

# MS-Cluster v2 - A short user's manual

Ari M. Frank - arf@cs.ucsd.edu

October 18, 2010

## Abstract

MS-Cluster is a program that performs large-scale clustering of MS/MS proteomics data. This document will explain how to run the software and how to create and search spectral archives using it. If you have any questions, corrections or bug reports, please contact the author.

## 1 Copyright

Copyright 2010, The Regents of the University of California All Rights Reserved

Permission to use, copy, modify and distribute any part of this program for educational, research and non-profit purposes, without fee, and without a written agreement is hereby granted, provided that the above copyright notice, this paragraph and the following three paragraphs appear in all copies.

Those desiring to incorporate this work into commercial products or use for commercial purposes should contact the Technology Transfer & Intellectual Property Services, University of California, San Diego, 9500 Gilman Drive, Mail Code 0910, La Jolla, CA 92093-0910,

Ph: (858) 534-5815, FAX: (858) 534-7345, E-MAIL:invent@ucsd.edu.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE SOFTWARE PROVIDED HEREIN IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATION TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS. THE UNIVERSITY OF CALIFORNIA MAKES NO REPRESENTATIONS AND EXTENDS NO WARRANTIES OF ANY KIND, EITHER IMPLIED OR EXPRESS, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR THAT THE USE OF THE SOFTWARE WILL NOT INFRINGE ANY PATENT, TRADEMARK OR OTHER RIGHTS.

## 2 Installation

The MS-Cluster package includes a Makefile source files, model files and a pre-compiled WIN32 executable. To compile the program simply type make in the main directory, and if all goes well the executable MSCLuster\_bin should be generated.

## 3 Clustering

### 3.1 Overview

MS-Cluster v2 algorithm [2] is based on the algorithmic design of the MS-Cluster algorithm [1]. The main difference between the two involves the implementation which is much more efficient in the newer version. This enables the program to process very large datasets that can reach billions of spectra, compared to tens of millions that were the limit of the previous version.

The initial input to the clustering algorithm is MS/MS spectra (mgf or mzXML formats). The program first preprocesses the data and writes it into machine readable binary files (“dat” files). The preprocessing involves filtering and merging adjacent MS/MS peaks, normalizing the peak intensities and quality filtering that removes spectra that are not likely to contain a peptide signal (this step is optional). As opposed to the original input files which have scans with a variety of precursor  $m/z$  values, the dat files contain only spectra with certain precursor  $m/z$  ranges (usually the file name will indicate it spectra’s  $m/z$  values). This aids the clustering process by making it easily load specific portions of the data.

The clustering algorithm outputs several types of files: consensus spectra for the clusters (mgf or dat format), cluster membership files, and lists of output files.

### 3.2 Naming conventions and output files

When running a clustering job you always need to provide the output name (e.g., HUMAN, UCSD.DATA, etc.). This name is used as a prefix for all output file names. Following that there are two indices which are by default 0. So a typical output file name will look something like HUMAN\_0.0\_xxx. The first index is the dataset or epoch index for the archive. If archives are merged, this index can get incremented. The second index is the batch index. To facilitate the timely execution of very large clustering jobs, we can split the task into several batches, each can be given its own unique batch index to avoid duplicate names (for files or clusters). The dataset and batch indices can be set with the command line flags `--dataset-idx` and `--batch-idx` as described below.

There are two main types of output files generated by the clustering algorithm: spectra files of the consensus spectra of each cluster (in mgf or MS-Cluster’s binary dat formats) and cluster membership files which lists the members of each cluster (“clust” files). The entries in the clust file have the following format. Each cluster has a header line listing its unique cluster id, the cluster size, the cluster precursor  $m/z$  and the cluster’s charge (typically will be 0 unless the option `--assign-charges` is used. Following that there are lines for each cluster member with the following format: dataset index (typically 0), file index, scan number, precursor  $m/z$ , similarity to consensus, p-value of the similarity to the consensus, and the precursor charge (typically 0). If the spectrum was annotated with a sequence (e.g., an mgf file with the tag `SEQ=...`) then the sequence will appear at the end.

For example, the following is an entry for a cluster of size 3:

# ↓ Cluster Idx	size	cluster $m/z$	charge				
PNNL_0_11.1229412	3	760.883	0				
56	7364	5809	761.147	0.81	0	0	
56	11654	2635	761.18	0.39	4.167e-02	0	
70	298	14394	761.09	0.50	0	0	
# ↑ dataset idx	file idx	scan number	$m/z$	similarity	p-value	charge	

It has two spectra from dataset 56 and one from dataset 70. The second spectrum has a lower similarity of 0.39 to the consensus which has a p-value of 0.071 (the other two spectra are much more similar and have a p-value of close to 0).

### 3.3 Command-line options

MS-Cluster is an executable that is designed to be run from the command line. Below is an explanation of the most common flags that can be used. The full list of flags can be viewed by running the program with the flag `--help`.

- `--output-name <NAME>` - sets name of all output files to NAME (mandatory flag for most applications).
- `--list <path>` - The full path to a text file with the input MS/MS spectra files. The text file should have the full path to each input file on a separate line.
- `--dat-list <path>` - If the data was already preprocessed and the temporary dat files were not deleted (for e.g., by specifying the `--keep-dat` flag) then the clustering can commence without that stage.

Many of the flags are optional (the more important ones appear on top):

- `--model-dir <path>` - directory where model files are kept (default `./Models`). *Note that if the current directory is not the same as the one that holds the MS-Cluster executable you need to specify the full path to the “Models” directory with this flag. If running MS-Cluster on Windows and not from the current directory you should specify the path to “Models\_Windows”.*
- `--memory-gb <M>` - the number of GB of RAM available for this process (Must allocate at least 0.5GB, do not allocate more than 3GB on a 32-bit machine). Unless the clustering job involves more than tens of millions of spectra, the default memory size should do.
- `--sqs <X>` - threshold for quality filtering  $0.0 \leq X \leq 1.0$  (typical value should be  $X=0.05$ ). If this flag is used, the low quality spectra do not get clustered.
- `--fragment-tolerance <X>` - the tolerance in Da for fragment peaks (default  $X=0.34$  Da.) Peaks in the original spectra that are closer than this get merged in the preprocessing.
- `--window <X>` - default  $X=2.0$  Da. This is the window width for clustering. If two spectra have precursors that don't fall in the same window they will not get clustered.
- `--mixture-prob <X>` - the probability wrongfully adding a spectrum to a cluster (default  $X=0.05$ ). A higher value increases the chance of joining two spectra that actually do not belong to the same peptide.
- `--correct-pm` - tries to correct the precursor m/z (uses correct value for clustering, but outputs the original m/z)
- `--peak-density  $\leq X \leq$`  - for spectra preprocessing; the number of peaks to keep in a window of 200 Da (default is  $X=20$ , but higher values might be needed to avoid loss of identifications with certain database search programs).
- `--assign-charges` - tries to assign clusters with charges according to the charges in the spectra files (default assigns charge 0 to all spectra).
- `--min-mz <X>` - the minimal precursor m/z to process (default  $X=0$ )
- `--max-mz <X>` - the maximal precursor m/z to process (default  $X=10000$ )
- `--tmp-dir <path>` - path to where dat files are written (default `./tmp`).

- `--out-dir <path>` - path to where the output files (e.g., \*.mgf, \*.clust, archive or library) are written. The default is to ./out/
- `--dat-only` - only create dat files, do not cluster (performs first and second passes in tandem).
- `--keep-dataset-idx` - for large cluster jobs with pre-processed dat files (see example of PNNL archive below).
- `--dataset-idx <X>` - default X=0, this number is added to file names and cluster titles (to represent the number of times clustering was run on an incrementally increasing archive).
- `--batch-idx <X>` - default X=0, a clustering job can be split to batches run on different m/z values in parallel.
- `--start-file-index <X>` - start numbering the source files from X (useful if splitting dat creation with large clustering jobs).
- `--output-file-size <X>` - number of clusters written per output file (default X=20000).
- `--PTMs <PTM string>` - separated by a colons (no spaces) e.g., M+16:S+80:N+1. Should be used if spectra are annotated with modified peptides.
- `--keep-dat` - do not delete the dat files that were written in the tmp directory.
- `--first-pass` - only perform first pass of dat writing (input should be with `--list`).
- `--second-pass` - only perform the second pass of dat writing (input should be with `--dat-list`).
- `--major-increment` - size of m/z slices used for first pass file generation and output files (default is X=25), value must be in whole Da.
- `--use-input-titles` - use the titles given in the spectra as cluster titles (when possible).
- `--output-mgf` - write the clusters as mgf files (default if `--create-archive` is not used).
- `--convert <type>` - convert the input files given with `--list` to another type (type="dat" or "mgf").

### 3.4 Examples

Create a list of the full paths to the input files and call it list.txt.

- `MSCluter_bin --list list.txt --output-name CLUSTERS` Clusters the spectra, the results will appear in the directory `out` where MS-Cluster is installed (there should be a directory called `mgf` with the mgf file and `clust` with the cluster membership files).
- Adding the flag `--sqz 0.05` will remove many of the low quality spectra (and possibly a very small number of the good ones).
- `--out-dir /home/joe/cluter_results` will send the output files and directories to that directory.

### 3.5 Splitting large jobs

With large jobs you might want to preprocess the data separately (and work in parallel on several nodes). Assume list.txt has 300 files which are split into 3 groups of 100 files each (list1.txt, list2.txt, list3.txt). the preprocessing can be done as follows:

- `MSCluter_bin --list list1.txt --output-name CLUSTERS --batch-idx 0 --file-start-idx 0 --dat-only`
- `MSCluter_bin --list list2.txt --output-name CLUSTERS --batch-idx 1 --file-start-idx 100 --dat-only`
- `MSCluter_bin --list list3.txt --output-name CLUSTERS --batch-idx 2 --file-start-idx 200 --dat-only`

These runs will produce directories of dat files and dat file lists called `CLUSTERS_0_0_dat.list.txt`, ... , `CLUSTERS_0_2_dat.list.txt`. Now these lists can be consolidated into a single list called `CLUSTERS_dat.list.txt` and the clustering job can be run on it.

If the job is large, you might want to split the clustering portion on several nodes too. This should be done by splitting according to  $m/z$  ranges (either by providing dat lists with files from different  $m/z$  ranges) or by using the `--min-mz` and `--max-mz` flags.

## 4 Spectral archives

### 4.1 Overview

Spectral archives are a new clustering-based platform for MS/MS analysis [2]. An archive is basically a clustered dataset that can be added to (by merging archives) and it can also be searched similar to the way a spectrum library is searched.

### 4.2 Creating an archive

To create an archive you need to run with the command line you would like to use for clustering with the addition of the flag `--create-archive`. If running with `--dat-list` you need to add the flag `--list` with the list of the original MS/MS files.

The archive will be created in the output directory. There will be an archive directory (e.g., `ARCHIVE_0_0`) and a main archive file (e.g., `ARCHIVE_0_0_archive.txt`). When running an archive search or merger, this is the file that needs to be supplied.

### 4.3 Merging archives

Archives are designed to be continually grown, such as when new datasets become available. This can be done by merging the archives through clustering the consensus spectra of both archives. So for instance, if we have an archive called `HUMAN`, and we want to add a bunch of datasets from liver samples, we first create an archive called `LIVER` from these samples. Assuming the paths to the archive files are `/archives/HUMAN_0_0_archive.txt` and `/archvies/LIVER_0_0_archive.txt`, we can then merge the two archives with the command: `MSCluster_bin --merge-archives /archives/HUMAN_0_0_archive.txt /archvies/LIVER_0_0_archive.txt --output-name HUMAN` Note that the main archive file (the one we want to add to) is listed first. This command creates a new archive called `HUMAN_1_0` which has clusters from both archives (including new clusters that are created by merging spectra from both archives).

## 4.4 Adding peptide identifications to archives

When generating archives, it might be useful to identify the peptides that generated the spectra for instance using a database search. To do this you need to convert the archive spectra into mgf files using a command like:

```
MSCluster_bin --convert-archive --dat-list /archives/HUMAN_0.0/HUMAN_0.0_dat_list.txt --output-name HUMAN.
```

This command will create mgf files in the output directory for all the consensus files. These files can then be searched by the user and the confident identifications can then be extracted and organized in a text file. Each line in the file represents a single id and should start with the unique id of the cluster. The rest of the columns can be anything you like. The first row lists the column names and the first column should have the name "CLUSTER". For example, an id file can look something like:

CLUSTER	SEQ	CHARGE	SCORE	DELTA	FDR
A_thaliana_0.0.104	LAKPVYEAVR	3	2.610	1.474	0.0040 tr—Q9FL43—Q9FL43_ARATH
A_thaliana_0.0.180	VFDADTK 2	2.491	0.975	0.0180	tr—B3H6B0—B3H6B0_ARATH

When the archive is searched these identifications (the whole lines) can be appended to the search results.

## 4.5 Searching archives

Archive can be searched like spectrum libraries. Query spectra that have high similarity (i.e., a p-value below the threshold are returned in the results file). To run a search you must provide the path to the archive file (with the flag `--search-archive` and the list of query files `--list` and the output name to give the results `--output-name`. Additional optional flags are:

- `--num-results X` - the maximal number of results to return for each query spectrum (default X=10)
- `--max-pvalue X` - the maximal p-value needed of a match to be reported (default X=0.1)
- `--output-matched` - output the matched spectra (as mgf files)
- `--id-list <path>` - give the path to a list of files, each being an id file as described above (each path on a separate line).

## 5 Creating large archives - PNNL case example

We recently generated a large archive from over a billion spectra. The process required over 400 CPU days to complete and started off with TB of data so naturally it was not performed using a single PC but rather using a grid.

One characteristic of the PNNL archive is that it was generated from many organisms. To easily keep track of which organism contributed to each spectrum we decided to designate each organism with its own dataset id. Clustering this data was done by splitting the preprocessing and clustering in to many batches, as described above (Section 3.5). However, rather than incrementally growing the archive by adding each dataset separately (which would have been much more time consuming for over 100 datasets), we imbedded the dataset id into the preprocessing stage using the flag `--dataset-idx` and then when clustering we used the flag `--keep-dataset-idx`.

When all batches were clustered, we created a single archive.txt file for the whole PNNL dataset to enable searching of the archive.

## References

- [1] A.M. Frank, N. Bandeira, Z. Shen, S. Tanner, S.P. Briggs, R.D. Smith, and P.A. Pevzner. Clustering millions of tandem mass spectra. *J. Proteome Res.*, 7:113–122, 2008.

- [2] A.M Frank, M.E. Monroe, A.R. Shah, J.J. Carver, N.F. Bandeira, R.J. Moore, G.R. Anderson, R.D. Smith, and P.A. Pevzner. Spectral archives: A novel approach to analyzing tandem mass spectra. submitted, 2010.