomcat 8.5\webapps” and copy the WormExp folder from the database folder into it.
   1. Attention: due to WormExp being hardcoded, two copies of the same folder are necessary at the moment: wormexpp and WormExp (problem will be solved in the future…)
4. Run TomCat
5. Go into your browser and write “localhost:8080/wormexpp/”
6. There should now be an exact copy of the WormExp website in your browser.

With this locally run server, test runs can be conducted, and changes can be made.

#### Selecting Test sets

##### Category changes

Test gene sets should be selected appropriate to the changes. However, standard test sets will be implemented and should be applied before every official update to confirm a smooth transition. Curated test sets are saved in the folder “test sets” and can be updated if necessary. Test sets for every category have one old and one new data set.

First, test gene sets will be collected (see 4.3.10.2 for more details) and tested on the old database to get a comparison baseline. Goal of the test run is to compare outputs of old version and new database version and check that it correctly adds new gene sets. Additionally, if new categories were added, it should be checked that those are selectable, and the server correctly accesses the new data files.

Things that should be tested:

* Upload of files
* All categories are selectable by server (especially important with new categories)
* Enrichment analysis also able to select all categories

1. Take one test set from every category and test if the finds the same hits
2. Make enrichment-analysis on respective category
3. Search the database by selecting three WormbaseIDs that are known to be in all categories to test search function (e.g. daf-2, clec-4,…)

##### Source code changes

Select test sets depening on the changes of the source code

#### Test runs

Document all outputs in word file

1. Use curated test files for test of old database -> document!
2. Use curated test files for test of new database -> document!
3. Fuse data bases and test again

### Checklist

|  |  |  |
| --- | --- | --- |
| **To-Do** | **Updated?** | **Fused?** |
| Wormbase\_version\_changes | yes | Not necessary |
| c.elegans.WS283.geneIDs | updated to from WS235 to WS283 | Not necessary |
| reference | Yes | No |
| Chemicalexposure-otherStress | Yes | No |
| DAF Insulin food | Yes | No |
| Development-Dauer-Aging | Yes | No |
| Kim Mounts | no | Not necessary |
| Mutants | Yes | No |
| Other | no | Not necessary |
| Pathogen (Microbes) | Yes | No |
| Targets | Yes | No |
| Tissue-specific | Yes | No |
| Epigenetics | Newly created; yes | Not necessary |
| WormExp\_info | yes | No |
| datafinder.ipynb | newly created; | Not necessary |

1. **Document History**

Reason for change

|  |  |  |
| --- | --- | --- |
| **Version number** | **Description of change** | **Valid from:** |
| 01 | 1. Version for establishing internal standards | 27.01.2022 |

Publication bibliography

Dozmorov, Mikhail (2016): GEOparse. Reading the NCBI’s GEO microarray SOFT files in R/BioConductor: public domain.

evogen (2015): Manual for WormExp, 2015. Available online at https://academic.oup.com/bioinformatics/article/32/6/943/1744078.

Yang, Wentao; Dierking, Katja; Schulenburg, Hinrich (2016): WormExp: a web-based application for a Caenorhabditis elegans-specific gene expression enrichnment analysis. In *Bioinformatics Advance Access* 32 (6).