1 Test data set

ST data set no\_digestion.mzXML

use MS-Deconv 0.8.1.412 to deconvolute the mzXML file.

java -cp \*: edu.ucsd.proteomics.msdeconv.console.MsDeconvConsole -o msalign 2006\_08\_16\_ALS\_C4\_40\_lipo\_not\_digested\_30000.mzXML

2 Java results

Download it from uscd web server 0.7.1.7134.zip

Run it with parameter TARGET+DECOY, C57, and 0.01 FDR.

3 Comparison on zero\_ptm\_search

In testcases/zeroptmsearch, run test\_st.sh

Java 0.7 version identified some protein-spectrum-matches missed by C++. The reason is that the recalibration steps in Java 0.7 is not included in C++.

4 Comparison on diagonalfilter

In testcases/diagonalfilter run test\_st.sh

Java 0.7 version identified 7 protein-spectrum-matches missed by C++ (with 1% spectrum level FDR). The reason is that C++ implementation uses a different sorting algorithm that may report different order for SimplePrsms with same scores compared with Java 0.7 version.

5 Comparison on ptmsearch

In testcases/ptmsearch run test\_st.sh

Java 0.7 version identified XXX protein-spectrum-matches with more matched peaks. There are two possible reasons:

(1) JAVA implement use the maximum of matched peak number to refine diagonal shift, but c++ use median to find the best diagonal shift. Sometimes, c++ implementation reports less matched peaks, but the diagonal shift is more reasonable.

(2) In prsm.cpp, a peak is counted once if it matched two or more fragment ions.

(3) Some proteoforms in java identification starts with (M)[42.01], which is not allowed in c++ implementation.

(4) When a proteoform contains two ptms, the refinement methods in Java implementation is different from c++ implementation.