# MGVB: a new proteomics toolset for fast and efficient data analysis

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

## Abstract

MGVB is a collection of tools for proteomics data analysis. It covers data processing from *in silico* digestion of protein sequences to comprehensive identification of pos-translational modifications and solving the protein inference problem. The toolset is developed with efficiency in mind. It enables analysis at a fraction of the resources cost typically required by existing commercial and free tools. MGVB, as it is a native application, is faster than existing proteomics tools such as MaxQuant and, at the same time, finds very similar, in some cases even larger number of peptides at a chosen level of statistical significance. It implements a probabilistic scoring function to match spectra to sequences, and a novel combinatorial search strategy for finding post-translational modifications, and a Bayesian approach to locate modification sites. This report describes the algorithms behind the tools, presents benchmarking data sets analysis comparing MGVB performance to MaxQuant/Andromeda, and provides step by step instructions for using it in typical analytical scenarios.

# Introduction

Proteomics aims to identify and quantify the proteins expressed in a sample under study at the genome scale (reviewed in[1]). Presently, the technology of choice—almost universally employed in large-scale proteomics studies—is high-resolution mass spectrometry of proteolytic digests. Modern hybrid mass spectrometers, when interfaced with nano-scale liquid chromatography, generate many thousands tandem spectra of peptide precursors per run; typical projects often generate more than a million spectra (see for example[2–5] describing the Orbitrap mass analyser and large-scale proteomics projects that have used it).

Raw mass spectra are processed by computational pipelines to identify and quantify the proteins. These pipelines utilise search engines that typically match peptide fragmentation spectra to the theoretically predicted sequence specific fragments, i.e. predicted from genomic sequences. Matching is a probabilistic process prone to false positive and false negative results. To account for this, search engines such as Mascot and Andromeda, apply filters, most commonly based on the number of reverse database hits.[6,7]

With the advance of instrumentation data processing is becoming the bottleneck of proteomics workflows. To illustrate: practitioners of the art know that at the present a nano-LC-MS/MS experiment will generate more than 100,000 spectra in an hour or two but MaxQuant or Mascot or Sequest analysis of the file would take substantial time even on a powerful multicore workstation. In our hands, even when MaxQuant is run under Mono on a high-performance computing cluster, analyses programmed to search for post-translational modifications take longer than the time needed to generate the raw files.

Even more challenging computationally is the recently proposed open search approach, which attempts to identify peptides modified by unknown groups. One implementation of this approach is the MSFragger algorithm.[8] It uses precomputed indexed database of peptide fragments to search the MS/MS spectra. MSFragger was publicised as an ultrafast algorithm and it is indeed very fast but, as illustrated in Fig.1, would struggle with certain types of modified peptides.

Part of the reason for this level of performance is technical: most freely-available search engines are implemented as non native applications running in Java or .Net virtual machines, in part to avoid platform dependence. This puts substantial overheads on memory requirements and execution speed. Another reason is the relative ease of developing in Java and C#, as they are object oriented garbage-collected languages with rich application libraries ecosystems. However the platform dependence is a less severe problem nowadays as it used to be in the past. A native search engine could potentially provide much faster processing and would come with the added benefit of much smaller carbon footprint as it would use much less memory and processor time. This was the initial motivation behind the MGVB project: to develop a native search engine that could perform as well as the state of the art programs in terms of peptide and protein coverage, but do it faster and with less energy consumption. As it turned out—in the process of development—a new approach to post-translational modification analysis was conceived. It is a combinatorial search that combines some of the advantages of the open database search but is more capable for certain types of modifications as it is can handle efficiently peptides modified on more that one site and by more than one type of modification, a capability that open search algorithms such as the MS-Fragger lack by design as they only recognise the total delta mass of the modifications. This leads to limitations in identifying modified fragment ions when two or more modifications are found on the precursor peptide. This is illustrated in Fig 1.

MGVB consists of several programs, which prepare peptide sequences, extract raw spectral data to proprietary binary files, match spectra, infer proteins and compute spectral counts to quantify the identified proteins. The `scorer` program implements probabilistic algorithms for spectra assignment to sequences, similar to MaxQuant/Andromeda.[7] To speed up modified peptides identification, for each candidate precursor sequence, predicted fragments are packaged in a balanced binary search tree, which is used for matching the spectral peaks. A binomial probability score is assigned based on the number of fragments matched. Modifications sites are determined by a Bayesian updating algorithm, which con-

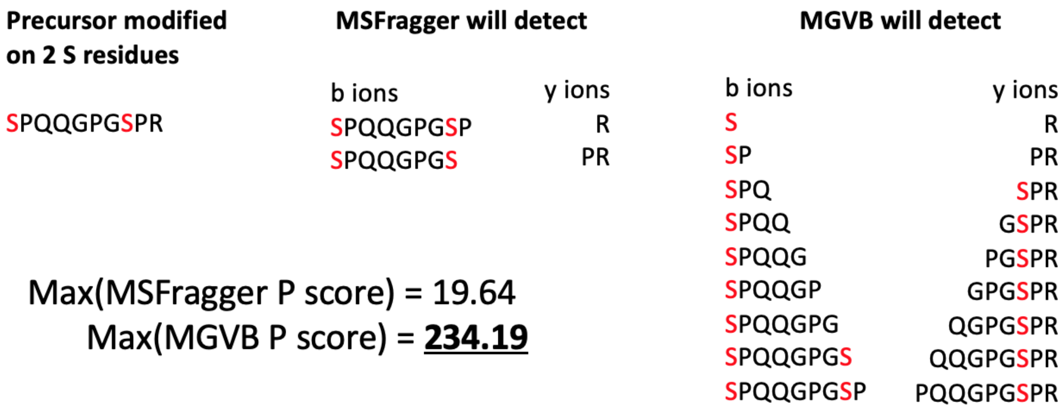


Figure 1: An example where open database search algorithms would likely fail: two PTMs located near the start and end of the peptide sequence. The combinatorial PTM search algorithm implemented in MGVB can identify y and b fragments and accurately map modification sites in cases where 2 or more residues are modified. Existing open database search engines such as MSFragger cannot do this as they only recognise either unmodified fragment ions or fragment ions that carry all modifications. As a consequence the maximum score that can be achieved by MSFragger would be more than 10 times smaller than the one assigned by MGVB. The match would then likely lose the competition with alternative candidates and will not be reported by MSFragger while MGVB will report it as top candidate. The P scores presented in the figure were calculated assuming q = 5. The score for MSFragger was calculated assuming 4 out of 18 fragment ions were detected. The score for MGVB was calculated assuming 18 out of 18 ions were detected.

siders assigned fragments as experimental evidences of possible PTM localisation models to compute the final posterior probabilities of localisation. `Scorer` outputs results to a text file with extension `*.raw.ms2.txt`. The file contains information about the top and the second best matches including number of fragments matched, score, retention time, proteins corresponding to the matched spectrum, precursor mass and more.

The combinatorial search algorithm, named `scorer_mpi`, uses a precomputed database of combinations of up to 3 different post-translational modifications masses called `mod_comb`. The current compilation of `mod_comb` consists of 1394 triplet combinations of 30 different modifications from the Unimod database (Unimod is accessible at `http://www.unimod.org`). `Scorer_mpi` executes an open database search to identify a set of candidate precursors. It then calculate the delta mass for each of the candidates and searches the mod_comb database for combinations matching the delta-mass. Once such combination is identified, scorer_mpi uses the same Bayesian updating algorithm as the scorer program to accurately determine the location of each modification from the combination.

A very important advantage of this approach is that only the initial precursor search needs to be restricted to high-mass accuracy. The search for MS/MS fragments matches does not need such accuracy. This makes the huge amount of legacy raw data acquired in the High/Low mode of analysis amenable to processing by scorer_mpi.

# Methods

This section describes the implementation and the algorithms used in MGVB.

## Implementation

With a single exception, MGVB is implemented in C and compiled in a collection of binary executable files, which can be used on hardware running the Linux operating system. The one exception, the program used to extract spectra from the proprietary raw files is implemented in C# in order to leverage the API provided by Thermo Fisher Scientific, but availability of .Net is not required to run it as the program is packaged into an executable file. Table 1 lists the different programs, and gives brief description of their utility.

The following external libraries were used to develop MGVB: the arbitrary precision number library MPFR from GNU;[13] OpenMP (for shared memory parallelism);[14] Open MPI (for message-passing parallelism);[15] sqlite3 (for storing and querying data);[16] GNU Scientific Library (for fitting linear models with interaction terms for precursor mass recalibration).[17]

In addition to the binary files included in Table 1, there is a collection of shell scripts, which automate the various modes of analysis possible with MGVB. These are described in Appendix A.

Running the MGVB pipeline using the provided shell scripts will create numerous files on

Table 1: List of MGVB modules

| Name | Description |
|---|---|
| `digest_universal` | In silico digestion of sequences from FASTA files. |
| `mod_pep` | Generate modified peptides sequences. |
| `toSQL` | Create sqlite tables with sequences from FASTA files. |
| `extractRaw` | Extracts from raw files and writes MS2 and MS1 files to disk[1]. |
| `parseMS` | Parses MS2 files to proprietary binary mms files. |
| `scorer` | Uses mms files to search for peptide matches[2]. |
| `scorer_mpi` | Uses mms and two sqlite databases to search for PTM[3]. |
| `deep_seq` | Prepares sequences for combinatorial PTM search. |
| `select_by_pep` | Prepares data for applying the local FDR filtering. |
| `create_aggregated` | Generate combined statistics for filtering. |
| `select_sig_psms` | Filters candidate peptides using the local FDR approach. |
| `select_by_pep_open` | Used for data generated by `scorer_mpi`. |
| `mgvb` | Consolidates results for multiple samples. |

[a] MS1 and MS2 files are text files that can be used by various proteomics search engines.[18]
[b] Uses the OpenMP library for shared memory parallel processing.
[c] Uses the Open MPI library for message passing parallel computations.

the host computer. Many of the files contain intermediate information and can be deleted after the project is complete. There are MS1 and MS2 files, which are human readable files with MS and MS/MS scan information closely following the format that is used by RawConverter.[18] The mms files are binary files containing the same information. The db files are sqlite database files used to generate the final report. They contain all peptide and protein level data obtained in the course of analysis.

## Outline of algorithms

The detailed description of the scoring algorithm implemented in the `scorer` and `scorer_mpi` programs is not provided as it closely follows the published Andromeda scoring algorithm: the same binomial score and the same approach of filtering the top q spectral peaks as MaxQuant/Andromeda are used.

Where MGVB differs is the algorithm for PTM localisation. Andromeda uses a mapping algorithm that is similar to the one used to match spectra to sequences.[7] A binomial probability derived score is computed for each possible PTM assignment model and models with scores above a preselected threshold are selected.

MGVB departs from this approach and uses a simple Bayesian updating algorithm to assign localisation probabilities. The process starts with generation of all possible PTM localisation models. These are assigned equal prior probabilities. These priors are then updated using the detected fragments as experimental evidences to obtain the posterior probabilities of localisation.

The Bayesian updating algorithm is described in Algorithm 1. MGVB assumes a likelihood of 0.7 if a fragment is compatible with a localisation model and likelihood of 0.2 if not (the likelihood is the probability that the fragment will be observed given that the model in question is true).

**Algorithm 1** Calculate PTM localisation models probabilities

```
$models \leftarrow computeModels$                                  ▷ generates N models
$p \leftarrow (\frac{1}{N}, \frac{1}{N} \cdots \frac{1}{N})$        ▷ assigns uniform prior
$F \leftarrow$ masses of matched fragments in spectrum
for all $f$ in $F$ do
    for all $m$ in $models$ do
        if $f$, $m$ are compatible then
            $p_m \leftarrow p_m \times \lambda_1$          ▷ if compatible, lambda is set to 0.7
        else
            $p_m \leftarrow p_m \times \lambda_0$          ▷ if not compatible it is usually 0.2
        end if
    end for
end for
Renormalise $p$ to sum to 1
```

The combinatorial PTM search algorithm implemented by `scorer_mpi` is presented in Algorithm 2. It uses a precompiled database of combinations of up to 3 different PTMs and in its present version covers 30 different modifications derived from Unimod.

The three modules, `select_by_pep`, `create_aggregated`, and `select_sig_psms` (see Table 1) work as a filter pipeline to select peptide hits based on decoy database hits. The filtering pipeline computes the local false discovery rate (also known as posterior error probability, or PEP, see[19] for details of how PEP is computed) using all data available as baseline for calculating probabilities for different classes of peptide matches. As in MaxQuant the two variables are the binomial probability score and the length of the peptide sequence. PEP is computed after the score and length densities are estimated using a histogram method, and results are filtered to 1% FDR using PEP.

Algorithm 1 calculates posterior probabilities for the possible PTM localisation models. Model probabilities $\mathrm{p_j}$ are then converted to site probabilities $\mathrm{P_i}$ using the following equations:

$$P_i = \sum_{j=1}^{N} p_j \times \gamma_j^i. \tag{1}$$

where,

$$\gamma_j^i = \begin{cases} 1 & \text{if site i is modified under model j} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

**Algorithm 2** Combinatorial PTM search implemented by scorer_mpi

```
$mod\_comb$ ← generate mod_comb                              ▷ generates delta mass db
$peptides$ ← generate peptide db                             ▷ generates peptides db
for all $s$ in mms file do                                   ▷ mms file contains spectra
    $M$ ← $peptides$ matching parent of $s$                  ▷ usually ±500 Da
    for all $m$ in $M$ do
        $\delta$ ← compute delta mass for $m$
        $ptm$ ← PTM combinations from $mod\_comb$ matching $\delta$
        for all $c$ in $ptm$ do
            $score$ ← score $s$ against $m$ modified by $c$, save to results
        end for
    end for
end for
```

For combinatorial PTM data the algorithm is changed to implement a hybrid approach to optimising FDR: the raw binomial probability score is still used but an additional threshold is calculated as in the Mascot algorithm, which is $-log_{10}(0.01/n\_seq)$ where n_seq is the number of candidate sequences. Candidates that have scores below the threshold are filtered out. The filtering then proceeds to compute PEP as above and filter at 1% FDR using the PEP values. This is implemented in `select_by_pep_open`.

Algorithm 3 solves the protein inference problem: given a set of matched peptides passing the score thresholds, what is the optimal peptides to proteins assignment? This is not a trivial problem as many proteins encoded by distinct genes share sequence similarities, which cause peptides to be shared across groups of proteins. MGVB solves the protein inference problem by implementing a recursive algorithm, which assigns peptides to the protein with the highest protein score in the protein group that share these peptides. Two data structures are involved: a linked list of protein groups, each node containing a list of identities of the proteins in the group, count of spectra matching the group, and a pointer to the next element of the list. In addition, an array of protein data structures is used. This array is sorted by protein score to allow efficient searching.

**Algorithm 3** Recursive protein inference from peptide matches

```
pr_groups ← generate linked list of protein_group structs
proteins ← generate sorted array of tuples of protein IDs with scores
function PROCESS_PROT_GROUPS(pr_groups, proteins)
    if pr_groups is empty then
        return
    end if
    names ← split protein IDs in first node of pr_groups into array
    Pr ← the element of names with the highest score in proteins
    delete all pr_groups nodes containing Pr summing their counts to c
    save Pr, c to results
    return PROCESS_PROT_GROUPS(pr_groups, proteins)
end function
PROCESS_PROT_GROUPS(pr_groups, proteins)
```

Algorithm 3 outputs to a file containing spectral counts and protein IDs. MGVB also contains facilities for combining such files into an aggregated report file of tab-separated values, which can be further analysed in R, Python, Excel or any other environment for machine learning and statistical analysis. In addition, MGVB creates sqlite database files for each raw file analysed, which contain plethora of information about scans, peptides, modification etc. These can be analysed by sqlite functions or other software operating on SQL databases. Some simple but useful report functions are given in Appendix A.

## Results and discussion

MGVB was tested with three different datasets as described below. First, a collection of raw data files obtained from experiments with human cell lines was used.. An LTQ/Orbitrap Velos isnstrument was used to generate the data as described in.[4,9,10] The results were obtained by analysing data from immunoprecipitation experiments using GFP-tagged Scribble as bait expressed in human HEK293 cells as described in.[9] The performance of MGVB was compared to MaxQuant, version 1.6.1.0 running on the same hardware.

The following performance metrics were compared: number of MS/MS spectra identified at 1% FDR (significant PSMs), number of proteins identified at 1% FDR, number of PTM identified for the target protein Scribble, speed: time for completing the different steps of the analysis, memory consumption, CPU time.

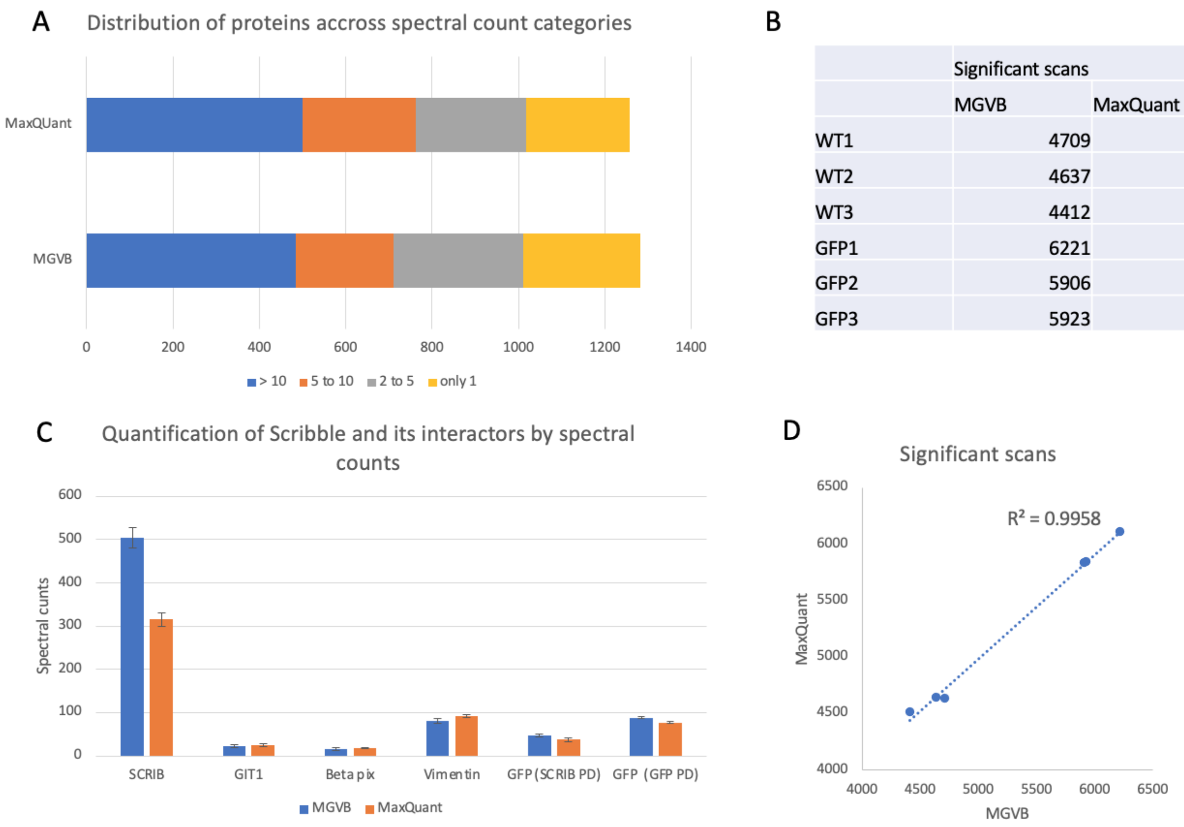
Another experiment was performed with data from TP53 immunorecipitation experiments downloaded from the MassIVE repository (dataset ID: MSV000095580). This dataset is generated on a newer instrument (quadrupole Orbitrap instrument).

The third experiment was with 3 raw datafiles from a large-scale phosphoproteomics study.[11] This dataset contains data from phosphopeptide enriched samples from mouse biopsies (MassIVE dataset ID: MSV000082502).

The results from the benchmarking experiments are summarised in figures 2-3 and Supplementary Tables 1 and 2. Fig. 2 shows the peptide and protein identification performance of MGVB compared to MaxQuant/Andromeda. MGVB and MaxQuant find very similar numbers of peptides and proteins at the chosen FDR. However, MGVB assigns more spectra to the bait protein—which is the most abundant protein in the sample by a large margin. Similarly, MGVB assigns slightly more spectra to GFP. The spectral counts for the known

Scribble-interacting proteins GIT1 and ARHGEF7 assigned by MGVB and MaxQuant are very similar.

Fig. 3 compares the performance of MGVB and MaxQuant in terms of execution speed, memory consumption and CPU usage. MaxQuant was run under Mono on the same hardware configuration as MGVB. MaxQuant and MGVB were set to search for up to 3 PTMs per peptide and the modifications were set to N-terminal acetylation, methionine oxidation and STY phosphorylation. MGVB outperformed MaxQuant in speed by a large margin and consumed significantly less memory and CPU time per run in all benchmarking experiments. In these experiments MaxQuant was set to not perform second peptide search to make speed comparisons fair. Both search engines conducted first search and precursor mass recalibration followed by a main search.



|  | Significant scans |  |
|---|---|---|
|  | MGVB | MaxQuant |
| WT1 | 4709 | 4633 |
| WT2 | 4637 | 4635 |
| WT3 | 4412 | 4508 |
| GFP1 | 6221 | 6103 |
| GFP2 | 5906 | 5832 |
| GFP3 | 5923 | 5839 |

Figure 2: Performance metrics of MGVB compared to MaxQuant. MGVB and MaxQuant were run on the HPC cluster Ceres at University of Essex, UK. Both programs were run on 24-core instances through the grid engine. MaxQuant was run under Mono. Both programs were set to use at most 24 threads: MGVB by setting the environmental variable OMP_NUM_THREADS=24; MaxQuant, by setting the number of threads in mqpar.xml. **A**: number of proteins identified stratified by spectral count ranges. **B**: number of PSMs passing the FDR filters obtained from the 6 raw files, the numbers of scans assigned from each file were computed by summing up the MS/MS counts from proteinGroups.txt (MaxQuant) and consolidated_results.txt (MGVB); **C**: spectral counts assigned to the bait protein Scribble and 4 of its known interacting proteins, and to the negative control bait, GFP; **D**: correlation plot of the data presented in **B**. Error bars represent standard deviation from 3 independent LC-MS/MS runs.

The newest version of MaxQuant, which can now run in .Net 8 in linux was also tested with the Scribble dataset. The performance was similar to the version used with Mono (data not shown).

In the experiments with data downloaded from MassIVE, the newest MaxQuant was used. The results from these experiments are summarised in supplementary tables 4 and 5. In the experiments with data from TP53 immunoprecipitation, both MGVB and MaxQuant

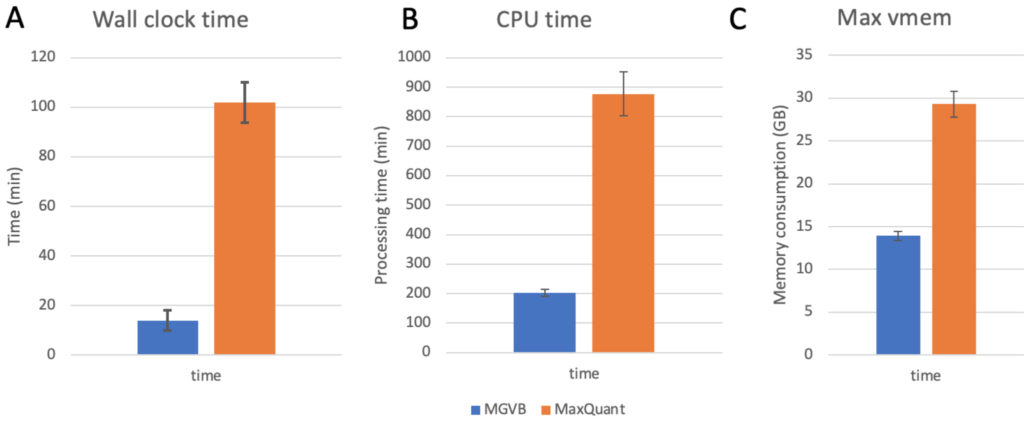


Figure 3: Performance metrics of MGVB compared to MaxQuant. MGVB and MaxQuant were run on the HPC cluster Ceres at University of Essex as described in Fig. 2. **A**: wall clock times for processing 6 raw file; **B**: CPU times for the same task; **C**: maximum used memory for the same task. Error bars represent standard deviation from 3 independent experiments.

identified the bait protein, TP53, as one of the most abundant proteins in two of the raw data files generated from cell transfected with wild type or mutant TP53, but not in the raw file from the negative control experiment with cells transfected with an empty plasmid. Quite interestingly, both MGVB and MaxQuant identified a number of proteins that are pulled down with the wild type TP53 but not the mutant. The top 3 that were noticed by a very preliminary analysis were Culin 7, Culin 9 and Mdm2. This is potentially important as it suggest that this very frequent mutation, R175H—an oncogenic driver in many human tumours—changes the way TP53 interacts with the ubiquitination and degradation machinery of the cell.

In the experiment with datasets from phosphopeptides isolated from mouse biopsies, which have very high complexity, MaxQuant reported 13-16% more identified PSMs (SI Table 5) demonstrating its advanced peak picking and deisotoping capability. This difference was mostly due to the extra MSMS scans identified with precursor isotope index different from 0 and 1, which are the isotopes that MGVB considers by default. This difference diminishes to 3-5% if MGVB is set to search each MSMS scan repeatedly for isotope index -1, 0, 1, and 2 but this is neither efficient nor should be necessary as the majority of precursors have well defined isotope patterns in the MS scan data and the mono-isotopic peak can easily be identified by an appropriate algorithm. This is being implemented in the next version of MGVB, which is now in a testing/debugging stage.

We have also recently used MGVB and MaxQuant to search for differentially expressed proteins and post-translational modifications in a dataset obtained in experiments, in which a collection of human breast cancer cell lines were treated with interferon gamma.[12] The raw data and results are available from MassIVE, dataset ID: MSV000095497.

The open search capabilities of MGVB are demonstrated with data in Supplementary Table 3. The table shows that the combinatorial search identifies most known phosphorylation sites of Scribble, as well as known ubiquitination sites and some candidate novel acetylation and methylation sites of potential interest.

MGVB can perform combinatorial PTM searches to identify pre-selected sets of post-translational modifications in two modes of operation: in an unrestricted mode it searches against the entire genome of the organism under study. This is challenging and requires a high-performance computer. Typically, in such experiments MGVB was run on up to 40 cpus using its inbuilt high-performance message-passing functionality. An alternative mode, the focused search, restricts the analysis to a subset of sequences selected from a preceding ultrafast stringent search. Typically, MGVB was set to initially search without any modification allowed and no missed cleavages. Such searches complete under a minute on a multicore workstation (8 cores). Proteins identified in such searches are then subjected to a focused combinatorial search. Supplementary Table 1 shows that the focused approach correctly identifies most of the phosphorylation sites known for Scribble and suggests several novel modifications.

Compared to MGVB, recently published open search algorithms such as the one implemented in MSFragger[8] provide unrestricted modification mass search. However, such algorithms identify candidate peptide modifications by matching the delta mass of the identified peptide to candidate modification in a process that is inherently limited to recognising the modification as a single added group. A lengthy post processing employing machine learning is required to make sense of the initial assignments. For example, a delta mass of W units cannot be immediately assigned to a modification of type A on residue X plus a modification of type B on residue Y because the algorithm has no way of learning what the individual masses A and B are from their sum. It will report the delta mass and the peptide sequence but it would be up to the researcher to hunt for the identities of A and B. Even more limiting, to the extend of making open search engine unusable for such cases, is the fact that if there are two modification on two different residues, many of the spectral peaks corresponding to y and b fragments will not be assigned by the open search engine (see Fig 1 for illustration).

In contrast, MGVB would use the scorer_mpi module to execute a combinatorial search, which would directly identify the two modifications, A and B, and will localise them to the correct residues—all in a single step analysis.

# Concluding remarks

MGVB is work in progress. It is easy to use, and delivers results comparable, if not even better than existing tools. If what is needed is seamless expression profiling, or PTM analysis of your favourite protein, or protein interaction analysis, and the user prefers a single command line tool that can be used with minimum effort, it will deliver. It is usually faster than the existing platforms and uses much less power and memory.

However, there are still important things to be done. MGVB does not implement second peptide search, and the handling of isotopes could be improved, which will boost speed significantly and increase the percentage of confidently identified PSMs. It will be also good to implement ion intensity-based quatification in the future. Perhaps most important, MGVB should implement more advanced machine learning algorithms for optimising FDR and should be able to handle isobaric tags and stable-isotope based quantitative workflows.

This is all planed and will be implemented in the next, open source edition of MGVB.

## Supplementary information

Supplementary Table 1 contains the proteinGroups.txt from MaxQuant for the 6 raw files from the Scribble immunoprecipitation experiments. Supplementary Table 2 contains the consolidated protein report from MGVB for the same raw files. Supplementary Table 3 contains results from combinatorial search of post-translational modifications performed on WT1.raw data file as described in Appendix A.2. Supplementary Table 4 contains results generated by MGVB and MaxQuant with the 3 raw files from the TP53 immunoprecipitation experiments downloaded from the Massive repository. Supplementary Table 4 contains results generated by MGVB and MaxQuant with the 3 raw files from the phosphopeptide enrichment of mouse biopsies downloaded from the Massive repository. Supplementary files contain a single MGVB archive file, which can be used to install the toolset and run it, an sqlite3 file, `db1_min.db`, which is needed to carry out combinatorial searches, and two config files: `config.rms` and `config_focused.rms`, which can be edited to easily set up custom searches.

## Data and code availability

The raw data files used to generate the reported results for Scribble immunoprecipitation are available via ProteomeXchange with identifier PXD051331. The raw data files for the TP53 immunoprecipitation experiment are available from MassIVE (`https://massive.ucsd.edu/ProteoSAFe/static/massive.jsp`) with identifier MSV000095580. The raw data files for the analysis of enriched phosphopeptides from mouse biopsies are available from MassIVE with identifier MSV000082502.

MGVB binaries, source code necessary to compile and make the portable versions of scorer and scorer_mpi, along with templates of config files, modification database, shell scripts, and instructions are available from GitHub at `https://github.com/mvm1964/MGVB`.

## Acknowledgments

The author acknowledges financial support from NERC (grant NE/P002374/1), from BBSRC (grant BB/V006916/1), and from University of Essex (internal grant to acquire the LTQ/Orbitrap Velos instrument).

## Declarations

- No conflict of interest is reported.
- No ethics approval was necessary.
- The author was solely responsible for developing MGVB and preparing the manuscript.

## References

(1) Aebersold, R.; Mann, M. Mass-spectrometric exploration of proteome structure and function. *Nature* **2016**, *537*, 347–55.

(2) Scigelova, M.; Makarov, A. Orbitrap mass analyzer–overview and applications in proteomics. *Proteomics* **2006**, *6 Suppl 2*, 16–21.

(3) Makarov, A.; Denisov, E.; Kholomeev, A.; Balschun, W.; Lange, O.; Strupat, K.; Horning, S. Performance evaluation of a hybrid linear ion trap/orbitrap mass spectrometer. *Anal Chem* **2006**, *78*, 2113–20.

(4) Croner, R. S.; Sturzl, M.; Rau, T. T.; Metodieva, G.; Geppert, C. I.; Naschberger, E.; Lausen, B.; Metodiev, M. V. Quantitative proteome profiling of lymph node-positive vs.

-negative colorectal carcinomas pinpoints MX1 as a marker for lymph node metastasis. *Int J Cancer* **2014**, *135*, 2878–86.

(5) Alldridge, L.; Metodieva, G.; Greenwood, C.; Al-Janabi, K.; Thwaites, L.; Sauven, P.; Metodiev, M. Proteome profiling of breast tumors by gel electrophoresis and nanoscale electrospray ionization mass spectrometry. *J Proteome Res* **2008**, *7*, 1458–69.

(6) Perkins, D. N.; Pappin, D. J.; Creasy, D. M.; Cottrell, J. S. Probability-based protein identification by searching sequence databases using mass spectrometry data. *Electrophoresis* **1999**, *20*, 3551–67.

(7) Cox, J.; Neuhauser, N.; Michalski, A.; Scheltema, R. A.; Olsen, J. V.; Mann, M. Andromeda: a peptide search engine integrated into the MaxQuant environment. *J Proteome Res* **2011**, *10*, 1794–805.

(8) Kong, A. T.; Leprevost, F. V.; Avtonomov, D. M.; Mellacheruvu, D.; Nesvizhskii, A. I. MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. *Nat Methods* **2017**, *14*, 513–520.

(9) Metodieva, G.; Adoki, S.; Lausen, B.; Metodiev, M. V. Decreased Usage of Specific Scrib Exons Defines a More Malignant Phenotype of Breast Cancer With Worsened Survival. *EBioMedicine* **2016**, *8*, 150–158.

(10) Greenwood, C.; Metodieva, G.; Al-Janabi, K.; Lausen, B.; Alldridge, L.; Leng, L.; Bucala, R.; Fernandez, N.; Metodiev, M. V. Stat1 and CD74 overexpression is co-dependent and linked to increased invasion and lymph node metastasis in triple-negative breast cancer. *J Proteomics* **2012**, *75*, 3031–40.

(11) Humphrey, S.; Azimifar, S.; Mann, M. High-throughput phosphoproteomics reveals in vivo insulin signaling dynamics. *Nat. Biotechnol.* **2015**, *33*, 990–5.

(12) Vasileva-Slaveva, M.; Yordanov, A.; Metodieva, G.; Metodiev, M. Exploring Protein Post-Translational Modifications of Breast Cancer Cells Using a Novel Combinatorial Search Algorithm. *Int J Mol Sci* **2024**, *25*.

(13) Fousse, L.; Hanrot, G.; Lefèvre, V.; Pélissier, P.; Zimmermann, P. MPFR: A Multiple-Precision Binary Floating-Point Library With Correct Rounding. *ACM Transactions on Mathematical Software* **2007**, *33*.

(14) Chandra, R.; Dagum, L.; Kohr, D.; Menon, R.; Maydan, D.; McDonald, J. *Parallel programming in OpenMP*; Morgan Kaufmann, 2001.

(15) Message Passing Interface Forum, MPI: A Message-Passing Interface Standard Version 4.0. 2021.

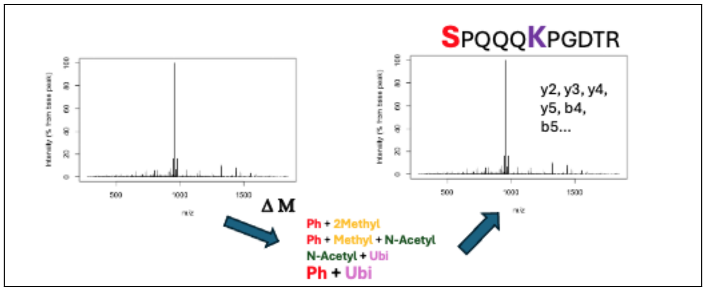
(16) Hipp, R. D. SQLite. 2020; `https://www.sqlite.org/index.html`.

(17) Gough, B. *GNU scientific library reference manual*; Network Theory Ltd., 2009.

(18) He, L.; Diedrich, J.; Chu, Y. Y.; Yates, r., J. R. Extracting Accurate Precursor Information for Tandem Mass Spectra by RawConverter. *Anal Chem* **2015**, *87*, 11361–7.

(19) Käll, L.; Storey, J. D.; MacCoss, M. J.; Noble, W. S. Posterior error probabilities and false discovery rates: two sides of the same coin. *J Proteome Res* **2008**, *7*, 40–4.

# Graphical TOC Entry


SPQQQKPGDTR
Intensity (% from base peak)
m/z
y2, y3, y4,
y5, b4,
b5...
Δ M
Ph + 2Methyl
Ph + Methyl + N-Acetyl
N-Acetyl + Ubi
Ph + Ubi