M-SPLIT manual:

To run M-SPLIT:

1. Download spectral library from NIST (or whatever your preferred source), best place is through peptide atlas: <http://www.peptideatlas.org/> (click on SPECTRAL LIBS link on the left panel). M-SPLIT support .msp and .sptxt format.
2. Use SpectraST to generate decoy spectral library. This allows the use of target/decoy strategy to choose a scoring threshold and estimate FDR after search with M-SPLIT. One of the outputs from SpectraST will be a .sptxt format library file. Note M-SPLIT does not require the decoy library, but without the decoy library it cannot estimate FDR. SpectraST: <http://tools.proteomecenter.org/wiki/index.php?title=Software:SpectraST>
3. After the spectral library and decoy library are constructed run M-SPLIT as follow: java –Xmx800M –jar MSPLIT\_v1.0.jar <library file> <query file> <precursor mass tolerance> <outputfile> This will search the library and find the best *pair* of spectra in the library that best matched to the query spectrum

Precursor mass tolerance is in unit Da. Usually one should use a relateive large tolerance like 2Da to allow for the identification of mixture spectra even if query is high accuracy MS data.

After the search, M-SPLIT use a SVM to determine whether a match is significant,

* SVM classification is now done using the svm-light package please go download the binary here:
* <http://svmlight.joachims.org/>
* And put the appropriate binary: svm\_learn and svm\_classify in the svm\_light\_linux or svm\_light\_window folder
* Use the spectrumMatchClassify.pl script to perform the classification as well as estimate FDR by the target/decoy method. Run the script as follow
* ./spectrumMatchClassify.pl <search result file> <filteredOutputFile> <fdr>

FDR is in fraction, i.e. to enforce 1% FDR use 0.01

Output:

Outputs are in tab-delimited format, each column has the following meanings:

We denote M as the query spectrum and A and B as the pair of spectra best matched to M. In case of mixture matches some columns have two values, separated by a “!”.

|  |  |
| --- | --- |
| Column | Content |
| 1 | Query spectrum file |
| 2 | Scan number of query spectrum |
| 3 | Peptide annotation |
| 4 | Protein |
| 5 | Charge of peptide |
| 6 | cosine(M, A+B) |
| 7 | cosine(M, A)[!cosein(M,B)] |
| 8 | cosine(A,B) |
| 9 | alpha (estimated by optimal cosine) |
| 10 | alpha (estimated by residual method) |
| 11 | # of peaks that account of 85% of total intensity |
| 12 | dot-bias(M, A+B) (see paper) |
| 13 | dot-bias(M, A) [! dot-bias(M,B)] |
| 14 | Projected-cosine(M,A+B) |
| 15 | Projected-cosine(M,A) [!projected-cosine(M,B) |
| 16 | mean cosine (of all candidate considered during the search) |
| 17 | mean delta cosine (i.e. cosine(M,A+B)-cosine(M,A)) |
| 18 | Precursor m/z |
| 19 | precursor m/z of A (!precusor m/z of B) |
| 20 | Svm1-score (this score tells whether result is a match) |
| 21 | Svm2-score (this score tells whether result is a mixture match) |

OTHER NOTE:

When you run M-SPLIT on a particular spectral library, please allow sufficiently large amount of memory, this is because M-SPLIT try to load the whole library into memory and created an indexed library that ends in file extension .map. Once this file is created, M-SPLIT uses less memory in subsequent runs. Alternatively you can try to generate the .map library file prior to a search by running the command:

java –Xmx3500M -cp MSPLIT\_v1.0.jar CandidateSpectrumGenerator <spectral library file>