Concise guide for the MGVB pipeline

Metodi V. Metodiev*

School of Life Sciences, University of Essex, United Kingdom

E-mail: mmetod@essex.ac.uk

Phone: +123 (0)123 4445556. Fax: + 44 (0) 1206 872592

MGVB is provided free to download and use for academic research and in software projects, but is not entirely open source at the present. It is the intention of the author that it will be made open source in the near future—following rigorous evaluations and feedback from the proteomics research community.

All programs listed in Table 1 can be run separately. The format of the commands and arguments for each one can be retrieved from the shell scripts `mgvb_auto.sh` and `mgvb_focused.sh`. The user can then write their own script to compose their own pipelines according to specific project needs. For example, it is not always necessary to extract from raw data files. This can be done just once and then the ms2 files can be used to run the analysis. For researchers, who rely on external LC-MS/MS services, it could be possible even to request the raw data be delivered in as ms2 files. Most facilities are familiar with the ms2 format and can do this. Alternatively, ms2 files can be parsed to binary mms files and these then can be used in multiple runs of the pipeline to search for different combinations of PTMs.

For combinatorial PTM searches, one does not always need to start from scratch. If `mgvb_auto.sh` has been already executed, it will be convenient to use the generated mms.txt files and proceed directly to run `scorer_mpi`. The shell script can be changed accordingly to do this.

A1. Differential expression and protein interaction analysis

MGVB is distributed as a single archive file. Download and expand the file in a convenient location on your system. When placed in this directory and expanded, the archive will create a subdirectory tree containing mgvb/bin. To install the executables and run the pipeline do:

1. Put mgvb/bin to the path:

   ```
   export PATH=$PATH:[path to mgvb]/bin
   ```

2. Copy raw data files and `config.rms` to a new project directory.

3. In the project directory, edit `config.rms` to set up the analysis. Change only the necessary entries: raw file names, fasta files and experiment names. If not interested in phosphorylations comment out the entry. Make sure precTol and tol are appropriate for the type of data—High/Low or High/High. The config file contains further instructions as comments to each line of code.

4. From within the project directory, execute:

   ```
   mgvb_auto.sh
   ```

5. Multiple result files will be generated. For each raw file there will be a `*.raw.ms2.mms.txt` file containing the top 2 candidate PSMs for each MS/MS scan. A `*.raw.ms2.mms.txt.db` file contains sqlite tables with peptides, proteins, protein groups and significant PSMs. A `*.raw.ms2.mms.sig_proteins.txt` file contains spectral counts for significant proteins for each raw file. Finally, the `consolidated_results.txt` file contains a table with all significant proteins detected in all raw files with their spectral counts.

A2. Automated focused analysis

1. Make sure MGVB is installed (executables and *.sh scripts are on the path).

2. Start with a clean directory containing `*.raw` files, copy `db1_min.db` file, `config.rms` and `config_focused.rms` from mgvb/bin to this directory.

3. Edit `config.rms` as in Appendix A1.

4. Execute:

   ```
   focussed_mgvb.sh cpu_num
   ```

Here cpu_num is the intended number of cpu/cores to be used.

A3. Interrogate results using sqlite3 functionality

1. Open desired file (let's say it is `WT1.raw.ms2.mms.txt.db`):

   ```
   sqlite3 WT1.raw.ms2.mms.txt.db
   ```

2. Inside the sqlite program execute the following to get all significant MS/MS scans assigned to a specific protein (let's say it is Scribble, which is encoded by SCRIB and in the significant protein list has id = 921. The significant protein list is in the file that has "`sig_proteins`" in its extension):

   ```
   select * from sig_scans where Sequence in (select sequence \
   from unnested where Protein = "921");
   ```

3. Alternatively, if one wishes to export the results to tab-delimited text file to be processed by Excel, Python or R, open the desired file in sqlite3 and execute:

   ```
   .header on
   .separator \t
   .output output_file_name.tsv
   select * from sig_scans where Sequence in (select sequence \
   from unnested where Protein = "921");
   ```

A4. Addressing common portability issues with Open MPI

The combinatorial PTM search engine, `scorer_mpi`, is provided as a binary file compiled with Open MPI version 5. This covers target platforms that implement this newest version of Open MPI. For example, the author uses it with the MGVB pipeline on a laptop running Ubuntu 22.04. However, it is possible that users might experience problems with some of the shared libraries required by the Open MPI implementation running on their server. For such cases the MGVB distribution provides an alternative portable `scorer_mpi` package, which includes a header file (`scmpi.h`), a shared library (`libscmpi.so`) and a source code file (`scorer_mpi_portable.c`), which can be easily compiled into a binary file to be used in the MGVB pipeline. The following steps will do this.

1. After mgvb.tar.gz file has been expanded there will be `/scorer_mpi_pack` directory in the main mgvb directory. Move to this directory:

   ```
   cd mgvb/scorer_mpi_pack
   ```

2. Build the binary. Assuming the `mgvb/scorer_mpi_pack` is in /home/user directory (replace "user" in the following command with the proper user name):

   ```
   mpicc -std=c99 -L/home/user/mgvb/scorer_mpi_pack -g \
       scorer_mpi_portable.c -o scorer_mpi_portable -lscmpi \
       -lmpfr -lgmp -ldl -lm
   ```

3. Modify and export `LD_LIBRARY_PATH` variable:

   ```
   export LD_LIBRARY_PATH=/home/user/mgvb/scorer_mpi_pack:\
   $LD_LIBRARY_PATH
   ```

4. Replace `scorer_mpi` with `scorer_mpi_portable`:

```
cd ../bin

mv scorer_mpi scorer_mpi_old

cp ../scorer_mpi_pack/scorer_mpi_portable scorer_mpi
```

The pipeline is now ready to run.