Using moFF to Extract Peptide Ion Intensities from LC-MS experiments

Lennart Martens (lennart..martens@ugent.be)
Department of Medical Protein Research VIB ,Department of Biochemistry, Ghent University
Andrea Argentini
Department of Medical Protein Research VIB ,Department of Biochemistry, Ghent University
Ludger J. E. Goeminne
Department of Medical Protein Research VIB ,Department of Biochemistry, Ghent University,Department of Applied Mathematics, Computer Science and Statistics, Ghent University,
Kenneth Verheggen
Department of Medical Protein Research VIB ,Department of Biochemistry, Ghent University
Niels Hulstaert
Department of Medical Protein Research VIB ,Department of Biochemistry, Ghent University
An Staes
Department of Medical Protein Research VIB ,Department of Biochemistry, Ghent University
Lieven Clement
Department of Applied Mathematics, Computer Science and Statistics, Ghent University,

Method Article

Keywords: MS1 intensities, proteomics, LC-MS, label-free quantification

Posted Date: November 29th, 2016

DOI: https://doi.org/10.1038/protex.2016.085

Abstract

Label free quantification approaches based on MS1 intensities extract directly from the raw file have become really popular due to low cost and the reliability of the result. Quantification softwares like MaxQuant provide accurate MS1 intensities but they need demanding computational steps that also limit their integration in automated pipeline for large number of LC-MS experiments. This protocols shows how to use moFF \(modest Feature Finder), a scriptable and operating system independent software to for extracting peak intensities from Thermo raw files using an apex approach and match-between-runs functionality. This protocol shows also the use of the command line and the graphic-user interface version of moFF \(https://github.com/compomics/moff-gui)

Introduction

Quantitative mass spectrometry \(MS) based proteomics aims to quantify all proteins in a sample[1]. Quantitative approaches fall in two main groups: labelled and label free. In labelled approaches the quantification is based on the labelling of the peptides using an isotopic or isobaric mass tag. Label-free approaches, does not require these additional costs for sample preparation and can be performed on unlimited number of samples. The most accurate label-free quantification methods are based on MS1 signals, extracting peptide intensities by finding the best peak in the three relevant dimensions \(m/z, retention time, intensity). The associated workflow consist in the feature detection and the feature alignment[2]. A feature is a triplet composed by the mass-over-charge \(m/z), RT and intensity founded in the raw data. In the feature combination step, features that belongs to the same peptides are grouped in cluster where the m/z values correspond to the isotopic masses of a peptides and the RT time interval correspond to the elution profile of the peptides. The intensity of possible peptide \(a cluster of features) is the sum of all the peaks in the retention time interval identified. The feature alignment \(called “match-between-runs”) is intended to match features across runs that lack identified fragment spectra in some of the runs. MaxQuant[3] is the most popular software for protein quantification, it detects features by fitting a Gaussian peak shape to the three relevant dimensions \(intensity, RT, and m/z) and then estimates peptide intensity as the volume of this complex 3D feature. Despite the precise intensities computed, MaxQuant suffers of speed penalties when the size of the dataset is increased and of a lack of integration in own pipeline. The increasing size also the complexity of the proteomics data in public repository \(ProteomeXchange[4]) and their re-analysis has been shown to be promising for novel discovery[5]. To face this new challenge there is a need of quantification tool fast reliable and cluster friendly that can scale with the increasing size of complex quantitative data sets present in public proteomics repositories. **moFF Overview** moFF \(modest Feature Finder) is a simple, fast and operating system independent MS1-based relative quantification algorithm. moFF is based on python and works directly on Thermo raw file and mzML as well. The access to Thermo raw file is based on the unthermo raw file library [6] that allow moFF to work both on Linux and Windows system. The access to mzML files is based on the python library pymzML[7]. moFF consists in two modules : the match-between-run and the apex extraction module. The complete workflow is showed in Figure 1 See figure in Figures

section. As input, moFF needs a list of identified features \(e.g the result of Mascot or X\!Tandem) where each feature should be characterized by a minimum set of information. The match-between-runs module \(mbr) performs a RT alignment across the runs, in order to match undefined features that are identified in other run. This process increase the number of quantified feature across the replicates and reduces the missing values in the MS1 intensity matrix used in further analysis. Both matched and identified feature are then processed by the apex module where the apex peaks are extracted directly from their XiC retrieved from the raw files \(see Figure 2). See figure in Figures section. moFF provides two quality measures of the peak extracted: - Shape of the peak \(**log_L_R**): if the peak has a symmetrical shape the value will be around 0, otherwise for left or right skewed shape the value is respectively greater or less than 0 -Signal-to-noise \(**SNR**): this measure how the apex intensity is higher with respect to the level of the noise presence in the XiC extracted. The parameters of moFF are the following: - The size of XiC windows retrieved for each feature. - The retention time \(RT) window used to search the apex. - The precursor mass tolerance The match-between-run has also other parameters: - The retention time \(RT) windows used to search the apex intensity for the matched peak - Outlier filter and its width value. This filter works on the training set used to train the RT predicted models - Weighted or an unweighted combination of the predicted retention time model when a features is matched in several runs.

Equipment

Computer - Operating system: Windows or Linux - Python 2.7 installed and also Java 1.7 for moFF-GUI - Download "moFF-GUI":https://github.com/compomics/moff-gui or "moFF":https://github.com/compomics/moFF **Input data** - Raw file: Themo raw file of mzML file - Identified features listed in a tab delimited file. The minimum information required for each feature are: _peptide_: sequence of the peptide _prot_: protein ID _rt_: feature retention time \(The retention time must be specified in second) _mz_: mass over charge _mass_: mass of the feature _charge_: charge of the ionized feature - PeptideShaker cps files along with the sequence database \(FASTA) and spectra \(MGF) used in SearchGui \(only for moFF-GUI)

Procedure

moFF from the command line with identified features in tab-delimited file as input 1. Put your input identified features files in _input_. Put your raw file in another folder a called _rawFolder_. 2. Run moFF \(match-between-runs and apex) using the following command: **python moff_all.py –inputF input/ –raw_repo rawFolder/ –output_folder my_output ** To set all the parameters and options of moFF, please read the full list in the "documentation":https://github.com/compomics/moFF/blob/master/README.md#entire-workflow 3. Collect all the results in the output folder **moFF-GUI with PeptideShaker result as input** 1. Run moFF-GUI and set the folder where PeptideShaker is installed and the output folder where all the results are collected. Click _proceed_ to continue 2. Choose which module to run. 'Apex' is just the apex MS1 extraction, and 'matching-between-run' for the match-between-runs module plus the apex module. Click

proceed to continue \(Figure 3). See figure in Figures section. 3. Insert the Thermo RawFile or mzML files. For each raw file you have to associate also the relative cpsx file. Moreover, you can also insert the fasta and the mgf files used in PeptideShaker/SearchGui. Click _proceed_ to continue \(Figure 4). See figure in Figures section. 4. Setting of the moFF parameters \(Figure 5): - XiC retention window - Peak retention time windows - Precursor mass tolerance - Match-between-run parameters_: -Peak retention time windows for matched peak - Weighting/unweighting and the activation of outlier filtering and its width value - Selection of a set of specific peptides \(loaded as tab-delimited file) and use them as training set of the mbr procedure instead of the shared features of the runs. See figure in Figures section. 5 Start the procedure clicking on _start_. Collect your result in the output folder

Timing

The time taken largely depend by the number of input feature and by the length of the XiC extracted \(XiC retention window).

Anticipated Results

moFF produces for each run/raw file a file with the results and a log file with detailed information about the apex intensity extraction \(see Figure 6). See figure in Figures section. The output of match-between-run module is the set of the input files enriched with the matched features founded and a separated log file that contains all the detail of procedure.

References

1.Vaudel, M., Sickmann, A. & Martens, L. Peptide and protein quantification: a map of the minefield. Proteomics 10, 650–70 \(2010) 2.Sandin, M., Teleman, J., Malmström, J. & Levander, F. Data processing methods and quality control strategies for label-free LC-MS protein quantification. Biochim. Biophys. Acta 1844, 29–41 \(2014). 3.Cox, J. & Mann, M. MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. Nat Biotech 26, 1367–1372 \(2008) 4.Vizcaíno, J. A. et al. ProteomeXchange provides globally coordinated proteomics data submission and dissemination. Nat. Biotechnol. 32, 223–6 \(2014) 5. Vaudel, M. et al. Exploring the potential of public proteomics data. Proteomics 16, 214–25 \(2016) 6.Kelchtermans, P. et al. An open source, platform-independent library and online scripting environment for accessing Thermo Scientific RAW files. J. Proteome Res. \(2015). doi:10.1021/acs.jproteome.5b00778 7. Bald, T. et al. pymzML–Python module for high-throughput bioinformatics on mass spectrometry data. Bioinformatics 28, 1052–3 \(2012).

Figures

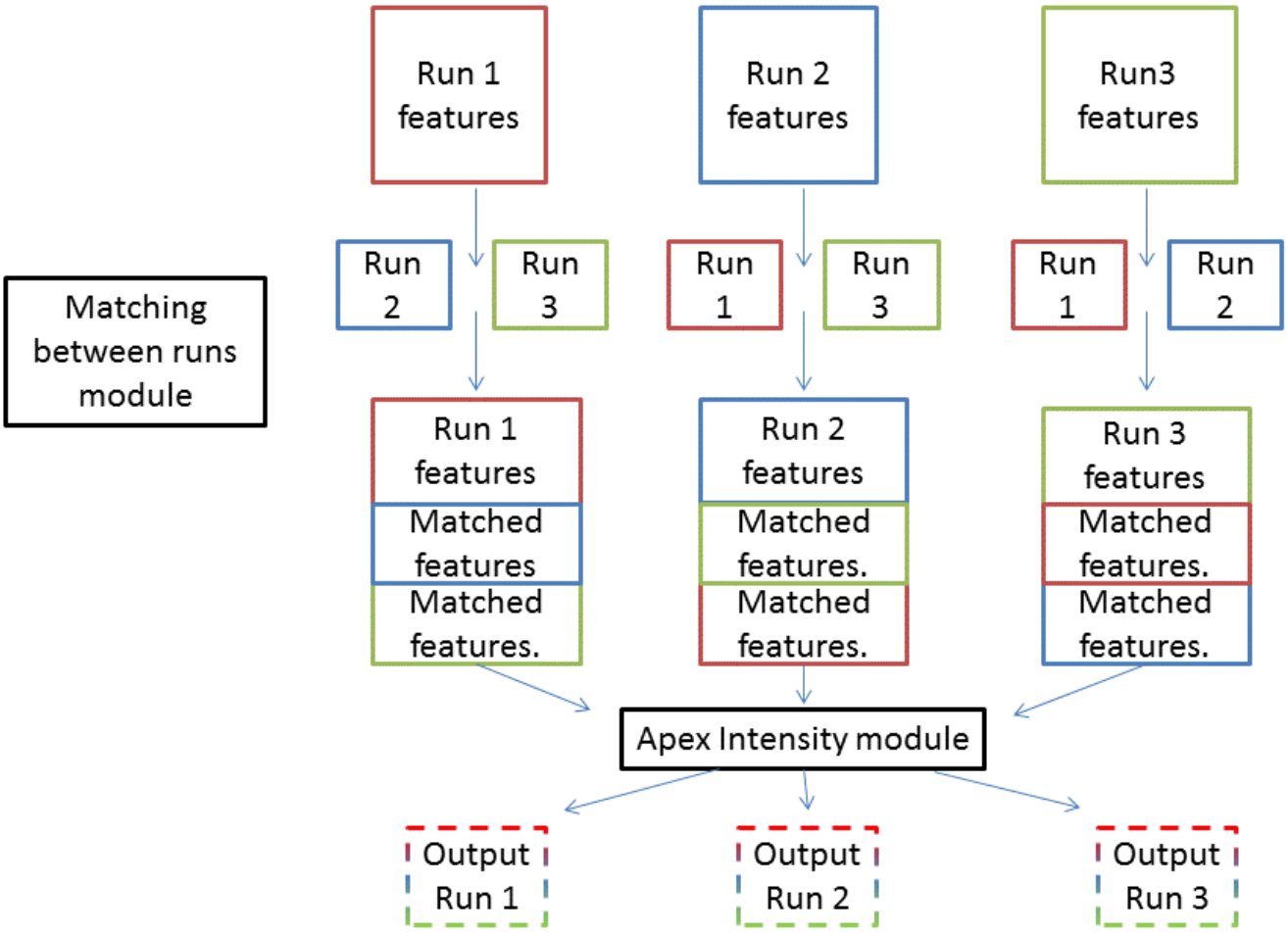

Figure 1

Figure 1 moFF workflow overview. The optional match-between-runs module takes as input a list of features of interest for a given run, and then matches the corresponding features in other runs. Upon matching, all peptides (either identified directly, or indirectly by matching between runs) are processed by the Apex intensity module to extract feature intensity and associated quality measures. The results are then written to tab-delimited output files.

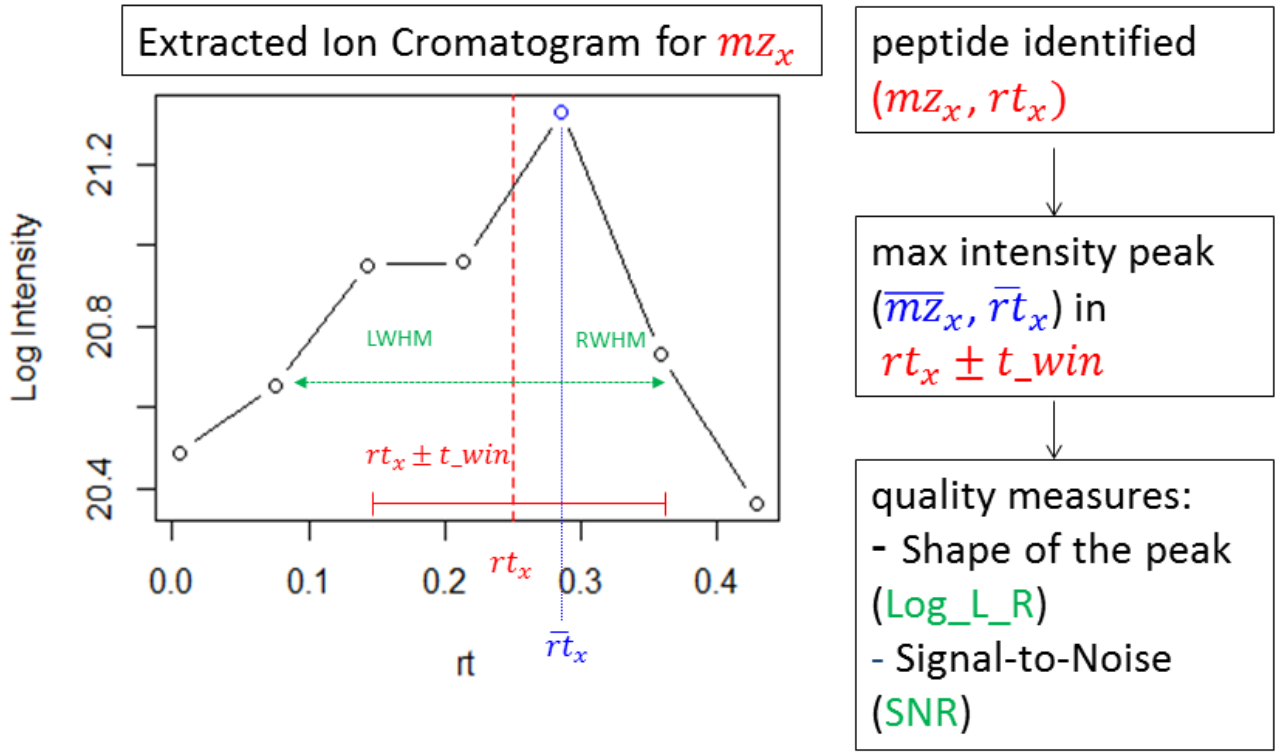

Figure 2

Overview of the apex intensity module Around the RT of the input feature, a time window is constructed which results in a local XIC. The peak apex is then located in this local XIC. The log_L_R metric measures the skewness of the peak around the RT of the obtained apex point. The SNR metric provides the ratio of peak height to noise, where the noise value is set as the lowest intensity value in the local XIC

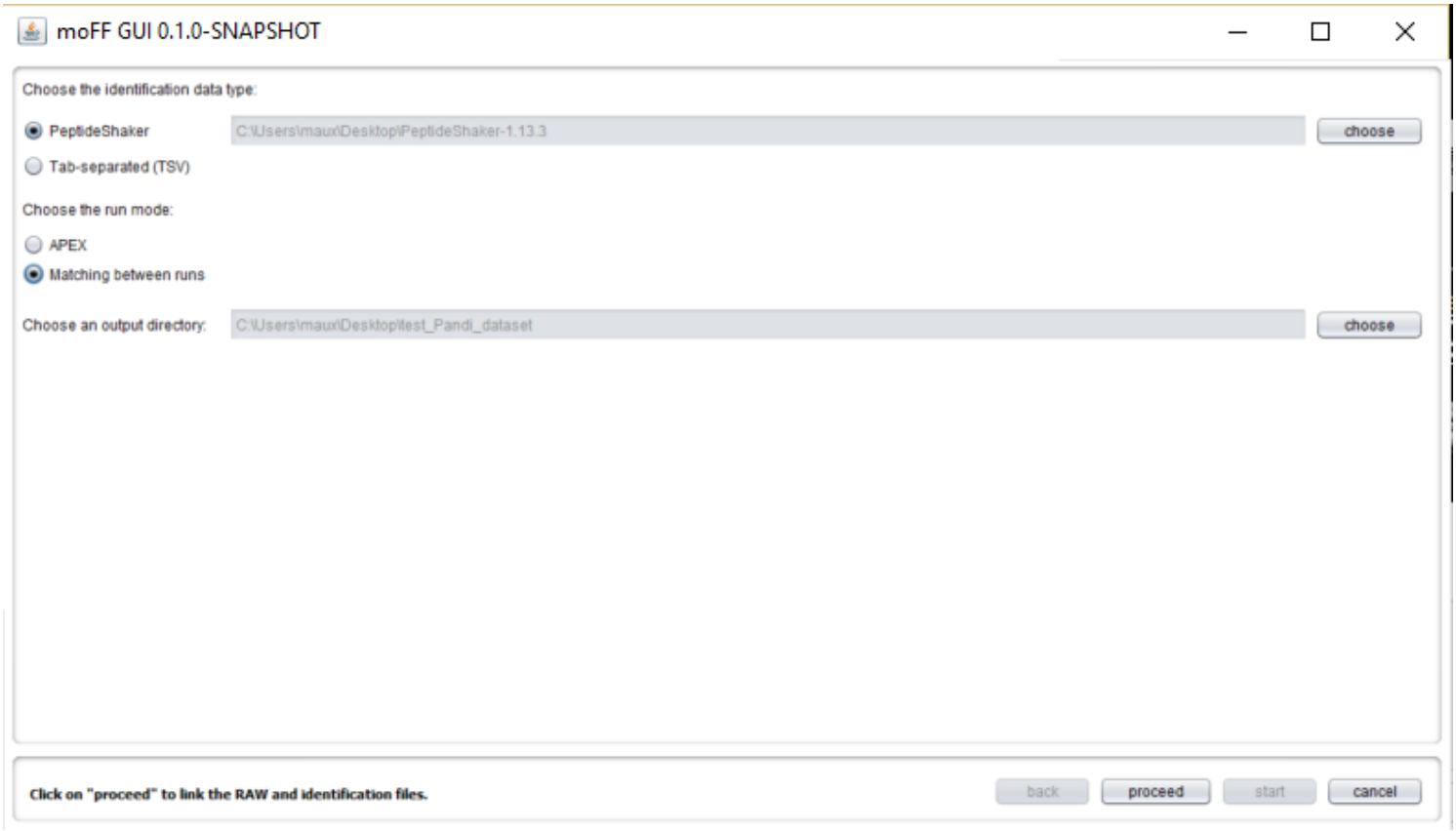

Figure 3

moFF GUI Overview Setting the PeptideShaker folder and the moFF output folder where to collect your results.

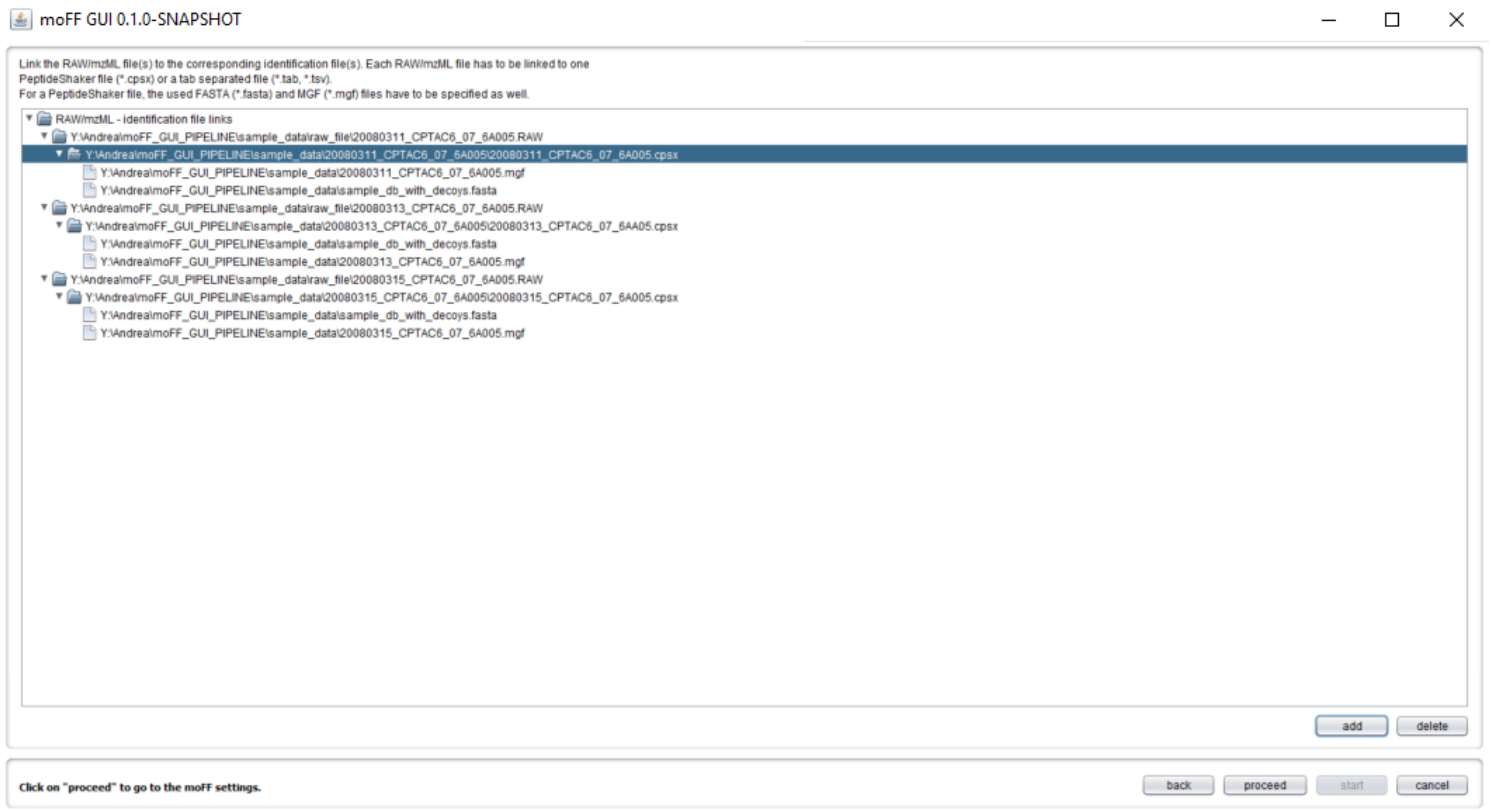

Figure 4

moFF GUI Overview (2) Load the raw and cps files into moFF-GUI

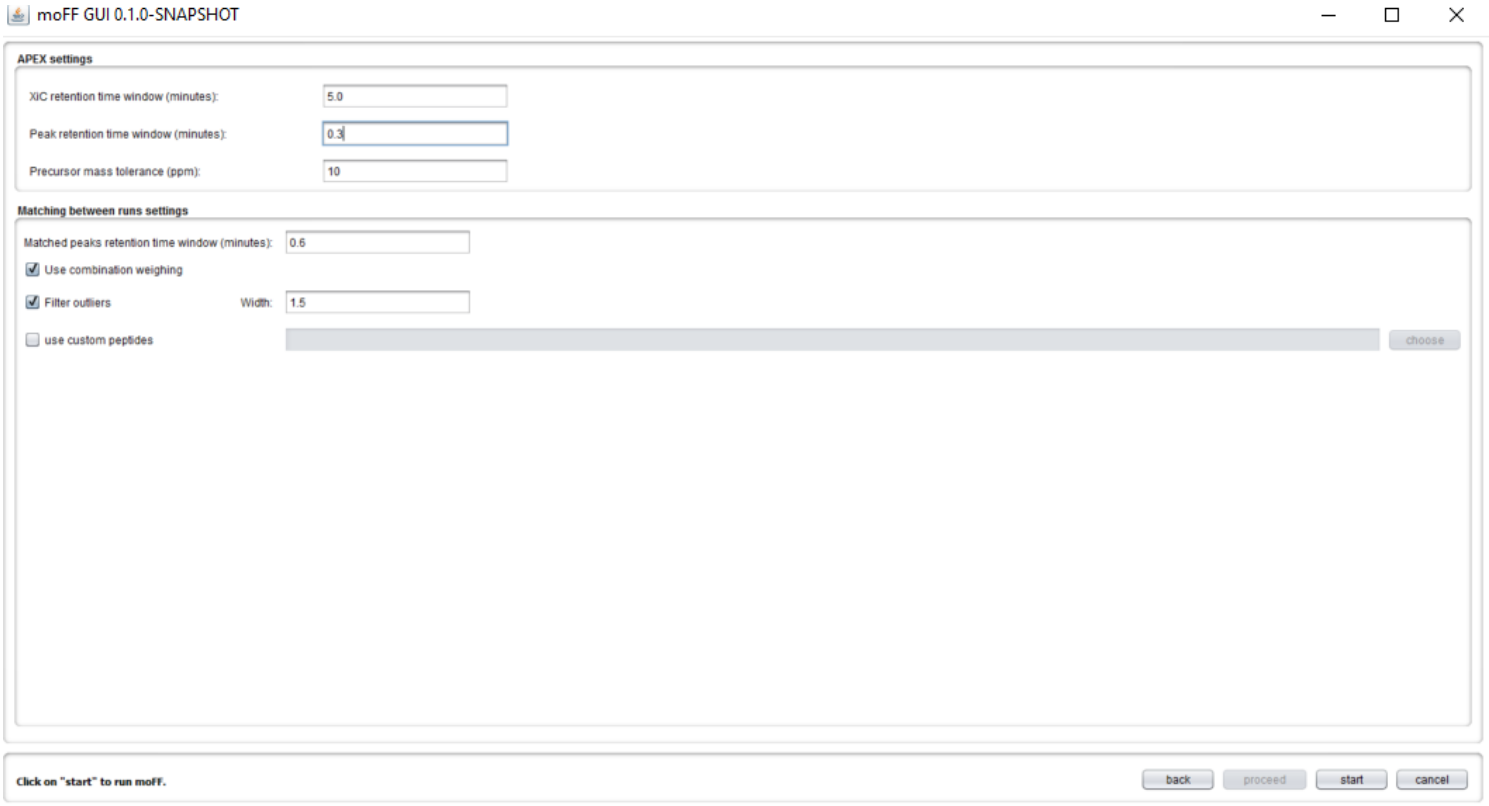

Figure 5

moFF GUI Overview (3) Setting the moFF parameters

Figure 6

output files of the apex module The output of the apex module is composed by two files: the result file and log file. In the result file , you notice the intensity and log intensity extracted by moFF along with the quality measures log_L_R and SNR. The log file contains details information to each apex intensity extracted. Here you can easily spot spot eventually errors in the apex intensity extraction from the XiC data