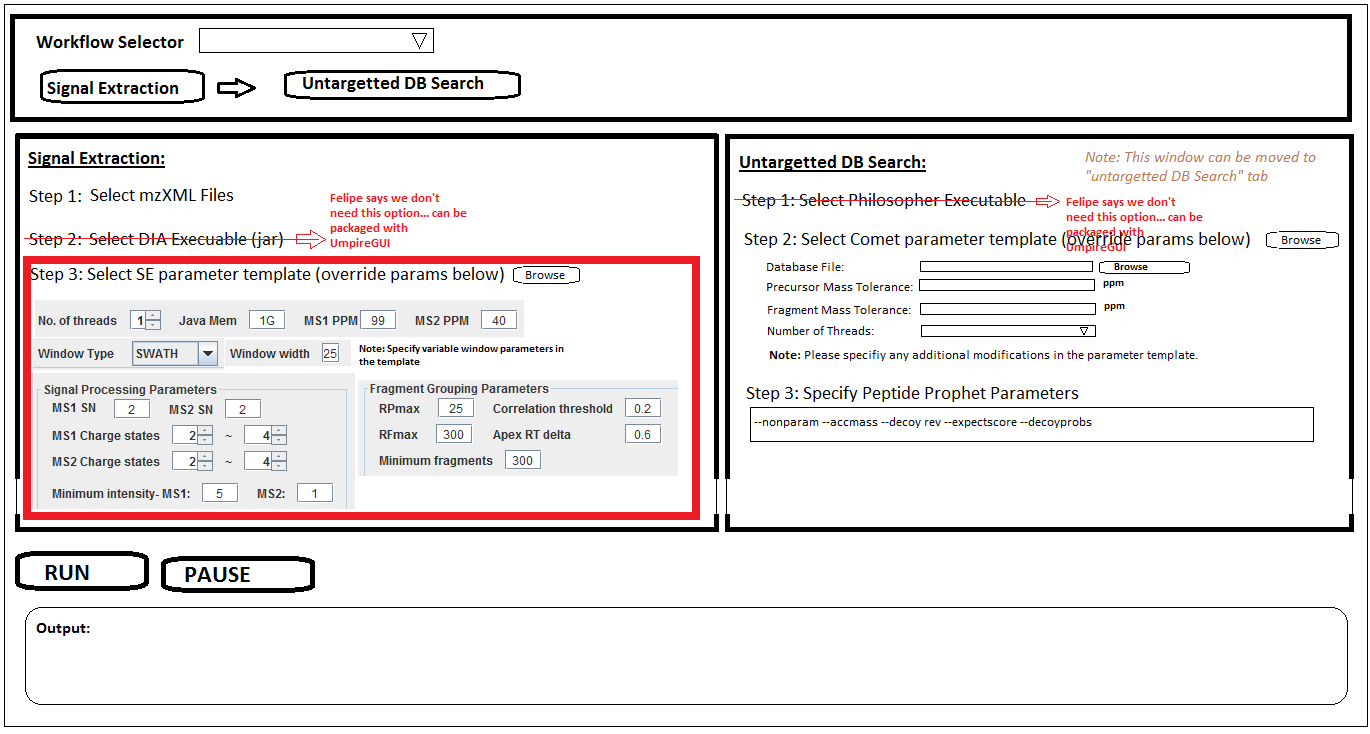
# Scripts (SE Module):

**>>** You need to run the following command on each of the selected mzXML files. The corresponding section on the GUI is highlighted below. I think it is enough to show the parameters in the mockup (for the tutorial, at least). We can check back with Alexey if there needs to be any thing else here. You would generate one parameter file, but run each input (mzXML) file separately.   
**command:** java -d64 -jar -Xmx8G DIA\_Umpire\_SE.jar <mzXMLFile> <umpire-se-params>

**Expected output:** Each command will generate three files (mzXMLfileBaseName\_Q1.mgf, mzXMLfileBaseName\_Q2.mgf, mzXMLfileBaseName\_Q3.mgf)

**E.g. (UPS dataset)**  
java -d64 -jar -Xmx8G E:\nesvi\DIAUmpire\DIA\_Umpire\_SE.jar E:\nesvi\UPS\LongSwath\_UPS1\_1ug\_rep1.mzXML E:\nesvi\UPS\diaumpire\_se.params  
java -d64 -jar -Xmx8G E:\nesvi\DIAUmpire\DIA\_Umpire\_SE.jar E:\nesvi\UPS\LongSwath\_UPS1\_1ug\_rep2.mzXML E:\nesvi\UPS\diaumpire\_se.params



# Scripts (DB Search module):

**>> First,** we need to convert mfg files generated by the SE module to mzXML format. The rest of the pipeline works only on mzXML files. This conversion is done using 'msconvert.exe" (http://proteowizard.sourceforge.net/tools.shtml), an executable that can be shipped along with UmpireGUI. You need to run the following command on each mgf file generated by the SE module. There are *no input parameters* from the user here.   
  
**Command:** msconvert.exe --verbose --32 --zlib --mzXML --outdir <output directory>  <mgf-fileName>

**Expected output:** A (corresponding) mzXML file for each mfg file.

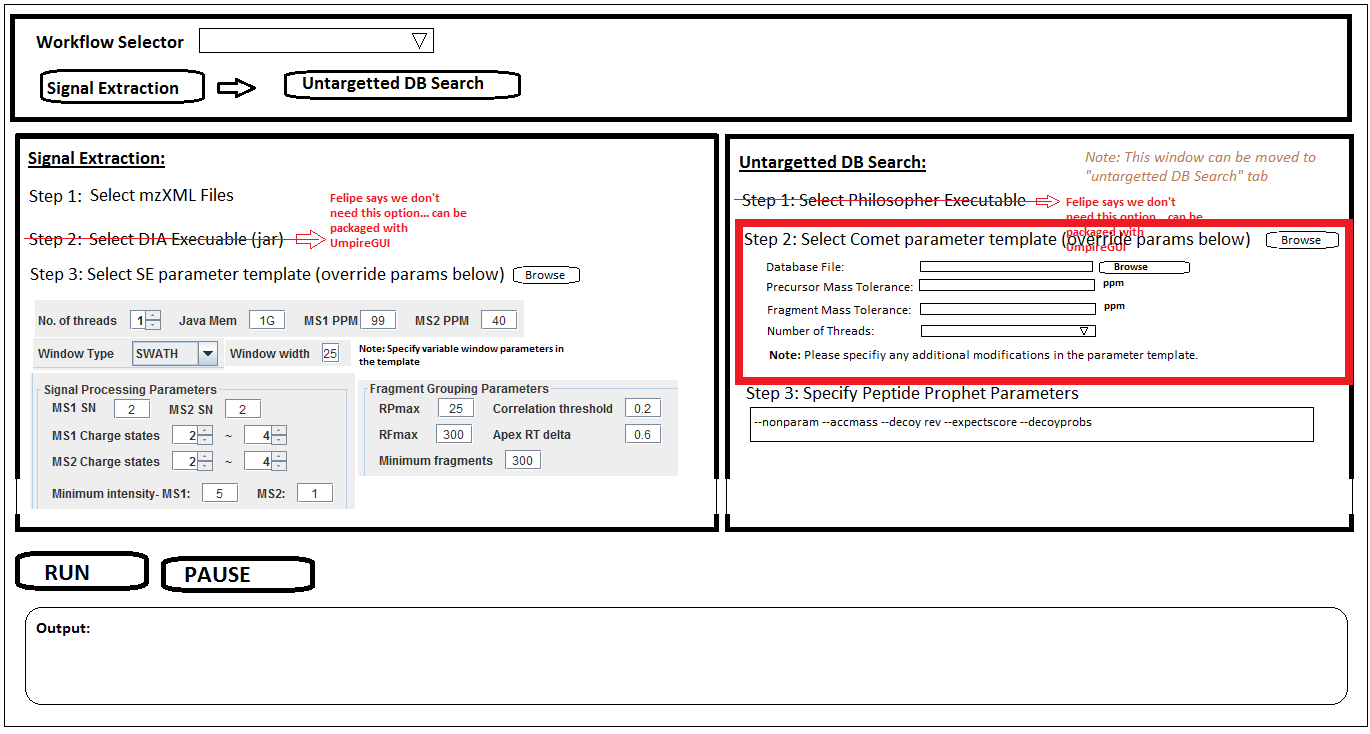
**E.g. (UPS dataset)**msconvert.exe --verbose --32 --zlib --mzXML --outdir E:\nesvi\UPS\  E:\nesvi\UPS\LongSwath\_UPS1\_1ug\_rep1\_Q1.mgf  
msconvert.exe --verbose --32 --zlib --mzXML --outdir E:\nesvi\UPS\  E:\nesvi\UPS\LongSwath\_UPS1\_1ug\_rep1\_Q2.mgf  
msconvert.exe --verbose --32 --zlib --mzXML --outdir E:\nesvi\UPS\  E:\nesvi\UPS\LongSwath\_UPS1\_1ug\_rep1\_Q3.mgf  
msconvert.exe --verbose --32 --zlib --mzXML --outdir E:\nesvi\UPS\  E:\nesvi\UPS\LongSwath\_UPS1\_1ug\_rep2\_Q1.mgf  
msconvert.exe --verbose --32 --zlib --mzXML --outdir E:\nesvi\UPS\  E:\nesvi\UPS\LongSwath\_UPS1\_1ug\_rep2\_Q2.mgf  
msconvert.exe --verbose --32 --zlib --mzXML --outdir E:\nesvi\UPS\  E:\nesvi\UPS\LongSwath\_UPS1\_1ug\_rep2\_Q3.mgf

**>> Second,** run COMET on each mzXML file generated by msConvert. The corresponding section on the GUI is shown below. So you would generate a single comet param file and run each mzXML file with the same params. Again, the params shown on the mockup should be sufficient for the tutorial.

**Command:** philosopher\_windows\_amd64.exe comet --param <comet-params> <mzXML file(s)>

**Expected output:** A  (corresponding) 'pep.xml' file for each mzXML file

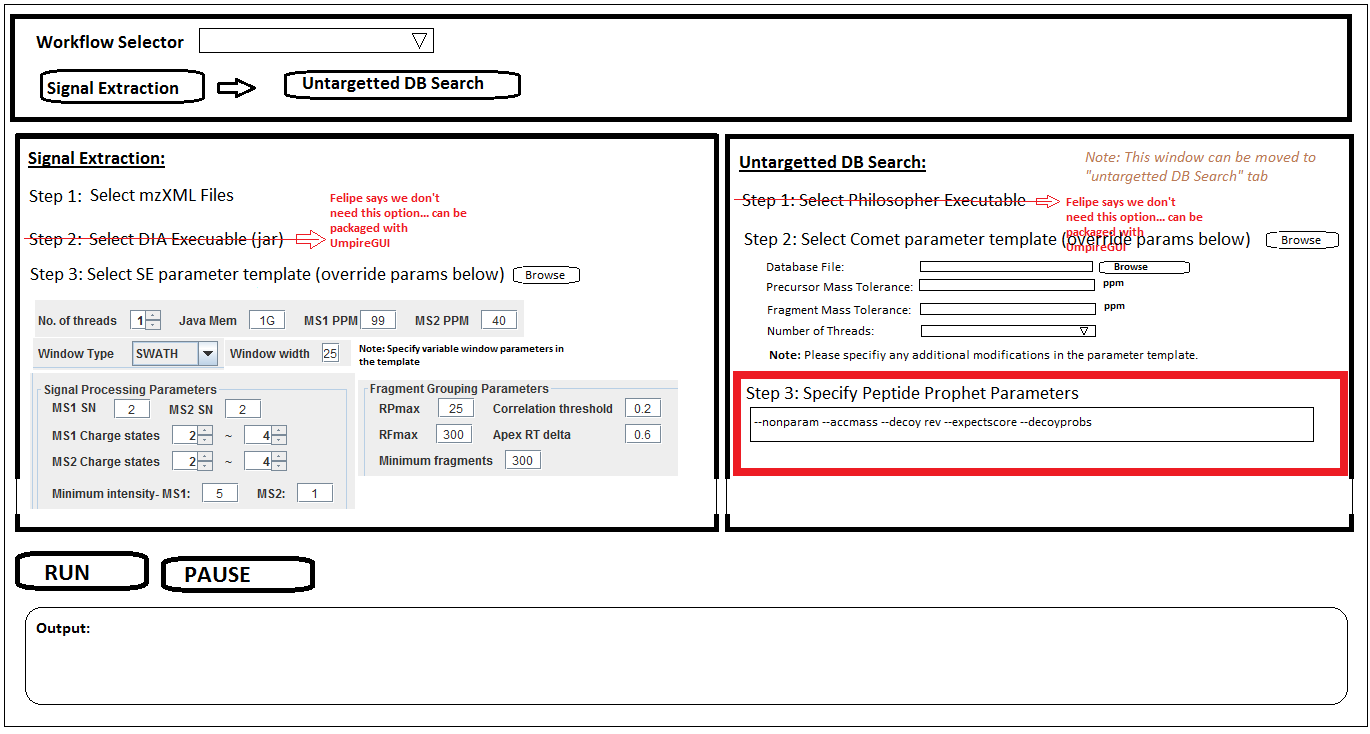
**E.g. (UPS dataset)**  
philosopher\_windows\_amd64.exe comet --param C:\Users\nesvi\Desktop\UPS\comet.params.txt C:\Users\nesvi\Desktop\UPS\\*\_Q\*.mzXML



**>> Third,** run PeptideProphet on each 'pep.xml' file generated by COMET. The corresponding section on the GUI is shown below. The user would just type in parameters as a string into a text box on the GUI (we will provide some default parameter string here).  Note: You’d 're-use" the database file name specified in "Step 2: Select Comet parameter ..." for running the command. Importantly, Felipe's tool automatically iterates over each (comet) pep.xml file, so you just need to run one command here.

**Command:** philosopher\_windows\_amd64.exe peptideprophet <peptideProphetOptions> --database <fastaFile> <list of (comet generated) pep.xml files>

**Expected output:** The command outputs an "interact-xxxx.pep.xml" for each "pep.xml" generated by comet. (Note: Both the outputs of Comet and peptideProphet are called "pep.xml", but their contents are different. In order to avoid confusion, we append a prefix ("interact-") for PeptideProphet generated pep.xml files. Felipe will pass on a new version of Philosopher that can include a prefix).

**E.g. (UPS dataset)**  
philosopher\_windows\_amd64.exe peptideprophet --nonparam --accmass --decoy rev --expectscore --decoyprobs --database C:\Users\nesvi\Desktop\UPS\UPS\_PlusRev.fasta C:\Users\nesvi\Desktop\UPS\\*\_Q\*  
  


# Final output:

The whole pipeline produces three output files (interact-xyz\_Q1.pep.xml, interact-xyz\_Q2.pep.xml, interact-xyz\_Q3.pep.xml) for each input file (xyz.mzXML).