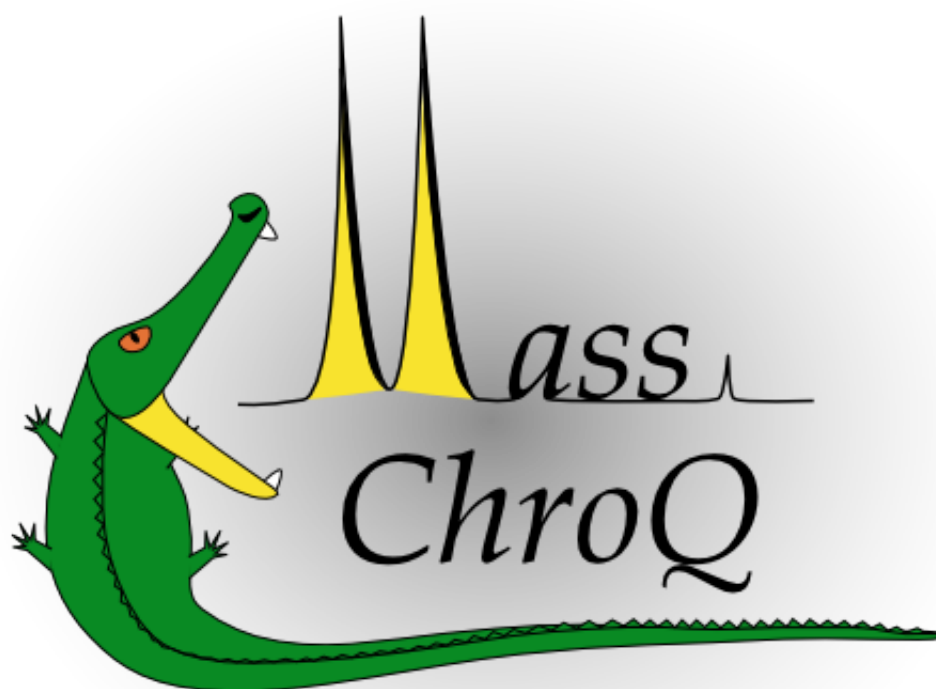


MassChroQ manual

Mass Chromatogram Quantification software





PLATEFORME D'ANALYSE PROTÉOMIQUE DE PARIS SUD-OUEST

MassChroQ manual

First edition for MassChroQ 1.0 *Hungry Crocklet*

Author: Edlira NANO

Contributors: Olivier LANGELLA, Benoît VALOT, Michel ZIVY

Copyright ©2010–2011 B. Valot, O.Langella, E.Nano, M.Zivy

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation Licence Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the appendix [C](#) entitled "GNU Free Documentation License".

Contents

Preface	vii
1 Introduction	1
1.1 What is MassChroQ?	1
1.2 MassChroQ features overview	2
1.3 Help	3
2 Installing and running MassChroQ	5
2.1 Installation	5
2.1.1 Linux platforms (32 and 64 bytes)	5
2.1.2 Windows platforms	6
2.1.3 SVN repository	6
2.2 Running MassChroQ	7
2.2.1 MassChroQ's XML input file	7
2.2.2 Peptide identification file parsing option	8
2.2.3 Temporary working directory option	9
3 How MassChroQ works	11
3.1 Quantified items	11
3.1.1 Identified peptides	11
3.1.2 Identified isotopes	12
3.2 Parsing of LC-MS/MS files in mzXML or mzML formats	12
3.3 Grouping of LC-MS runs	13
3.4 XIC extraction	13
3.5 XIC filtering	14
3.6 Peak detection	14
3.6.1 The Moulon peak detection algorithm	15
3.6.2 The Zivy peak detection algorithm	15
3.7 Alignment	20
3.7.1 The OBI-Warp alignment method	20
3.7.2 The MS/MS alignment method	21

3.7.3	Previous alignments and cascade alignments	24
3.8	Peak matching	24
3.8.1	The best RT method	24
3.8.2	Smart Quantification	25
4	The masschroqML format	27
4.1	Data files	27
4.2	Groups	28
4.3	Peptide text files	29
4.4	Identified peptides and proteins	30
4.5	Isotope labels	32
4.6	Alignments	33
4.7	Quantification methods	34
4.8	Quantifying	36
4.8.1	Quantifying peptides	36
4.8.2	Quantifying isotopes	37
4.8.3	Quantifying m/z values	37
4.8.4	Quantifying $(m/z, rt)$ values	37
4.9	Result files	38
4.10	Trace files	40
5	Specification of the identified peptides files	43
5.1	masschroqML peptide files instructions	43
5.1.1	masschroq command-line <code>-parse-peptides</code> option	43
5.1.2	Peptide text file format specification	44
6	Parameters cheat-sheet	47
6.1	Alignment parameters	47
6.2	Quantification parameters	48
	Appendices	53
A	masschroqML complete input example file	53
B	Peptide identification example tsv file	57
C	GNU Free Documentation License	59
1.	APPLICABILITY AND DEFINITIONS	59
2.	VERBATIM COPYING	61
3.	COPYING IN QUANTITY	62
4.	MODIFICATIONS	62
5.	COMBINING DOCUMENTS	64

6. COLLECTIONS OF DOCUMENTS	65
7. AGGREGATION WITH INDEPENDENT WORKS	65
8. TRANSLATION	66
9. TERMINATION	66
10. FUTURE REVISIONS OF THIS LICENSE	66
11. RELICENSING	67

Preface

MassChroQ is an open-source software released under the [Gnu General Public Licence version 3](#). It is developed at the [PAPPSO](#) team (the Paris South-West Proteomics Analysis Platform) by:

Benoît Valot benoit.valot@moulon.inra.fr

Olivier Langella olivier.langella@moulon.inra.fr

Edlira Nano edlira.nano@moulon.inra.fr

Michel Zivy michel.zivy@moulon.inra.fr

This manual contains detailed information about the concepts and features of MassChroQ v.1.0. It is intended to provide all the technical information needed for an advanced comprehension and use.

In chapter [1](#) you will be introduced to the main features of MassChroQ.

In chapter [2](#) you will learn how to install and run MassChroQ with a quick overview of its available command-line options.

In chapter [3](#) you will learn how MassChroQ works in depth: every analysis step is detailed and the most important algorithms (peak detection, alignment) are explained step-by-step.

In chapter [4](#) the input xml file to MassChroQ is explained using an example file which we explain line by line.

In chapter [5](#) you will find the complete specification of the identified peptides CSV file format.

In chapter [6](#) you will find a cheat-sheet of all the parameters in MassChroQ with corresponding recommended values.

Finally in the appendixes you will find the complete files used in this manual.

Chapter 1

Introduction

1.1 What is MassChroQ?

MassChroQ (Mass Chromatogram Quantification) software performs alignment, XIC extraction, peak detection and quantification on data obtained from LC-MS (Liquid Chromatography-Mass Spectrometry) techniques.

A full description and evaluation of MassChroQ can be found in [\[VLNZ11\]](#).

MassChroQ can analyze:

- label-free data as well as isotopic labeled ones (e.g. SILAC, ICAT N-terminal, C-terminal labels);
- data obtained from high-resolution systems (HR) as well as low-resolution ones (LR).
- It is able to take into account complex peptide or protein experiments on data (e.g. peptide or protein separations prior to LC with SDS-PAGE, SCX fractionations, etc.).
- It is fully configurable and every single step of its analysis is fully traceable.
- Its modular implementation facilitates integration of third-part libraries as well as MassChroQ's integration into them.
- It is platform-independent and uses or produces only open format data.

MassChroQ is developed in the C++ language using the Qt framework. Version 1.0 is its first public release.

MassChroQ comes as a stand-alone command-line program and also with a library for integration in other softwares or proteomic pipelines.

On the MassChroQ homepage at <http://pappso.inra.fr/bioinfo/masschroq/> you can find download instructions, various documentation files and the latest news about this project.

On the MassChroQ development page hosted by SourceSup at <http://sourcesup.cru.fr/projects/masschroq/> you will find a subversion repository, a bug tracker and various forums. The source code is anonymously available via direct access to the subversion repository from <https://subversion.cru.fr/masschroq/>.

Feel free to contribute to the MassChroQ project by directly contacting one of its authors.

1.2 MassChroQ features overview

MassChroQ has been designed to perform quantification on a wide range of LC-MS data: label-free or isotopic labeled ones, high-resolution (HR) or low-resolution (LR) ones. To achieve this MassChroQ can combine and perform the following features :

- Determination of items of interest to be quantified. These items can be:
 - the identified peptides,
 - the identified isotopes,
 - a list of mass over charge (m/z) ratios,
 - a list of couples of m/z and retention time (rt) values.
- Alignment of samples within each group (two different alignment methods are proposed).
- Extraction of the XIC-s (Extracted Ion Chromatogram) of the predetermined items of interest.
- XIC filtering (several filters are provided for signal noise removal, spike removal, signal smoothing, etc.).
- Detection of peaks on these XICs.
- Quantification of the predefined items of interest (two different quantification methods are proposed).
- Grouping of LC-MS data that present similarities (for example grouping of the same LC fractions in an SCX fractionated analysis in order to perform alignment on them).

MassChroQ uses the notion of *groups* of LC-MS data according to their technical similarities. Grouping affects alignments: all runs from the same group will be aligned with the same method; and quantification: peak detection and quantification will be performed in all runs of the same group for peptides that were identified in at least one run of this group.

Groups also give the user the possibility to perform specialized analysis on several different sets of data in one shot. For example, in a peptide SCX separation experiment, only the samples of the same LC fraction should be aligned to each other. In that case, we form a group of these samples in MassChroQ and we assign the desired alignment method to it.

We can do the same with quantification methods: suppose we have a set of runs obtained with an HR Orbitrap spectrometer (which is known for producing artifact signal spikes) and another set of runs obtained with an LR LTQ spectrometer (which produces a certain baseline noise but no spikes). We can group the Orbitrap runs together and put the LTQ ones in another group. We then apply a quantification method containing an anti-spike XIC filter to the first group, and another quantification method containing a background filter to the latter and tell masschroq to perform analysis in both groups in one shot.

MassChroQ accepts mzXML as well as mzML LC-MS data formats.

To include the identified peptides/isotopes in a MassChroQ analysis there are two possibilities :

- by using the freely available open-source tool *X!Tandem pipeline* developed at our team;
- by directly providing to MassChroQ spreadsheet text files (*tsv* or *csv*) that contain the identified peptides for each sample (see section 5 for details on the peptides files format);

MassChroQ offers two alignment methods: the **OBI-Warp** alignment method which uses MS level one retention times only and the MS2 alignment method which uses MS level 2 retention times.

The quantification results in MassChroQ are summarized in a single file, sorted by group and sample, allowing comparisons to be performed easily. Several results file formats are available: gnumeric, TSV, xhtml and XML (masschroqML XML format).

1.3 Help

On the MassChroQ homepage (<http://pappso.inra.fr/bioinfo/masschroq/>) you can find :

- a [FAQ](#);
- masschroq's [manpage](#);
- masschroq's [schema](#);
- Dataset examples of masschroqML input files to MassChroQ for various ordinary situations (fractionated sample, isotopic labeled ones, etc);
- this user manual frequently updated;
- the code documentation (generated by doxygen);
- the latest news and the upcoming features on this project;
- BibTeX and text entries for MassChroQ citation.

On the MassChroQ project page hosted on [SourceSup](#) you can find :

- a [subversion repository](#);
- a bug tracker;
- several user and developer forums.

Chapter 2

Installing and running MassChroQ

2.1 Installation

2.1.1 Linux platforms (32 and 64 bytes)

- Debian and Ubuntu: Precompiled binary packages for MassChroQ *Hungry Crocklet* version 1.0 are available for [32-bit](#) and [64-bit](#) systems. Ubuntu's software center automatically installs the package and the dependencies by double-clicking on it.
- Other distributions: to build `masschroq` on Linux you need:
 - Qt 4.5.2 or higher. Most Linux distributions have packages available. In any case, be sure to get the `-dev` package for Qt in addition to any other libraries.
 - Cmake 2.6 or higher.

Archives for 32 and 64 bytes systems containing the source code and pre-compiled Linux binaries for MassChroQ release version 1.0 are available from <http://pappso.inra.fr/bioinfo/masschroq/>. Decompress the archive and if necessary rebuild the binaries with the commands :

```
cd path_to_masschroq_archive/masschroq-1.0
cmake .
make
sudo make install
```

This will install `masschroq` and its schema into your system; you will then be able to run the `masschroq` command from your console. You will also have access to `masschroq`'s manpage by typing `man masschroq`.

2.1.2 Windows platforms

A win32 setup installer named `masschroq_setup.exe` is available for download at: <http://pappso.inra.fr/bioinfo/masschroq/>. This setup installs MassChroQ in the chosen directory of your system together with a set of example files that you can immediately try by double-clicking on the `masschroqML` (MassChroQ input files) they contain.

For a more advanced use of MassChroQ on a Windows console:

- click on *start* → *Run*;
- type `cmd` on the Run window that appeared, and then press OK.

A console command-line window appears. To run `masschroq`, you type in it:


```
masschroq path_to_input_file\input_file.masschroqML
```

where `input_file.masschroqML` is the input file of your analysis. To use the `masschroq` options type:

```
masschroq --help
```

on the command-line console.

Please refer to the `masschroq_readme.pdf` file that comes with your Windows installation for further details.

 The *masschroqML* format is an XML format which in Windows are by default opened with Internet Explorer, and are impossible to edit. When you need to edit these files, we strongly recommend you to open them with text editors (other than Notepad, try it, you will see why), for example the free *Notepad++* editor which offers syntax highlighting with a nice look. And if you cannot stand Windows problems anymore, we recommend you the latest Ubuntu Desktop edition (all the advantages of Linux but nice and easy graphical use).

2.1.3 SVN repository

The subversion repository located at <https://subversion.cru.fr/masschroq/> is available read-only to the public at large. Assuming you have at least version 1.0.0 of [Subversion](#) installed, you can checkout MassChroQ sources, including latest developments from `trunk` by using the following command :

```
svn checkout https://subversion.cru.fr/masschroq/ my_masschroq_directory
```

2.2 Running MassChroQ

The command-line executable/binary of MassChroQ is called `masschroq`. Once installation is completed you can type `masschroq --help` on the command-line to get the list of the available options and their usage.

On Linux platforms a manpage for `masschroq` is automatically installed with the program.

2.2.1 MassChroQ's XML input file

The most important input parameter for `masschroq` is an XML file. This is where the user defines the LC-MS data to be analyzed and all the different treatments and parameters to be performed on them. This input XML file follows the *masschroqML* format, whose annotated schema can be found on the MassChroQ [homepage](#) and also in the `doc/schema/` directory of your installation directory.

A `masschroqML` input file can be generated automatically from this schema by any XML editor. You can of course use a standard text editor to manually write such a file, MassChroQ performs automatic AI verification of these files towards its schema and will guide you to the line where syntax errors have been introduced if any.

You can also use the `masschroqML` example templates we provide on MassChroQ's website and adjust them to your analysis. These templates together with the *masschroq_complete_input_example.xml* of appendix A can also be found in the `doc/xml_examples/` directory of your MassChroQ installation directory. They are also available on the MassChroQ homepage.

To produce a `masschroqML` input file that also includes all the identified peptides of your data, you can use the freely-available *X!Tandem pipeline*. Given a set of LC-MS runs, this pipeline performs in one shot peptide identification using the *X!Tandem* software, filtering of the identification results and export to several formats, one of them being a ready-to-use `masschroqML` input file.

If you want to perform identification on your data by using other engines, you should use `masschroq`'s `-parse-peptides` option to integrate these identified peptides in your `masschroqML` input file. For this, you will have to provide one peptide text file per data. For details on how to use this option see section 2.2.2. For details on the peptide text file format see section 5.

In section 4 the `masschroqML` format and the precise impact of every parameter it contains is explained in details.

2.2.2 Peptide identification file parsing option

The *X!Tandem pipeline* way of processing your LC-MS data has a double advantage :


- the identification, filtering and quantification processes are automated and linked through an intuitive graphical user interface;
- X!Tandem and the *X!Tandem pipeline* are open-source software, freely available, very reliable and easy to install.

However, if you do not need to use the full pipeline, you can provide your own spreadsheet text files containing the identified peptides for each sample. These text files can be in *tsv* or *csv* format (tabulation, comma or semi-colon separated values). One peptide text file per data has to be provided. The `-parse-peptides` command-line option makes MassChroQ parse them, combine the same peptides appearing in several data and integrate them in the XML input file.

To achieve this you will have to put in your masschroqML input file the references to the peptide text files to be parsed in the following way :

```
<masschroq>
...
<peptide_files_list>
<peptide_file data="sample0" path="peptides_sample0.txt"/>
<peptide_file data="sample1" path="peptides_sample1.txt"/>
</peptide_files_list>
...
</masschroq>
```

One peptide text file corresponds to a sample data. In the above example the `peptide_sample0.txt` file contains only the peptides identified in `sample0` data. Running `masschroq` on this input file will produce a new XML input file named `parsed-peptides_input_file.xml` containing the original `input_file.xml` with all the identified peptides integrated and organized in the masschroqML format.

 At this point MassChroQ automatically continues execution on the newly generated `parsed-peptides_input_file.xml` performing the analysis instructions it contains. If you do not want MassChroQ to continue the analysis, but only parse the peptide files you should use the `-parse-peptides` option as follows :

```
masschroq --parse-peptides input_file.xml
```

This will produce a new XML input file named `parsed-peptides_input_file.xml` containing the original `input_file.xml` with all the identified peptides integrated and organized in the masschroqML format but will not continue analysis on it.


2.2.3 Temporary working directory option

While it parses the mzXML/mzML data files, MassChroQ writes the spectra information they contain into temporary files that it accesses during execution time each time it needs them. One temporary file per data file is created. By default `masschroq` puts these temporary files on its currently working directory. These files are named `masschroq_tmp_file` followed by a randomly generated extension. You should not delete or alter these files while `masschroq` is working. It will automatically delete them when finished.

To change the directory where MassChroQ puts its temporary working files you can specify the directory of your choice by using the `-tmp-dir` option on the command-line as follows :

```
masschroq --tmp-dir DIRECTORY input_file.xml
```

MassChroQ will perform analysis on `input_file.xml` and will put the temporary files in `DIRECTORY` instead of its current working directory.

 The size of a temporary file produced is very close to the size of the corresponding data file. Be careful when specifying another working directory not to choose one that has size limitations.

Chapter 3

How MassChroQ works

In this section we give an in-depth explanation of MassChroQ's operation. In the following section (4) we give you detailed practical usage information.

3.1 Quantified items

In MassChroQ the user determines operating items of interest on which XIC extraction, peak detection and quantification processes will be performed. These items can be :

- all the identified peptides in the data being analyzed;
- all the identified isotopes in the data being analyzed;
- a given list of mass over charge (m/z) values;
- a given list of couples of mass over charge and retention times values (mz - rt).

The user can choose one or several of the above items. MassChroQ will extract the corresponding XICs in every sample data being analyzed and will perform peak detection and quantification on each of them.

3.1.1 Identified peptides

A peptide in MassChroQ is defined as :

- a unique amino-acid sequence;
- a unique MH value : mass of the peptide plus mass of an H^+ ion).

MassChroQ does not perform identification of peptides/proteins. This has to be performed upstream using the identification tool of your choice. It is up to the user to provide the identified peptides (in case he wants to operate on them) in TSV or CSV (tab, comma or semi-colon separated values) format files (for details on this format see section 5). MassChroQ parses these files and puts the information they contain in its input masschroqML file.

Most of the identification tools (Mascot, X!Tandem, Phenyx) offer the possibility to export identification results in TSV files. Our *X!Tandem pipeline* offers the possibility to directly export X!Tandem identification results in a masschroq XML input file (no parsing is necessary).

3.1.2 Identified isotopes

If isotopic labelling has been performed and the user needs to quantify all the identified isotopes, he simply describes the different isotopic labelings performed on the different data being analysed in the masschroqML file. During analysis MassChroQ automatically computes isotopic masses (using these descriptions) and quantifies the desired isotopes.

For a precise explanation of how to describe and quantify isotopes see section 4.5.

3.2 Parsing of LC-MS/MS files in mzXML or mzML formats

MassChroQ can parse mzXML as well as mzML LC-MS files. Due to the simpler, far more robust and stable nature of the mzXML format, we highly recommend its use in MassChroQ instead of the mzML one.

MassChroQ does not validate mzXML and mzML files against their respective schema (it is not its goal), the user has to provide valid files. Usually, most of the proteomic pipeline tools that convert raw data to mzXML/mzML format produce valid files.

In both formats, MassChroQ parses all the MS levels it finds (1, 2 and greater). If the items to be quantified are the identified peptides, LC-MS run files should contain MS levels 1 and 2 in order for MassChroQ to work. Indeed, MassChroQ uses the MS/MS information to compute the real observed retention times of these peptides in each run. This retention time will be used later during peak matching to assign the computed quantitative value to the right corresponding observed peptide and to avoid false assignments.

MassChroQ automatically decodes the base64 encoded spectra in both mzXML and mzML formats, in both 32 or 64 bytes precision. MassChroQ processes nei-

ther compressed spectra, nor compressed mzXML/mzML files. If this feature is important to you, please let us know and we will try to implement it sooner than scheduled.

3.3 Grouping of LC-MS runs

In MassChroQ the user defines groups of LC-MS runs. As explained in section 1.2, the user is supposed to group the runs presenting technical similarities (for example a group of samples of the same fraction, or a group of samples obtained from an LTQ low resolution spectrometer, etc.).

The user can define several different groups in the same analysis. He can then define different alignment methods and different quantification methods for each of these groups. This allows him to run specialized analysis on several different set of samples in one shot.

Groups do not only offer flexibility, they are also helpfull with some extra possibilities that MassChroQ implements:

Efficient XIC extraction: XICs for a given identified peptide will only be extracted in groups where the MS/MS allowed its identification, no unnecessary extractions will be performed.

Smart quantification: peptides identified in at least one run of the group, will be quantified in every run of this group, including those where they have not been identified. See section 3.8.2 for more details on how this is done in MassChroQ.

The final quantification results in MassChroQ are sorted by group and by run, associating to each identified peptide (or other chosen entity) its quantitative value in every group and in every run of the analysis. They allow easy statistical analysis without ambiguity.

3.4 XIC extraction

The underlying operating items in MassChroQ are the XICs. Whatever operating item the user has chosen to analyze (identified peptides, isotopes, mz or mz-rt values), MassChroQ extracts and analyzes the XIC corresponding to the mz of this item.


More precisely, in the XIC of a given item of interest, the mz of this item is extracted from the entire LS-MS run. The intensity (within a mass tolerance range centered on the given mz) is plotted at every retention-time in the analysis.

The size of the mass tolerance window depends on the mass accuracy and the mass resolution of the spectrometer. In MassChroQ the user can define it as a XIC extraction parameter (`mz_range` and `ppm_range` parameters).

The intensity of XICs in MassChroQ can be represented in two different ways:

- as the summed intensity across the range of masses (also called TIC : Total Ion Current chromatogram);
- as the most intense (maximal) peak in the range of masses (base peak chromatogram).

The user can choose one of this representations in the XIC extraction parameters.

 In MassChroQ we have purposefully chosen to perform quantification on the extracted XICs as explained above, rather than on feature detection on the 2D virtual image which many other software use. Indeed, the latter needs high resolution in MS mode in order to be able to identify isotopic profiles. By contrast, quantification based on XICs can be used with low-resolution as well as with high-resolution mass spectrometers by simply adapting the window size of XIC extraction.

3.5 XIC filtering

The following XIC filters are implemented in MassChroQ :

- The background filter : this is a median filter by default. If the user wants he can add an Open (max/min) morphological filter. Both of this filters are widely used for baseline signal noise removal.
- The smoothing filter : a moving window average filter that smooths the signal. This filter can be useful in rare cases of very noisy signals, but as the Zivy detection method already performs a temporary smoothing filter before peak detection, this filter is usually unnecessary.
- The anti-spike filter : removes artifact spikes that some HR spectrometers (e.g. Orbitrap) introduce in the signal.

3.6 Peak detection

Once the XICs are extracted, MassChroQ detects all the peaks on them. He then computes the peak boundaries and integrates the peak area, the latter being the quantitative value.

The following peak detection methods are possible in MassChroQ :

- the *Moulon peak-detection* method : a simple threshold peak detection method;
- the *Zivy peak-detection* method : a combination of open-close morphological filtering of the signal with a local maxima detection algorithm using thresholds.


3.6.1 The Moulon peak detection algorithm

The Moulon peak detection is the historical method in MassChroQ (it has been widely replaced by the Zivy one in practice). This method proceeds as follows:


- it applies an optional moving average filter to smooth XIC intensities (the moving window being a user-defined parameter);
- browses intensity signal in ascending retention time order and when it reaches the `tic start` intensity value parameter it begins detection of a maximal local intensity;
- ends detection of the maximal local intensity when reaching the `tic stop` intensity value threshold parameter.

3.6.2 The Zivy peak detection algorithm

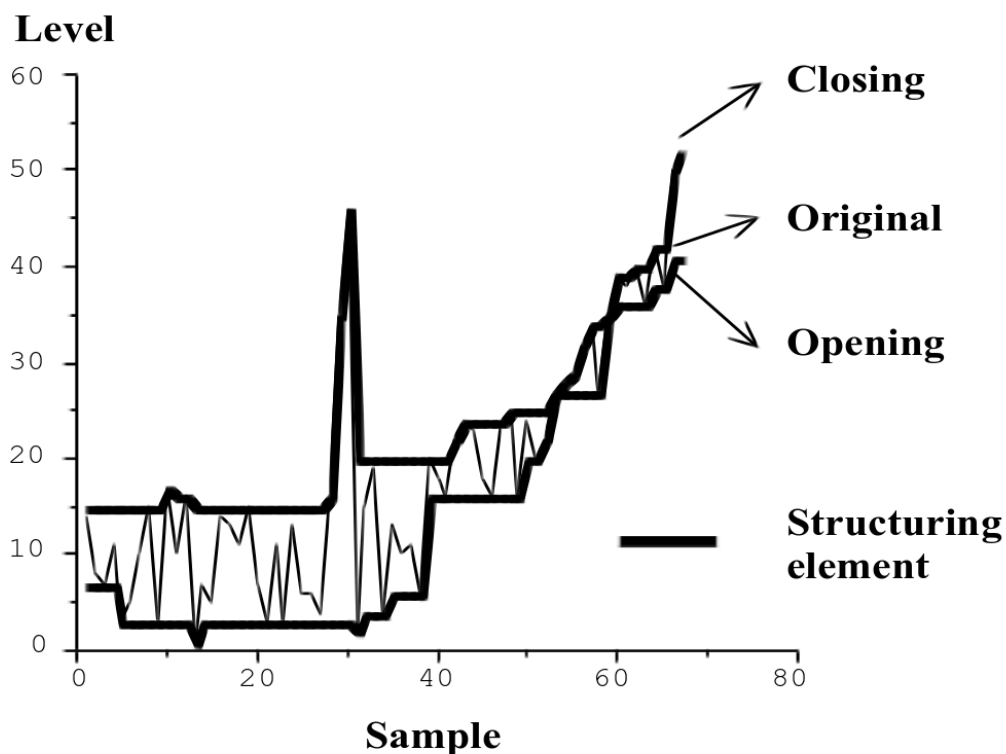
The Zivy peak detection method has widely replaced the Moulon one in practice in our laboratory, giving much more accurate and precise results.

 The Zivy peak detection method is a peak localization method: its purpose is to determine the peak positions and the peak boundaries on the signal. Peak intensities and peak area are then computed on the original unaltered signal.

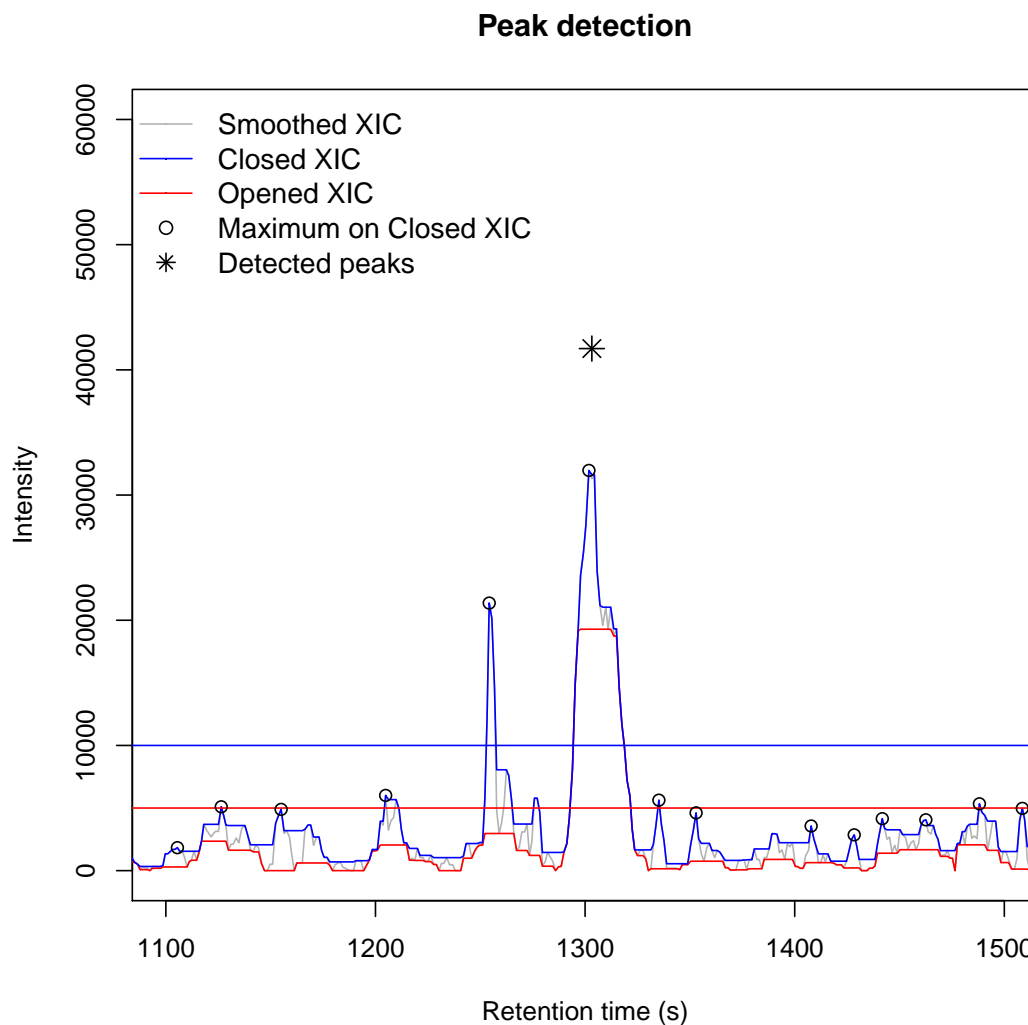
This method uses morphological opening and closing signal transforms with small flat linear structural elements (also known as respectively max/min and min/max transforms). Mathematical morphology was born in 1964 from the collaborative work of Georges Matheron and John Serra at the *Ecole des Mines de Paris* (see [Ser82]). Since then, morphological transforms have been widely used in image processing to remove noise and to detect peaks or edges showing their efficiency in particular in noisy signals (see [Mar04] and [LHS87]).

 On a one-dimensional signal, the open (resp. close) transform with a flat linear structural element (i.e. a segment) of size R is equivalent to replacing the signal values at every point by the maximum of the minimum (resp. minimum of the maximum) of all the points in a neighborhood of radius R (see [LL88] and [GLB00]). This is what we do in MassChroQ.

Schematically, as illustrated in the figure below, opening and closing transforms with flat linear structural elements both smooth and simplify the original signal: opening removes small peaks by flattening them from the top and closing fills small holes by filling them from below.



The opening eliminates peaks that are thinner than the structural element. By choosing an appropriate size of radius R the opening separates background signal from relevant one. Indeed, thin background peaks are eliminated, but in return intense relevant peaks are flattened, which makes the opening not suited for peak detection. In contrast, the closing eliminates valleys thinner than the structural element and not only it perfectly preserves peaks, but it also smooths them and clarifies its boundaries. You can see it on the following figure which comes from a real trace file produced by MassChroQ.



Here is how the Zivy peak detection algorithm works:

- Peak localization in MassChroQ is performed on the closing signal, which preserves peaks but also clarifies its boundaries. More precisely, local maxima intensity positions are detected on the closed signal.
- These maxima are then filtered using two intensity thresholds; only the ones that overpass the thresholds are retained.

Threshold on the open signal: If a local maxima position reported on the open signal does not exceed in intensity the open threshold it is eliminated. This eliminates intense and thin noise peaks (almost spikes) that are flattened by the open transform given their thinness.

Threshold on the closed signal: If a local maxima position reported on the close signal does not exceed in intensity the threshold on the close signal, it is eliminated. This eliminates very wide noise peaks (hence not flattened by the open transform) but not intense.

- The final retained local maxima are the peak positions in MassChroQ. We use this positions to compute peak boundaries on the closed signal and also peak intensity, and peak area on the original signal. Indeed, morphological transforms in MassChroQ are solely used to detect peak positions: the peak intensity and peak area are computed on the original signal using these positions.



In the figure above one can see different interesting cases: for example the very intense peak on the left of the unique detected peak is eliminated because its intensity in the open signal does not exceed the open threshold (too thin to be a relevant peak). This peak is indeed a noisy pulsing spectrometer effect.

Here follows a precise description of the Zivy peak detection algorithm:

Algorithm 1 Zivy peak detection algorithm

Input: a XIC X (set of retention time/intensity values),
mean_filter_half_edge, *minmax_half_edge*, *maxmin_half_edge*,
detection_threshold_on_max and *detection_threshold_on_min*.

Output: the set of relevant peak positions on X .

- 1: Apply a moving average filter of *mean_filter_half_edge* half-window size on the X intensities (optional)
 - 2: Compute \mathcal{O}_X : the open transform of X with a structural segment of radius *maxmin_half_edge*.
 - 3: Compute \mathcal{C}_X : the close transform of X with structural segment of radius *minmax_half_edge*.
 - 4: Compute **{local-max-points}**: the set of (retention time/intensity) points corresponding to the local maximum intensities on the close \mathcal{C}_X signal.
 - 5: **for all** *point* in **{local-max-points}** **do**
 - 6: **if** (*intensity(point(\mathcal{O}_X))* > *detection_threshold_on_min* **and** *intensity(point(\mathcal{C}_X))* > *detection_threshold_on_max*) **then**
 - 7: *point* is a peak
 - 8: **end if**
 - 9: **end for**
 - 10: **return** the set of the such obtained peaks.
-

Line 1: optionally smooth the signal with a moving average filter;

Line 2 and 3: compute the open and close transforms of the signal;

Line 4: perform a local maximum detection on the intensities of the closed signal; we obtain **{local-max-points}**, a preliminary set of (retention time, intensity) potential peak positions.

Lines 5-10: for every such (retention time, intensity) point:

- check that the corresponding retention time point in the opened signal \mathcal{O}_X has a greater intensity than the open threshold *detection_threshold_on_min*;
- check that the corresponding retention time point in the closed signal \mathcal{C}_X has a greater intensity than the close threshold *detection_threshold_on_max*;

The final peak positions are the points that verify both of these conditions.

Afterwards: In reality, the peak boundaries and peak area (final quantification value) are also computed during the detection Zivy algorithm. Indeed, after having selected the final peak positions, we compute the peak boundaries positions on the closed signal. The closed signal does not preserve peak boundary intensity values but it preserves boundary acquisition positions. Then, the peak area is computed by using these boundary positions on the original signal. So the final quantification value is computed on the original unaltered signal.

The Zivy peak detection method is inspired in part by the morphological *top-hat* peak detection method first introduced in [Mey78].

3.7 Alignment

LC-MS instruments do not trigger MS/MS at exactly the same retention time in every sample, hence retention time distortions can often occur between runs. Thus, RTs must be aligned in every sample before peak matching.

MassChroQ performs alignment of samples of the same group. For each group of runs to be aligned the user chooses a *reference alignment run* : the reference run against which all the other runs of the group will be aligned. It stands to reason that the user should choose a representative run as reference alignment run, otherwise all the runs of the group will present abnormal retention time deviations affecting peak matching and quantification values.

Two different alignment algorithms are implemented in MassChroQ : the **OBI-Warp** alignment method and the MS2 alignment method.

3.7.1 The OBI-Warp alignment method

The OBI-Warp (Ordered Bijective Interpolated Warping) is an external integrated library developed by John T.Prince in the University of Texas at Austin. This method is based on the MS level 1 data solely. It considers the spectra as matrixes that it aligns along the (MS level 1) retention time axis by using dynamic time warping and a one-to-one interpolated warp function.

For more information on the OBI-Warp alignment library see <http://obi-warp.sourceforge.net/> and [PM06].

3.7.2 The MS/MS alignment method

The in-house developed MS/MS alignment method is based on MS/MS (MS level 2) data, hence it is not possible to use this method with data not containing MS/MS acquisition.

This method uses MS/MS identifications as landmarks to evaluate time deviation along the chromatography. More precisely, suppose we want to align two runs. We first calculate the retention time deviation of the MS/MS identified peptides these two runs have in common. Then, by linear interpolation, we use this deviation curve to calculate a tendency deviation curve of the MS level one retention times. Of course this deviation curve is accurate only if there are enough points that allowed its shape, i.e. if there are enough common identified peptides in the two runs. For each alignment MassChroQ will output the number of shared peptides. Also, he will output a .trace file for each run aligned. These files contain the traces of every alignment step and can be used for precise checking og the alignment procedure ans parameter adjustment.

Here follows a precise explanation step by step of the MS/MS alignment algorithm. Let Run_1 be the run whose MS level one retention times are going to be aligned towards the MS level one retention times of the reference run Run_{ref} . We will use the following notation conventions :

- RT_{ms1} denotes a set of MS level one retention times and RT_{ms2} a set of MS/MS retention times.
- $RT_{ms2}(Run_1)$ denotes the set of the retention times of the identified peptides during MS/MS acquisition in Run_1 .
- rt_{ms1} denotes a retention time member of the RT_{ms1} set.

The MS/MS alignment algorithm proceeds as follows:

Algorithm 2 MS/MS alignment of $RT_{ms1}(Run_1)$ towards $RT_{ms1}(Run_{ref})$

Input: $RT_{ms1}(Run_1)$, $RT_{ms1}(Run_{ref})$,
 $RT_{ms2}(Run_1) \neq \emptyset$, $RT_{ms2}(Run_{ref}) \neq \emptyset$,
 $ms1_smoothing_window$, $ms2_smoothing_window$,
 $ms2_tendency_window$.

Output: $alignedRT_{ms1}(Run_1)$ set of values.

- 1: Get the list *Peps* of all the peptides identified in both Run_1 and Run_{ref} ;
 - 2: **for all** *pep* in *Peps* **do**
 - 3: $\Delta rt_{ms2}(Run_1) = rt_{ms2}(Run_1) - rt_{ms2}(Run_{ref})$
 - 4: **end for**
 - 5: To each rt_{ms2} in Run_1 , associate its $\Delta rt_{ms2}(Run_1)$ as computed above.
 - 6: Let $(RT_{ms2}, \Delta RT_{ms2})(Run_1)$ be the set of all such pairs.
 - 7: Apply a moving median of size $ms2_tendency_window$ on the ΔRT_{ms2} set;
 - 8: Apply a moving average of size $ms2_smoothing_window$ on the ΔRT_{ms2} set;
 - 9: Add first and last elements to the $(RT_{ms2}, \Delta RT_{ms2})(Run_1)$ (by extrapolation of MS1 retention time values) as follows:
 - 10: $first-rt_{ms2} = first-rt_{ms1} - 1$ and $\Delta first-rt_{ms2} = \Delta first-rt_{ms1}$
 - 11: $last-rt_{ms2} = last-rt_{ms1} + 1$ and $\Delta last-rt_{ms2} = \Delta last-rt_{ms1}$
 - 12: **for all** rt_{ms1} in $RT_{ms1}(Run_1)$ **do**
 - 13: Compute a corresponding $\Delta rt_{ms1}(Run_1)$ value by linear interpolation on the set of $(RT_{ms2}, \Delta RT_{ms2})(Run_1)$, i.e.:
 - 14: $\Delta rt_{ms1} \leftarrow$ linear interpolant of $(rt_{ms2}^1, \Delta rt_{ms2}^1)$ and $(rt_{ms2}^2, \Delta rt_{ms2}^2)$ where rt_{ms2}^1 and rt_{ms2}^2 are the two nearest surrounding points to rt_{ms1} .
 - 15: **end for**
 - 16: Apply a moving average of size $ms1_smoothing_window$ on the such obtained ΔRT_{ms1} set;
 - 17: **for all** rt_{ms1} in $RT_{ms1}(Run_1)$ **do**
 - 18: $alignedrt_{ms1}(Run_1) = rt_{ms1} - \Delta rt_{ms1}$
 - 19: **end for**
 - 20: Automatically check the $alignedRT_{ms1}(Run_1)$ curve slope and correct it if necessary;
 - 21: **return** the set of $alignedrt_{ms1}(Run_1)$ values.
-

Step 1 (lines 1-4): here we get the retention times of all the common peptides identified in both runs. The computation of peptide retention times follows the *best Rt* method explained in section 3.8.1.

Step 2 (lines 2-6): to each peptide MS2 retention time in Run_1 we associate its deviation from the peptide's MS2 retention time in Run_{ref} .

Step 3 (lines 7-8): smoothing of this deviation curve: we first apply a moving median filter followed by a moving average filter. The half-window sizes for these filters are parameters defined by the user. They can be set to zero if no filtering is needed.

Step 4 (lines 9-11): compute the first and last rt_{ms2} and $\Delta rt_{ms2}(Run_1)$ elements by linear extrapolation on the first and last $rt_{ms1}(Run_1)$ points.

Step 5 (lines 12-15): for each rt_{ms1} level one retention time in Run_1 to be aligned, we compute a corresponding Δrt_{ms1} value by linear interpolation on the set of $(RT_{ms2}, \Delta RT_{ms2})(Run_1)$ level 2 retention times and associated deviations (computed in Step 2). The linear interpolation is done on the two rt_{ms2} values that surround this rt_{ms1} .

Step 6 (line 16): we smooth the such obtained Δrt_{ms1} points by applying a moving average filter if the user has decided to;

Step 7 (lines 17-19): for each original MS1 retention time in Run_1 we compute its corresponding aligned MS level one retention time.


Step 8 (line 20): the such computed aligned MS1 retention times must be in an ascending order (as the original retention times are). Despite the smoothings, sometimes (rarely observed) this is not the case at this point. Indeed, since the computed value is a deviation in time, once we apply this deviation to the original time to obtain the aligned one, there is no guaranty that the ascending order of aligned time values is preserved. Hence, the algorithm always checks the ascending order of the aligned values and automatically applies a correction to ensure it if needed. The correction is performed as follows:

- compute the correction parameter as the slope of the original retention time curve ($rt_{ms1}(Run_1)$); divide this parameter by 4 for a finer correction;
- whenever two consecutive aligned retention time values are not in increasing order, we replace the smaller one with the biggest one plus the correction parameter.

3.7.3 Previous alignments and cascade alignments

Previous alignments

For each LS-MS run it aligns MassChroQ creates a `run_filename.time` file containing the original retention times of the run and the new corresponding ones after alignment. These files are placed in the same directory that the run files they correspond to. MassChroQ automatically looks for these files each time it runs and loads them if they exist. This way the user does not have to repeat an alignment that he has previously done, he simply provides the corresponding `.time` file. Also, this allows him to provide alignment values generated by other external alignment tools. The user just puts the `.time` files in the same directory as the run files, MassChroQ loads them and analyzes the runs with the aligned time values.

 For this to work, the `masschroqML` input file should not contain alignment directives. Indeed, if an alignment instruction asks MassChroQ to align a sample whose `.time` file has been loaded before, he destroys these `.time` values and performs the asked alignment. Thus, the alignment instructions in the `masschroqML` input files have the priority over the preloaded `.time` alignment files.

Cascade alignments

3.8 Peak matching

After alignment, xic extraction and peak detection, MassChroQ performs peak matching: the detected peaks are assigned to the peptides or other entities being quantified. Peak matching in MassChroQ is based on retention times, it is performed as follows: the quantitative value of a peak (i.e. the peak area) is assigned to a peptide if and only if the RT of this peptide is within the boundaries of this peak.

What is the RT of a peptide? In a given run a peptide can be identified or not. In case it has been identified, it can be identified at several retention times, this is why its RT is computed with the *best RT* method; in case it has not been identified, its RT is computed with the *smart quantification* method which allows quantification of peptides even in runs they have not been identified, provided they have been identified in another run of the same group.

3.8.1 The best RT method

A given peptide can be observed/identified at several different retention times in a given run, with different intensities or charge states. During parsing MassChroQ will get from the LC-MS run `mzXML` or `mzML` file all the retention times the

peptide has been identified in and the corresponding precursor intensities. He will then retain only the retention time corresponding to the most intense occurrence of this peptide. This will be the retention time of this peptide for this run during the rest of the analysis. We refer to this method as the *best RT* method.

3.8.2 Smart Quantification

If a given peptide has been identified in at least one sample of the group, during peak matching in the samples where this peptide has not been identified MassChroQ will nevertheless check for peaks corresponding to this peptide. Indeed MassChroQ computes the mean of this peptide's retention times in the samples it has been identified in (more precisely it computes the mean of all its best RTs). In the sample the peptide has not been identified in, MassChroQ checks whether this mean RT belongs to a detected peak area or not; if it does the peak and its quantification value are assigned to this peptide.

Chapter 4

The masschroqML format

In this section we give a detailed practical description of `masschroqML`, MassChroQ's XML input file format.

To illustrate the `masschroqML` format we will use the complete example file called `masschroq_complete_input_example.xml` included in appendix A. This file is also located in the `doc/xml_examples` directory of your MassChroQ installation directory.

Here follows an explanation line by line of this file.

4.1 Data files

In the `<rawdata>` block the user defines the data files to analyze.

```
3 <rawdata>
4 <data_file id="samp0" format="mzxml" path="bsa1.mzXML" type="centroid"/>
5 <data_file id="samp1" format="mzxml" path="bsa2.mzXML" type="profile"/>
6 <data_file id="samp2" format="mzml" path="/home/user/bsa3.mzml"
   type="profile"/>
7 <data_file id="samp3" format="mzml" path="/home/user/bsa4.mzml"
   type="profile"/>
8 </rawdata>
```

For each file to analyze a `<data_file>` tag line has to be present. The `<data_file>` tag gives MassChroQ all the information needed to identify and find a file on the system. It contains the following attributes:

- The `id` attribute is a unique name assigned to the data file in order to reference it later in the document. It is mandatory.
- The `format` attribute indicates the format of the data file. It can be `mzxml` or `mzml`. It is mandatory.

- The `path` attribute is the absolute or relative path to the physical file on your local system.
- The `type` attribute informs the user about the type of data acquisition of this file. It can be `profile` or `centroid`. This attribute is optional, MassChroQ does not use it.

In the above example four data files have been defined : `samp0` in `mzxml` format, `samp1` in `mzxml` format, `samp2` and `samp3` in `mzml` format.

4.2 Groups

In the `<groups>` block the user defines groups of data files.

```
9 <groups>
10 <group data_ids="samp0 samp1" id="G1"/>
11 <group data_ids="samp2 samp3" id="G2"/>
12 </groups>
```

- The `data_ids` attribute is the list of space separated data files identifiers that form this group. It is mandatory.
- The `id` attribute is a unique name assigned to the group in order to reference it later in the document. It is mandatory.

In the above example we have defined group `G1` containing the previously defined data `samp0` and `samp1` and the group `G2` containing `samp2` and `samp3`.

4.3 Peptide text files

In the `<peptide_files_list>` block the user defines the path to the peptide text files containing information about the identified peptides and proteins in the data being analyzed. These peptide files will be parsed and analyzed by MassChroQ.

```
13 <peptide_files_list>
14 <peptide_file data="samp0" path="bsa1_peptides.txt"/>
15 <peptide_file data="samp1" path="bsa2_peptides.txt"/>
16 <peptide_file data="samp2" path="bsa3_peptides.txt"/>
17 <peptide_file data="samp3" path="bsa4_peptides.txt"/>
18 </peptide_files_list>
```

- The `data` attribute contains the data id this peptide file correspond to. One peptide text file per data file should be provided. It is mandatory.
- The `path` attribute is the path of the peptide file in the local system. It is mandatory.

In this example, we are telling MassChroQ that the file `bsa1_peptides.txt` contains the identified peptides in data file `bsa1.mzxml` called `samp0`; the file `bsa2_peptides.txt` the ones identified in sample `samp1`.

4.4 Identified peptides and proteins

Once it has parsed the given peptide files to identify peptides, MassChroQ automatically puts the results in the following form :

```

19 <protein_list>
20 <protein desc="conta|P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
    id="P1.1"/>
21 <protein desc="conta|P02770|ALBU_RAT SERUM ALBUMIN PRECURSOR."
    id="P1.2"/>
22 </protein_list>
23 <peptide_list>
24 <peptide id="pep0" mh="1463.626" mods="114.08" prot_ids="P1.1"
    seq="TCVADESHAGCEK">
25 <observed_in data="samp0" scan="655" z="2"/>
26 <observed_in data="samp1" scan="798" z="2"/>
27 </peptide>
28 <peptide id="pep1" mh="1103.461" mods="57.04" prot_ids="P1.1"
    seq="ADESHAGCEK">
29 <observed_in data="samp3" scan="663" z="2"/>
30 </peptide>
31 </peptide_list>

```

It creates two blocks :

- the `protein_list` block containing the description of each identified protein;
- the `peptide_list` block containing for each identified peptide :
 - a unique id (mandatory),
 - the amino-acid sequence (mandatory),
 - the MH value (mass of the peptide plus mass of an $H+$ ion); the MH already takes into account the amino-acids modifications, (mandatory);
 - the `mods` attribute giving optional information about the amino-acid modifications on this peptide (here the total mass of the modification) (optional);
 - the id-s of the identified proteins this peptide belongs to (mandatory);
 - a list of `observed_in` elements that indicate the data file, the scan number and the charge state this peptide has been observed in (mandatory).



The user does not have to fill the `<protein_list>` and `<peptide_list>` blocks:

- The *X!Tandem pipeline*, when asked to export its results in masschroqML format, automatically creates these two blocks in this file.
- When you ask MassChroQ to parse peptide text files with an input file containing a `peptide_files_list` block as described in the previous section, it will create a new file named `parsed-peptides_input_file.xml` already containing the above peptide and protein blocks.


4.5 Isotope labels


The `isotope_label_list` block contains the list of the isotopic labels performed on the data being analyzed if any.

```
32 <isotope_label_list>
33 <isotope_label id="iso1">
34 <mod at="Nter" value="28.0"/>
35 <mod at="K" value="28.0"/>
36 </isotope_label>
37 <isotope_label id="iso2">
38 <mod at="Nter" value="32.0"/>
39 <mod at="K" value="32.0"/>
40 </isotope_label>
41 </isotope_label_list>
```

- The `id` attribute uniquely identifies an isotope label (mandatory).
- The `at` attribute indicates where the label has been performed (mandatory).
- The `value` attribute indicates the mass modification value for this label (mandatory).

In the above example we have defined two isotope labels : label `iso1` consisting in two modifications at Nter and K both with mass modification value of 28; and label `iso2` label containing Nter and K modifications each of value 32.

 Different `isotope_label` elements can be defined, for example to perform quantification on different groups of differently labeled samples.

 The identified peptides in the `peptide_list` block do not take into account the isotopic labelings (their MH value is not the modified one). MassChroQ computes their isotopic masses during quantification (if quantification on isotopes has been asked) using the information contained in the `isotope_label_list` block if any.


4.6 Alignments


```
42 <alignments>
43 <alignment_methods>
44 <alignment_method id="my_ms2">
45 <ms2>
46 <ms2_tendency_halfwindow>10</ms2_tendency_halfwindow>
47 <ms2_smoothing_halfwindow>5</ms2_smoothing_halfwindow>
48 <ms1_smoothing_halfwindow>3</ms1_smoothing_halfwindow>
49 </ms2>
50 </alignment_method>
51 <alignment_method id="my_obiwarp">
52 <obiwarp>
53 <lmat_precision>1</lmat_precision>
54 <mz_start>500</mz_start>
55 <mz_stop>1200</mz_stop>
56 </obiwarp>
57 </alignment_method>
58 </alignment_methods>
59 <align group_id="G1" method_id="my_ms2" reference_data_id="samp0"/>
60 <align group_id="G2" method_id="my_obiwarp" reference_data_id="samp2"/>
61 </alignments>
```

In the `alignments` block we define the different alignment methods that we will use to align our different groups of data. In this example we have defined two alignment methods:

- an `ms2` type alignment method with id `my_ms2` (mandatory);
- an `obiwarp` type alignment with id `my_obiwarp` (mandatory).

For each alignment method we have set the appropriate parameters, for example for the `my_ms2` method we have set the half moving window used to smooth the MS/MS retention times to 5. For details on every alignment parameter see the parameters cheat-sheet on section 6.

 In the `align` element (lines 57 and 58), we give MassChroQ the order to perform alignment on groups *G1* and *G2* using respectively the `my_ms2` method and the `my_obiwarp` method. All the attributes here are mandatory.

 The `reference_data_id` attribute (lines 57 and 58) indicates the data file in the group that will be used as a reference for the alignment : the other samples of the group will be aligned towards this reference sample. Thus, the choice of the reference sample in alignments is very important.

4.7 Quantification methods

In the `quantification_methods` block we define the different quantification methods we will use in this analysis.

```
62 <quantification_methods>
63 <quantification_method id="my_qzivy">
64   <xic_extraction xic_type="sum">
65     <mz_range max="1.5" min="0.5"/>
66   </xic_extraction>
67   <xic_filters>
68     <anti_spike half="5"/>
69     <background half_mediane="5" half_min_max="15"/>
70     <smoothing half="3"/>
71   </xic_filters>
72   <peak_detection>
73     <detection_zivy>
74       <mean_filter_half_edge>1</mean_filter_half_edge>
75       <minmax_half_edge>3</minmax_half_edge>
76       <maxmin_half_edge>2</maxmin_half_edge>
77       <detection_threshold_on_max>5000 </detection_threshold_on_max>
78       <detection_threshold_on_min>3000 </detection_threshold_on_min>
79     </detection_zivy>
80   </peak_detection>
81 </quantification_method>
```

As we can see in the lines above, a quantification method consists in a XIC extraction method, XIC filters and a peak detection method. For each quantification method we define:

- A unique quantification `id` used to reference this method (mandatory).
- The `xic_type` : the type of the XIC extraction. It can be `sum` if we want the XICs to be extracted by computing the sum of the intensities, or `max` for the maximum intensity (mandatory).
- The size of the mass tolerance window of extraction in `ppm_range` or `mz_range` (mandatory choice).
- A list of XIC filters to apply (optional).
- A peak detection method among `detection_zivy` and `detection_moulon`.

In the above example we have defined a quantification method called `my_qzivy` that :

- extracts XICs by computing the sum of the intensities;
- uses an m/z range for extraction of 0.5 to 1.5;
- applies an anti-spike filter on the XICs, followed by a background noise removal filter and a smoothing moving-window filter;
- uses a `detection_zivy` peak detection method.

In the next lines of our example file :

```
82 <quantification_method id="my_qmoulon">
83   <xic_extraction xic_type="max">
84     <ppm_range max="1.5" min="0.5"/>
85   </xic_extraction>
86   <xic_filters>
87     <background half_mediane="5" half_min_max="15"/>
88   </xic_filters>
89   <peak_detection>
90     <detection_moulon>
91       <smoothing_point>3</smoothing_point>
92       <TIC_start>5000</TIC_start>
93       <TIC_stop>3000</TIC_stop>
94     </detection_moulon>
95   </peak_detection>
96 </quantification_method>
97 </quantification_methods>
```

we define the `my_qmoulon` quantification method which extracts XICs based on the max intensity, using a ppm (parts per million) range of 0.5 – 1.5. It applies a background removal filter to them and uses the `detection_moulon` peak detection method.

4.8 Quantifying

In the `quantify` block we describe the items to be quantified in each group of data (peptides, isotopes, m/z values or $(m/z, rt)$ values) and the quantification methods to be used.

4.8.1 Quantifying peptides

```
116 <quantify id="q1" withingroup="G1" quantification_method_id="my_qzivy">  
117 <peptides_in_peptide_list mode="real_or_mean"/>  
118 </quantify>
```

As we can see in the lines above, we have chosen to:

- quantify all the peptides (`<peptides_in_peptide_list>` element) identified
- in group *G1* (`withingroup` attribute)
- by using the *myqzivy* quantification method (`quantification_method_id` attribute).
- The `mode` attribute indicates the way we compute this peptide's retention time in each run sample during quantification. This `rt` value is the one used for peak matching after peak detection during the quantification process. Two RT computation modes are available:

real_or_mean: if the peptide has been identified in a run sample, its retained RT is the observed one (more exactly the best RT one). If it has not been identified in this sample, its RT is the mean of the RTs in the run samples of the same group where this peptide has been identified. This way we can quantify a peptide in a sample even if it has not been identified in it. For details on this method see section 3.8.

mean: if this mode is selected, the RT of the peptide in a sample is the mean of the best RTs of this peptide in the other samples of the group the peptide was identified in. No difference is made whether the peptide was identified in this sample or not.

4.8.2 Quantifying isotopes

```
119 <quantify id="q2" withingroup="G2" quantification_method_id="my_moulon">
120 <peptides_in_peptide_list mode="real_or_mean" isotope_label_refs="iso1
    iso2"/>
```

As you can see above, to quantify isotopes it suffices to add the attribute `isotope_label_refs` containing the references to the isotope labels we want to quantify. These labels have been previously defined in the `masschroqML` file (see section 4.5). `MassChroQ` automatically computes the isotope masses after modification during quantification.

4.8.3 Quantifying m/z values

The list of the desired m/z values to be quantified can be given as follows:

```
119 <quantify id="q2" withingroup="G2" quantification_method_id="my_moulon">
120 <peptides_in_peptide_list mode="real_or_mean" isotope_label_refs="iso1
    iso2"/>
121 <mz_list>732.317 449.754 552.234 464.251 381.577 569.771
    575.256</mz_list>
```

4.8.4 Quantifying ($m/z, rt$) values

The list of the desired ($m/z, rt$) values to be quantified can be given as follows:

```
119 <quantify id="q2" withingroup="G2" quantification_method_id="my_moulon">
120 <peptides_in_peptide_list mode="real_or_mean" isotope_label_refs="iso1
    iso2"/>
121 <mz_list>732.317 449.754 552.234 464.251 381.577 569.771
    575.256</mz_list>
122 <mzrt_list>
123 <mzrt mz="732.317" rt="230.712"/>
124 <mzrt mz="575.256" rt="254.788"/>
125 </mzrt_list>
126 </quantify>
```

4.9 Result files


Quantification result files

```
98 <quantification>
99 <quantification_results>
100 <quantification_result output_file="result1" format="tsv"/>
101 <quantification_result output_file="result2" format="gnnumeric"/>
102 <quantification_result output_file="result3" format="xhtmltable"/>
103 <quantification_result output_file="result4" format="xml"
    xic_traces="true"/>
104 </quantification_results>
```

In the `quantification_results` block we define the format and name of the files that will contain the final quantification results. The format can be :

- `tsv` : tab-separated values text format; this format is ready to use for statistical analysis;
- `gnnumeric` : Gnome spreadsheet format to use with Gnumeric software;
- `xhtmltable` : xhtml (eXtensible HyperText Markup Language) format; all the results are in an xhtml table and you can visualize them via an internet browser for example;
- `xml` : eXtensible Markup Language format, more precisely this is also a `masschroqML` format; it is intended for development use, for example to integrate `masschroq` results in databases.


The user can choose multiple output formats for the same analysis as we have done in the example above. We have chosen to export results into a `tsv` file that will be called `result1.tsv`, a `gnnumeric` file that will be called `result2.gnumeric`, etc.

 The `tsv` and `xhtmltable` outputs will create two files : one with extension `_pep` containing the quantification results and a second one with extension `_prot` resuming for each peptide, the corresponding proteins and descriptions. The `gnnumeric` format will contain two separate sheets for each of them. In the example above two files named `results_pep.tsv` and `results_prot.tsv` will be created for the `tsv` format and a unique file named `results.gnumeric` containing two sheets for the `gnnumeric` format.

Alignment result (.time) files

When MassChroQ reads the `align` tag in its input XML file (lines 57 and 58 in appendix A), it launches the alignment of each sample in the indicated group towards the reference sample, using the indicated alignment method. For each LC-MS run it aligns, MassChroQ creates a `run_filename.time` file containing two tab-separated columns of values :

- the first column (with `old_rt` header) contains the original retention times of this sample as they appear in the raw data file;
- the second column (called `new_rt`) contains the corresponding computed aligned retention times.

 The `.time` files are useful for analysis and alignment checking, but they can also be used to avoid realigning samples or to inject external alignment values. Indeed, MassChroQ automatically loads previously generated `.time` files right after he parses the run files. This way, one does not have to repeat alignment on previously aligned files. Also it can use an external alignment tool and inject its results via `.time` files in masschroq's analysis. For details on this see section [3.7.3](#).

4.10 Trace files

Quantification traces

You can tell MassChroQ to produce detailed quantification traces in the following way :

```
105 <quantification_traces>
106 <peptide_traces peptide_ids="pep0 pep1" output_dir="pep_traces"
    format="tsv"/>
107 <all_xics_traces output_dir="all_xics_traces" format="tsv"/>
108 <mz_traces mz_values="634.635 449.754 552.234" output_dir="mz_traces"
    format="tsv"/>
109 <mzrt_traces output_dir="mzrt_traces" format="tsv">
110 <mzrt_values>
111 <mzrt_value mz="732.317" rt="230.712"/>
112 <mzrt_value mz="575.256" rt="254.788"/>
113 </mzrt_values>
114 </mzrt_traces>
115 </quantification_traces>
```

Three types of traces can be produced :

- `peptide_traces` : traces for a given list of space separated peptide ids;
- `all_xics_traces` : traces for all the quantified items of the current analysis (i.e. all the extracted XICs);
- `mz_traces` : traces for a given list of space separated mz values;
- `mzrt_traces` : traces for a given list of mz, rt couple of values;

The `output_dir` attribute (which is mandatory) indicates the directory name where the traces should be put in the local system. For each traced item (peptide, mz value or mz-rt value) a file is created in this directory. The traces file names contain information about the group, the MS run, the peptide id, the mz, and the rt of the traced item.

A trace file is always and by default in `tsv` format (the `format` attribute is optional). It contains from left to right order:

- a header giving the precise mz, rt and peptide id information.
- an `rt` column containing the retention time values of the XIC this traced item corresponds to;

- an **intensity** column containing the retention time values of the XIC this traced item corresponds to;
- for each filter applied to this XIC, an **filtered_intensity** column containing the intensity values after filtering;
- an **all_peaks** column containing the peak intensity values of all the detected peaks on this XIC (if a peak has been detected, the corresponding **rt** and **intensity** line will contain the peak's intensity value);
- a **selected_peaks** column containing the retained peaks after peak matching, i.e. the peaks that really correspond to the searched item.

Trace files are very useful for analysis checking, but also for parameter refining. Indeed, one can fastly trace a small list of peptides with different quantification method parameters and compare the traces (for example to see what detection threshold detects more peaks or what XIC filter better removes background noise).


Alignment traces : (.trace files)

For each sample it aligns, MassChroQ automatically creates a **sample_name.trace** file containing the retention time values at each step of the alignment, from the original state to the final aligned one, including the MS/MS alignment values before and after smoothing. These files are not used by masschroq in any way, they are solely intended to help the users check and analyze the alignment.

An alignment trace file is always and by default in **tsv** format. It contains from left to right order:

- an **rt_MS2** column containing the retention time values of the shared peptides in the run being aligned;
- a **deltaRT_MS2** column containing the differences between the retention time value of the shared peptides in the reference run and their retention time value in the run being aligned;
- a **post-median-deltaRT_MS2** column containing the previous **delta_RT** values after a moving window median filter;
- a **post-mean-deltaRT_MS2** column containing the previous **post-median-delta_RT** values after a moving window average filter;
- an **rt_MS1** column containing the level 1 retention time values of the run being aligned, before their alignment;

- a `smoothed-deltaRT_MS1` column containing the computed retention time deviations for the previous `rt_MS1` values (after linear interpolation). These values will be added to the original retention times to obtain the final aligned values. If a smoothing moving filter has been defined by the user, these deviation values are already smoothed.

 Alignment traces are only available for the in-house developed MS2 alignment method. The third-party OBi-Warp alignment method that we have integrated in MassChroQ does not support trace files.

Chapter 5

Specification of the identified peptides files

MassChroQ can parse peptide identification results from text files in Tab, Comma or Semi-Colon Separated Values formats (TSV or CSV).

5.1 masschroqML peptide files instructions

To make MassChroQ parse identified peptides and protein descriptions from text files you should put instruction in the masschroqML file as follows:

```
13 <peptide_files_list>
14 <peptide_file data="samp0" path="bsa1_peptides.txt"/>
15 <peptide_file data="samp1" path="bsa2_peptides.txt"/>
16 <peptide_file data="samp2" path="bsa3_peptides.txt"/>
17 <peptide_file data="samp3" path="bsa4_peptides.txt"/>
18 </peptide_files_list>
```

The `peptide_files_list` group should immediately follow the `</groups>` tag.

At most one peptide file per LC-MS run has to be provided. The `data` attribute references the run id the peptide file corresponds to; the `path` attribute is the path to the peptide file in your system.

5.1.1 masschroq command-line `-parse-peptides` option

Running `masschroq` with an input file containing `peptide_files_list` instructions, will make it parse the indicated peptide files and produce a new masschroqML input file named `parsed-peptides_input_file.xml` containing the original `input_file.xml` plus all the identified peptides integrated and organized in the masschroqML format.

Moreover, MassChroQ will automatically continue analysis on this newly produced file. If you want not to continue analysis but only to parse the peptide files and put them in the `parsed-peptides_input_file.xml` run MassChroQ with the `-parse-peptides` or `-p` option as follows:

```
masschroq --parse-peptides input_file.xml
```

5.1.2 Peptide text file format specification

Here are the first lines of the peptide identification example file presented in appendix B :


```

1 scan sequence mh z proteins mods
2 778 CCTKPESER 1166.4934 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
   114.08
3 839 NYQEAK 752.3585 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR." Ymod
4 1136 TCVADESHAGCEK 1463.5852 2 "P02769|ALBU_BOVIN SERUM ALBUMIN
   PRECURSOR." 57.04
5 1585 SHCIAEVEK 1072.5111 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
6 1935 NECFLSHK 1034.4729 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
7 1960 ECCDKPLLEK 1291.6019 3 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
8 1980 LCVLHEK 898.48236 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."

```

Separation character

In this example values are separated by tabulations. Other correct separation characters are comma “,” and semi-colon “;”.


 In a given peptide file, separation characters should not be melted: one has to exclusively use the tab, the comma or the semi-colon separator everywhere in the file. Melting them will cause masschroq to exit with a parsing error.

Header specification

As shown above, the header for the peptide files is:

```
1 scan sequence mh z proteins mods
```

 The first five columns are mandatory. The sixth column (`mods`) is optional.

 The header (the first five columns exactly) is mandatory: it must be present in all of your peptide files. If another header is used masschroq will exit with a parse error.

The other correct headers, depending on the chosen separation character are :

```
scan,sequence,mh,z,proteins,mods
```

and for the semi-colon :

```
scan;sequence;mh;z;proteins;mods
```

Values specification

The accepted values for a peptide text file are :

- **scan** (mandatory): integer representing the scan number in the mzXML or mzML file this peptide belongs to.
- **sequence**: (mandatory) string representing the amino-acid sequence of a peptide.
- **mh** (mandatory): real number representing the mass of a peptide plus the H^+ mass.
- **z** (mandatory): integer representing the charge state of this peptide.
- **proteins** (mandatory): string in free text representing the description of one protein this peptide belongs to. At most one protein per line is allowed.
- **mods** (optional): string representing the mass modification on this peptide if any. This is optional, it can help facilitate user's analysis but is not used by MassChroQ to compute the peptide mass modifications (which are computed by MassChroQ using isotopes and MH values).





One line represents a given peptide sequence in a given charge state in the given scan number of the corresponding run data, identified in the given protein. Hence, more than one line can be found for the same peptide sequence, with different scan numbers, charge states, or protein description values. This also means that for the same peptide sequence, with same MH, same scan number and same charge state but belonging to two different proteins, two lines should be put in the file, one per each protein (as in lines 5 and 6 above).

Chapter 6

Parameters cheat-sheet

In this chapter you will find the list of all MassChroQ parameters with an explanation, advised values for them and some practical advice.

 An archive containing several ready-to-use examples illustrating different alignment and quantification methods is available from <http://pappso.inra.fr/bioinfo/masschroq/>.

 In all the following moving window filters or transforms, the user is asked to enter a half window size. The corresponding window in MassChroQ will then be of size $2 * half_window + 1$.

6.1 Alignment parameters

The *ObiWarp* alignment parameters

- **lmat_precision (real number)**: matrix precision in Thompson. A good value is often 1.
- **mass_start (mz value)**: beginning of the mass window used to compute the matrix. We advise you (from our experience) to exclude the spectra masses containing contaminants present during the whole run, for example methylxyloxane contaminant.
- **mass_stop (mz value)**: end of the mass window used to compute the matrix.

The *MS2* alignment parameters

- **ms2_tendency_halfwindow (natural integer)**: half size of the window used to apply a moving median on the MS/MS retention time deviation curve. Used to create the tendency deviation curve. Of course the appropriate value for this window depends on the number of identified peptides that the two runs (reference run and run being aligned) have in common. Usually a good value is 10. While aligning, MassChroQ outputs on the console the number of peptides in common which you can use to readjust this parameter if necessary.
- **ms2_smoothing_halfwindow (natural integer)**: half size of the window used to apply a moving average on the MS/MS retention time deviation curve. Smooths the deviation curve. Same as the above parameter, usually a good value is 10.
- **ms1_smoothing_halfwindow (natural integer)**: half size of the window used to apply a moving median on the MS level 1 retention time corrections curve. This smoothing is not necessary most of the time, so its value should usually be 0. It could be used instead of the ms2 smoothing parameter in cases of a small number of shared identified peptides (< 100), in which case a good value is 20.

6.2 Quantification parameters

XIC extraction parameters

These parameters depend on your spectrometer.

- **xic_type (“sum” or “max”)**: type of the XIC intensity representation. It can be:
 - **sum** for the sum of intensities across the range of XIC masses;
 - **max** for the maximal intensity across the range of XIC masses;
- **mz_range** or **ppm_range** parameters: mass tolerance XIC window in mz or ppm resolution. For each of them you have to define:
 - **min**: the minimum value of the window range;
 - **max**: the maximal value of the window range.

XIC filters parameters

- Background filter: corrects baseline noise.
 - **half_mediane (natural integer)**: half size of the moving window used to apply a median on the XIC intensity values. Usually a value of 5 is sufficient, but depending on your spectrometer it can be greater.
 - **half_min_max (natural integer)**: half size of the moving window used to apply an open transform (a min then a max) on the XIC intensities. This filter is sometimes useful in LR signal; it smoothes the signal from below, so a good half-window value would be the peak width (in scan points number) divided by 2. For example for a peak of 30 seconds with one scan per second, this value would be 15. For security make this value a little greater, in this example 20.
- Smoothing filter:
 - **half (natural integer)**: half size of the moving window used to apply an average on the XIC intensities. Use this filter in rare cases of very noisy signal.
- Anti-spike filter: removes spikes on HR analysis.
 - **half (natural integer)**: half size of the moving window (in scan points) used to eliminate a spike. An intensity value is considered as a spike if it is the only value greater than 0 among the other values of the window. A good value is usually 5.

The Moulon peak-detection parameters

- **smoothing_point (natural integer)**: the half window size used to compute the average intensity in the filter preceding peak detection;
- **TIC_start**: intensity value representing the starting timestamp of peak detection, i.e. at this intensity value we start the “chronometer” and begin searching forward (in increasing retention time order) on the signal for local maximums.
- **TIC_stop**: intensity value representing the ending timestamp of peak detection, i.e. at this intensity value we stop the “chronometer” and consider the local maximum found since TIC_start as a peak.

The Zivy peak-detection parameters

- **mean_filter_half_edge (natural integer)** : half window size used to compute the average intensity in the smoothing filter preceding peak detection; this filter is used only for detection purpose, the original signal is not altered. A good value is 1 or 2.
- **minmax_half_edge (natural integer)**: the half window size used to apply the close (min/max) transform on the XIC intensities. This window determines the number of scan points over which two peaks will be considered separately, otherwise they would have been merged. A good half window value is usually 3 (which makes a window of 7).
- **maxmin_half_edge (natural integer)**: same as above but for the close (max/min) transform. This window determines the minimum peak width (in scan points number) below which the peak would not be detected. A good half window value is usually 2 (which makes a window of 5).
- **detection_threshold_on_max (intensity value)**: threshold on the close signal: a minimum intensity value below which peaks are not detected on the closing signal. This threshold is usually two or three times the background noise level (this latter depends on your mass spectrometer).
- **detection_threshold_on_min (intensity value)**: threshold on the open signal: a minimum intensity value below which peaks are not detected. It corresponds to the opening signal upper limit and it represents the background signal upper level. A good value would thus be slightly bigger than your background noise.

The *mode* parameter

The *mode* parameter in the *quantify* element indicates the computation mode of the retention time of peptides and isotopes. This retention time is the one used for peak matching. It can be:

- **real_or_mean**: if the peptide is identified in a run its retention time is the observed one; if not it is the mean of its retention times in the runs of the group (being quantified) in which this peptide was identified.
- **mean** : the retention time of the peptide in a run is always the mean retention time of its retention times in all the runs (of the current group) where this peptide was identified (whether the peptide is identified in this run or not).

Appendices

Appendix A

masschroqML complete input example file

```
1 <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2 <masschroq>
3 <rawdata>
4 <data_file id="samp0" format="mzxml" path="bsa1.mzXML" type="centroid"/>
5 <data_file id="samp1" format="mzxml" path="bsa2.mzXML" type="profile"/>
6 <data_file id="samp2" format="mzml" path="/home/user/bsa3.mzml"
   type="profile"/>
7 <data_file id="samp3" format="mzml" path="/home/user/bsa4.mzml"
   type="profile"/>
8 </rawdata>
9 <groups>
10 <group data_ids="samp0 samp1" id="G1"/>
11 <group data_ids="samp2 samp3" id="G2"/>
12 </groups>
13 <peptide_files_list>
14 <peptide_file data="samp0" path="bsa1_peptides.txt"/>
15 <peptide_file data="samp1" path="bsa2_peptides.txt"/>
16 <peptide_file data="samp2" path="bsa3_peptides.txt"/>
17 <peptide_file data="samp3" path="bsa4_peptides.txt"/>
18 </peptide_files_list>
19 <protein_list>
20 <protein desc="conta|P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
   id="P1.1"/>
21 <protein desc="conta|P02770|ALBU_RAT SERUM ALBUMIN PRECURSOR."
   id="P1.2"/>
22 </protein_list>
23 <peptide_list>
```

54 APPENDIX A. MASSCHROQML COMPLETE INPUT EXAMPLE FILE

```
24 <peptide id="pep0" mh="1463.626" mods="114.08" prot_ids="P1.1"
    seq="TCVADESHAGCEK">
25 <observed_in data="samp0" scan="655" z="2"/>
26 <observed_in data="samp1" scan="798" z="2"/>
27 </peptide>
28 <peptide id="pep1" mh="1103.461" mods="57.04" prot_ids="P1.1"
    seq="ADESHAGCEK">
29 <observed_in data="samp3" scan="663" z="2"/>
30 </peptide>
31 </peptide_list>
32 <isotope_label_list>
33 <isotope_label id="iso1">
34 <mod at="Nter" value="28.0"/>
35 <mod at="K" value="28.0"/>
36 </isotope_label>
37 <isotope_label id="iso2">
38 <mod at="Nter" value="32.0"/>
39 <mod at="K" value="32.0"/>
40 </isotope_label>
41 </isotope_label_list>
42 <alignments>
43 <alignment_methods>
44 <alignment_method id="my_ms2">
45 <ms2>
46 <ms2_tendency_halfwindow>10</ms2_tendency_halfwindow>
47 <ms2_smoothing_halfwindow>5</ms2_smoothing_halfwindow>
48 <ms1_smoothing_halfwindow>3</ms1_smoothing_halfwindow>
49 </ms2>
50 </alignment_method>
51 <alignment_method id="my_obiwarp">
52 <obiwarp>
53 <lmat_precision>1</lmat_precision>
54 <mz_start>500</mz_start>
55 <mz_stop>1200</mz_stop>
56 </obiwarp>
57 </alignment_method>
58 </alignment_methods>
59 <align group_id="G1" method_id="my_ms2" reference_data_id="samp0"/>
60 <align group_id="G2" method_id="my_obiwarp" reference_data_id="samp2"/>
61 </alignments>
62 <quantification_methods>
63 <quantification_method id="my_qzivy">
64 <xic_extraction xic_type="sum">
```

```
65 <mz_range max="1.5" min="0.5"/>
66 </xic_extraction>
67 <xic_filters>
68 <anti_spike half="5"/>
69 <background half_mediane="5" half_min_max="15"/>
70 <smoothing half="3"/>
71 </xic_filters>
72 <peak_detection>
73 <detection_zivy>
74 <mean_filter_half_edge>1</mean_filter_half_edge>
75 <minmax_half_edge>3</minmax_half_edge>
76 <maxmin_half_edge>2</maxmin_half_edge>
77 <detection_threshold_on_max>5000 </detection_threshold_on_max>
78 <detection_threshold_on_min>3000 </detection_threshold_on_min>
79 </detection_zivy>
80 </peak_detection>
81 </quantification_method>
82 <quantification_method id="my_qmoulon">
83 <xic_extraction xic_type="max">
84 <ppm_range max="1.5" min="0.5"/>
85 </xic_extraction>
86 <xic_filters>
87 <background half_mediane="5" half_min_max="15"/>
88 </xic_filters>
89 <peak_detection>
90 <detection_moulon>
91 <smoothing_point>3</smoothing_point>
92 <TIC_start>5000</TIC_start>
93 <TIC_stop>3000</TIC_stop>
94 </detection_moulon>
95 </peak_detection>
96 </quantification_method>
97 </quantification_methods>
98 <quantification>
99 <quantification_results>
100 <quantification_result output_file="result1" format ="tsv"/>
101 <quantification_result output_file="result2" format="gnumeric"/>
102 <quantification_result output_file="result3" format ="xhtmltable"/>
103 <quantification_result output_file="result4" format ="xml"
    xic_traces="true"/>
104 </quantification_results>
105 <quantification_traces>
```

```
106 <peptide_traces peptide_ids="pep0 pep1" output_dir="pep_traces"
    format="tsv"/>
107 <all_xics_traces output_dir="all_xics_traces" format="tsv"/>
108 <mz_traces mz_values="634.635 449.754 552.234" output_dir="mz_traces"
    format="tsv"/>
109 <mzrt_traces output_dir="mzrt_traces" format="tsv">
110 <mzrt_values>
111 <mzrt_value mz="732.317" rt="230.712"/>
112 <mzrt_value mz="575.256" rt="254.788"/>
113 </mzrt_values>
114 </mzrt_traces>
115 </quantification_traces>
116 <quantify id="q1" withingroup="G1" quantification_method_id="my_qzivy">
117 <peptides_in_peptide_list mode="real_or_mean"/>
118 </quantify>
119 <quantify id="q2" withingroup="G2" quantification_method_id="my_moulon">
120 <peptides_in_peptide_list mode="real_or_mean" isotope_label_refs="iso1
    iso2"/>
121 <mz_list>732.317 449.754 552.234 464.251 381.577 569.771
    575.256</mz_list>
122 <mzrt_list>
123 <mzrt mz="732.317" rt="230.712"/>
124 <mzrt mz="575.256" rt="254.788"/>
125 </mzrt_list>
126 </quantify>
127 </quantification>
128 </masschroq>
```

masschroq_complete_input_example.xml

Appendix B

Peptide identification example tsv file

```
1 scan sequence mh z proteins mods
2 778 CCTKPESER 1166.4934 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
  114.08
3 839 NYQEAK 752.3585 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR." Ymod
4 1136 TCVADESHAGCEK 1463.5852 2 "P02769|ALBU_BOVIN SERUM ALBUMIN
  PRECURSOR." 57.04
5 1585 SHCIAEVEK 1072.5111 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
6 1935 NECFLSHK 1034.4729 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
7 1960 ECCDKP LLEK 1291.6019 3 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
8 1980 LCVLHEK 898.48236 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
9 2089 CCTESLVNR 1138.4973 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
10 2237 YICDNQDTISSK 1443.6373 2 "P02769|ALBU_BOVIN SERUM ALBUMIN
  PRECURSOR."
11 2241 QNCDQFEK 1051.4155 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
12 2278 ETYGMADCCEK 1478.5223 2 "P02769|ALBU_BOVIN SERUM ALBUMIN
  PRECURSOR."
13 2702 QEPERNECFLSHK 1656.7439 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
  PRECURSOR."
14 2713 EYEATLEEECAK 1502.61 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
15 2753 ECCHGDLLECADDR 1749.6615 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
  PRECURSOR."
16 2867 CCAADDKEACFAVEGPK 1927.797 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
  PRECURSOR."
17 2910 EACFAVEGPK 1107.5137 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
18 3267 DDPHACYSTVFDK 1554.6533 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
  PRECURSOR."
19 3494 YLYEIAR 927.49396 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
```

```
20 3629 RHPEYAVSVLLR 1439.8123 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
21 3818 LKPDPTLCDEFK 1576.7699 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
22 3821 KVPQVSTPTLVEVSR 1639.9381 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
23 3870 KQTALVELLK 1142.715 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
24 4082 RHPEYAVSVLLR 1439.8177 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
25 4236 RPCFSALTPDETYVPK 1880.9225 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
26 4253 SLHTLFGDELCK 1419.6945 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
27 4460 LVNELTEFAK 1163.6317 2 "P02769|ALBU_BOVIN SERUM ALBUMIN PRECURSOR."
28 4499 SLHTLFGDELCK 1419.6956 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
29 4972 RHPYFYAPPELLYYANK 2045.0253 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
30 5084 LFTFHADICTLPDTEK 1907.9092 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
31 5308 LGEYGFQNALIVR 1479.807 2 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
32 5578 HPYFYAPPELLYYANK 1888.9224 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
33 6015 TVMENFVAFVVK 1399.6898 2 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
34 6689 MPCTEDYLSLILNR 1724.8433 3 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
35 6788 DAFLGSFLYEYSR 1567.748 2 "P02769|ALBU_BOVIN SERUM ALBUMIN
    PRECURSOR."
```

peptide_example_tsv_file.txt

Appendix C

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

<<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or “**History**”.) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use

the same title as a previous version if the original publisher of that version gives permission.

- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar

in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

the License in the document and put the following copyright and license notices just after the title page:

Bibliography

- [GLB00] J. Goutsias, L. Vincent, and D.S. Bloomberg. *Mathematical Morphology and its Applications to Image and Signal Processing*. Kluwer Academic Publishers, 2000.
- [LHS87] J. Lee, R. Haralick, and L. Shapiro. Morphologic edge detection. *IEEE Journal of Robotics and Automation*, 1987.
- [LL88] F. Leymarie and M. D. Levine. Curvature morphology. Technical Report TR-CIM-88-26, CIM McGill University, Montreal, Canada, December 1988.
- [Mar04] P. Maragos. *Handbook of Image and Video Processing*, chapter 3.3 Morphological filtering for image enhancement and feature detection, pages 135–156. Elsevier, 2nd edition, 2004.
- [Mey78] F. Meyer. Contrast feature extraction. *Special Issues of Practical Metallography*, 8:374–380, 1978.
- [PM06] J.T. Prince and E. Marcotte. Chromatographic alignment of ESI-LC-MS proteomics data sets by ordered bijective interpolated warping. *Analytical Chemistry*, 78(17):6140–6152, 2006.
- [Ser82] J. Serra. *Image Analysis and Mathematical Morphology*, volume I. Academic Press, London, 1982.
- [VLNZ11] B. Valot, O. Langella, E. Nano, and M. Zivy. Masschroq: A versatile tool for mass spectrometry quantification. *Proteomics*, 2011. in press.