Datacube Explorer

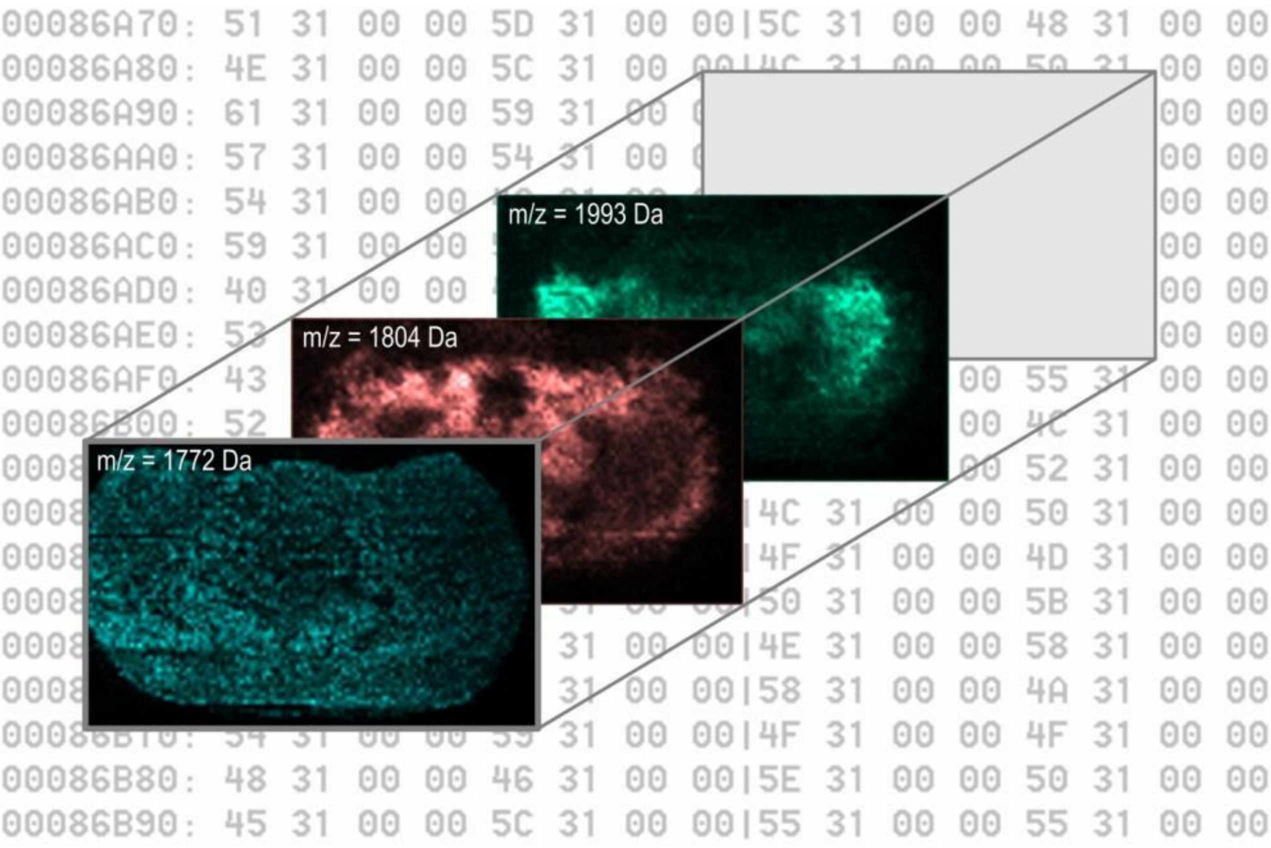
User Manual

Including:

Software Installation Instructions

Free Software Licensing

Datacube File Format Description



AMOLF, http://www.amolf.nl

Title:	Manual Datacube Explorer version 2.1
Author:	Ivo Klinkert
Date:	September 2017

TABLE OF CONTENTS

1. INTRODUCTION

This document is the user manual for the 'Datacube Explorer' software. This software application can be used to visualize and analyse mass spectrometry imaging datasets, both spatially and spectrally. The first version of the software was written back in 2007 and the software evolved over the years to serve the visualisation needs for the biomolecular imaging mass spectrometry (BIMS) group of Ron Heeren at the FOM Institute AMOLF in The Netherlands (http://www.amolf.nl). Especially with the development of new MSI instruments, custom build visualization tools are necessary because often no commercial visualization software is available for these new experimental cases.

The Datacube Explorer is capable of visualizing the following file formats:

- The Datacube format (developed at the FOM Institute AMOLF).
- The imzML data format [1-3].
- The Biomap (Analyze 7.5) file format.

Both this manual and the Datacube Explorer software application are written at the FOM Institute AMOLF. The Datacube Explorer has already been used in many scientific publications [4-13].

The process of generating 'datacube' datasets from different data sources is not described in this document. However, the required data format description to generate datacubes is available in Chapter 5.

1.1 Document Structure

The document first gives a short instruction about how to install the Datacube Explorer software (Chapter 2). Second, the document explains the free licensing system (Chapter 3). It mentions how to obtain a free licence key, and how to install this licence key. The main part of this document is the user manual of the Datacube Explorer software, which explains all features of the software (Chapter 4). The last part of this document is the datacube open file format description (Chapter 5). This opens the possibility to generate your own datacubes, and use the Datacube Explorer to visualize them.

1.2 Acknowledgements

The work on the development of the Datacube Explorer was carried out in the context of the Virtual Laboratory for e-science project. This project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ). This work is also part of the research program of the "Stichting voor Fundamenteel

Onderzoek der Materie (FOM)," which is financially supported by the "Nederlandse organisatie voor Wetenschappelijk Onderzoek (NWO)". The work on the implementation of imzML is part of the Computis Program, 6th European Framework Program for Research and Technological Development (FP6), project no. LSHGCT-2005-5181194.

1.3 Citations

When the use of the Datacube Explorer software application has directly or indirectly contributed to the content of your publication or presentation, please include a citation to the following article [14]: I. Klinkert, K. Chughtai, S.R. Ellis, and R.M.A. Heeren, *Methods for full resolution data exploration and visualization for large 2D and 3D mass spectrometry imaging datasets*, International Journal of Mass Spectrometry, 2014. 362: p. 40-47., http://dx.doi.org/10.1016/j.ijms.2013.12.012.

2. SOFTWARE INSTALLATION

2.1 System Requirements

The Datacube Explorer application requires the Microsoft .NET framework version 4.51 or higher to run. This version is freely available from Microsoft, and you will be asked by the installation program to install when missing.

To load large datacube files just as much (or more) system memory must be available to view the file at a reasonable speed. As from version 2.0 of the Datacube Explorer no software limitations on the data size exist anymore (only physical memory and other software resources that claim computer memory are the limitations).

2.2 Software Download

The Datacube Explorer software version 2.1 is available as executable. It is distributed as a .exe installation file.

The installation file can be downloaded from the AMOLF download page, http://www.amolf.nl/download/datacubeexplorer/. From here also two example datacube datasets are available. Example imzML datasets are available from http://www.imzml.org.

2.3 Installation

Run the ‘DatacubeExplorerSetup v2.1.exe’ file, which performs the installation. The application can be started from the Start menu: Programs – FOM Institute AMOLF – DatacubeExplorerV2.1 – Datacube Explorer.

Users working at the FOM Institute AMOLF who installed MsTools, can start the application via Programs – FOM Institute AMOLF – MsTools – DatacubeExplorer.

The application can be removed by using the 'Add or Remove Programs' function in MS Windows.

3. SOFTWARE LICENSING

In order to use the Datacube Explorer software, a licence key is required to activate the software for non-commercial usage.

3.1 Obtaining a Licence Key

No new licence keys are issued, but as of January 2015 a free licence key is available:

Organization & Name: **Free non-commercial community licence**
Licence Key: **DceKey2319614510214265181190**

Please be sure you enter **both** the “Organization & Name” **and** the “Licence Key” as mentioned above!

3.2 Software Activation

The key can be entered on the 'Software Registration Form' (Menu -> Tools -> Registration -> Enter Registration). The key will be stored on your system after it has been entered.

When a new key is entered, an existing key will be overwritten for the currently logged in user. Licence keys are stored per user in Windows, so when logging in as a different user on a system, a new licence key is required (this also opens the possibility of installing different licence keys for the Datacube Explorer for different users).

A licence key can be removed from a system via the 'Remove Registration' function (Menu -> Tools -> Registration -> Remove Registration) in the software.

The obtained licence key is for personal use only and is non-transferable. When entering the licence code you agree to the following statements:

```
Copyright (c) 2008-2017, AMOLF

Permission to use this software for any non-commercial purpose without fee is hereby
granted, provided that a personal licence key has been obtained from AMOLF.

THE SOFTWARE IS PROVIDED "AS IS" AND AMOLF DISCLAIMS ALL WARRANTIES WITH REGARD TO
THIS SOFTWARE. IN NO EVENT SHALL AMOLF BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT,
OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA
OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION,
ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
DISTRIBUTION OF THE SOFTWARE OUTSIDE OF YOUR OWN COMPANY OR INSTITUTE IS PROHIBITED.
```

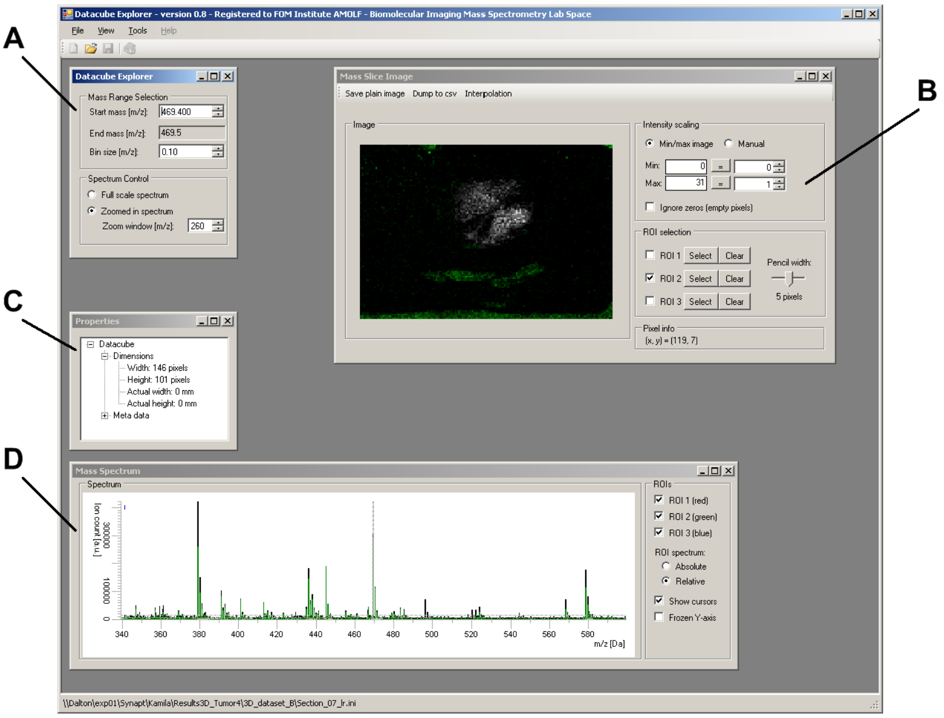
3.3 Datacube Explorer Software Updates

When an update of the Datacube Explorer application will be installed, the licence key will stay functional.

When the Datacube Explorer application will be uninstalled, the licence key will also stay available. When the Datacube Explorer needs to be uninstalled, and the licence has to be uninstalled too, use the 'Remove Registration' function first, before uninstalling the Datacube Explorer software.

4. DATACUBE EXPLORER

The Datacube Explorer software application is a tool for visualizing mass spectrometry imaging (MSI) datasets. The focus of the visualization is on image-based data analysis, with spectral information also available. Figure 4.1 shows a typical screen view of the application.



Datacube Explorer - version 0.8 - Registered to FOM Institute AMOLF - Biomolecular Imaging Mass Spectrometry Lab Space

Datacube Explorer

Mass Range Selection
Start mass [m/z]: 469.400
End mass [m/z]: 469.5
Bin size [m/z]: 0.10

Spectrum Control
○ Full scale spectrum
● Zoomed in spectrum
Zoom window [m/z]: 260

Mass Slice Image

Intensity scaling
● Min/max image ○ Manual
Min: 0 = 0
Max: 31 = 1
□ Ignore zeros (empty pixels)

ROI selection
□ ROI 1
☑ ROI 2
□ ROI 3
Pencil width: 5 pixels

Pixel info
(x, y) = (119, 7)

Properties

Datacube
Dimensions
Width: 146 pixels
Height: 101 pixels
Actual width: 0 mm
Actual height: 0 mm
Meta data

Mass Spectrum

ROIs
☑ ROI 1 (red)
☑ ROI 2 (green)
☑ ROI 3 (blue)
ROI spectrum:
○ Absolute
● Relative
☑ Show cursors
□ Frozen Y-axis

\\Dalton\exp01\Synapt\Kamila\Results3D_Tumor4\3D_dataset_B\Section_07_lr.ini

Figure 4.1: A dataset loaded into the Datacube Explorer

The main window contains four sub windows, as shown in Figure 4.1, with the following functionalities:

A – "Datacube Explorer": Selection of the mass range to display.

B – "Mass Slice Image": Image of the ion intensities of a selected m/z window.

C – "Properties": Meta data of the current dataset.

D– "Mass Spectrum": Summed spectrum of the entire dataset and the spectra of the regions-of-interests.

Paragraph 4.2 describes these functionalities in detail. First in paragraph 4.1 an explanation about the data format of the 'datacube' is given.

4.1 Datacube Data Format

The Datacube Explorer is a software application for visualizing MSI datasets, in this case so-called datacubes. A description of this datacube data format is given in this paragraph, where a full syntax specification can be found in Chapter 5.

4.1.1 Standard Datacube

A datacube dataset contains MSI data organized using three dimensions. The first two dimensions are the x and y-coordinates of the measured sample, and the third dimension is the m/z axis where the mass spectral information is stored. All positions in the datacube contain ion intensity information (e.g. ion counts), which can be considered the fourth dimension.

The actual x and y coordinates are the measurement positions, where every position contains a mass spectrum. These positions are called pixels in the datacube, as they are shown as pixels in the selected ion images. The spectral data of a dataset is stored in mass bins in a datacube, where all ion intensities in a very small mass range (the mass bin) are summed together. Therefore, the coordinate in the m/z dimension is the mass bin index, as the m/z axis is organized into fixed size mass bins. For example, when 0.01 Da sized bins are used, the mass resolution of that datacube is also 0.01 Da.

The data inside the datacube is stored in an image-based order, which is optimized for the retrieval of images for masses (or mass ranges) of interest. Therefore, the Datacube Explorer allows fast scrolling through different images from the entire dataset.

A datacube dataset actually consists of two files, one file contains all measured data whereas the other file contains the properties (or meta data) that define the datacube. These two corresponding files have the same file name, but different file extensions. The properties file has the '.ini' file extension, the data file the '.dat' file extension. When copying or moving a datacube, these two files should always be copied or moved together.

4.1.2 Mosaic Datacube

The use of a single datacube is in some cases insufficient to represent a single dataset. This can for example be the case when one single dataset becomes too large to fit into computer memory, if the required mass bin size is chosen to be very small and results in a large number of mass resolved images, or, when the time to convert the source data into datacubes takes so much time that parallel data processing must applied to the conversion process. In the single cube format, the achievable minimum mass bin size is limited by computer memory resources. To overcome these problems, the so-called mosaic datacube architecture [13, 14] enables the option of distributing the large amount of data among several datacubes. The term mosaic originates from the fact that individual datacubes in this architecture are organized in a mosaic pattern.

A mosaic datacube consists of a set of stand-alone datacubes, a kind of sub-datacubes of the overall datacube. A mosaic pattern is defined in 3D space: A mosaic cube is made from adjacent cubes horizontally and vertically, but also in the m/z direction. The definition of which cubes belong to the mosaic is defined in a separate mosaic definition file (.ini). The positions where the cubes are located in the overall mosaic are defined in the properties files of the individual datacubes.

Both standard datacubes and mosaic datacubes can be opened in the Datacube Explorer in the same way. This because for both types of datacubes, the .ini file should be opened. The Datacube Explorer will recognize a mosaic definition, and will read in the individual cubes and organize them accordingly.

The volume-of-interest (VOI) function (see paragraph 4.2.1) of the Datacube Explorer enables the visualization of sub-volumes of a full mosaic datacube when the entire mosaic datacube is too large to fit into memory. This means that all MSI data in the original datacube is available in full resolution, and every sub-volume of the entire dataset can be visualized with the same high resolution. With this approach computer memory resources are not limiting the visualization of high-resolution MSI data.

4.1.3 Datacube Configuration

As described above, the main data of the datacube is stored in two files, the data file (.dat) and the corresponding properties file (.ini). A third file, the configuration file (.cfg), is optional, and contains user settings of the datacube, made in the Datacube Explorer. The configuration file stores the user-selected region of interests (ROIs), so that the ROI selections are also available when the datacube is reloaded on the same, or another PC.

A configuration can be stored any time by using the 'Save Configuration' option from the 'Tools' menu. When a configuration is not saved, or is changed since it was last saved, the Datacube Explorer will ask if the configuration must be saved to prevent losing the latest configuration data.

The configuration file should have the same file name as the datacube, with the .cfg extension. By manually renaming the configuration file names, multiple configuration files per datacube can be created.

4.1.4 ImzML Support

The Datacube Explorer also supports the visualization of imzML datsets [1-3]. imzML is a common data format for mass spectrometry imaging, which is based on the international HUPO mzML standard for mass spectrometry (http://www.psidev.info/index.php?q=node/257) [15].

The latest definition of imzML, as well as some background information, can be found at http://www.imzml.org.

The Datacube Explorer can visualize imzML dataset. Version 1.4 of the Datacube Explorer has some minor limitations to imzML, which are mentioned in the list below.

- Z-position of spectra in the imzML file is ignored.
- It is assumed that the imzML file and the idb file have the same name, and are stored in the same directory.
- No check is performed if x and y coordinates fit into the specified x and y dimensions.
- Visualization is designed for imzML/mzML version 1.1 only.
- Only one scan per file is supported.
- Assumption: X and Y position will never be in a referenceable parameter group.
- Assumption: Data type within m/z arrays are the same for all spectra.
- Assumption: Data type within intensity arrays are the same for all spectra.
- Assumption: m/z array comes first, followed by the intensity array.
- Only uncompressed data can be read.

4.1.5 Biomap File Support

The Datacube Explorer also supports the visualization of Biomap datasets (Analyze 7.5 file format), which are widely used in the mass spectrometry imaging community.

4.2 Functionalities

How datasets can be opened is described in 4.2.1, how datasets can be visualized and analyzed is described in 4.2.2.

4.2.1 Opening Datasets

Datasets, both datacubes and imzML, can be opened in the Datacube Explorer in four different ways:

1. By using the 'Open' function (Menu -> File -> Open) in the application.
2. By using the 'Open VOI' function (Menu -> File -> Open VOI) in the application.
3. By using the drag-and-drop functionality.
4. By using the command line parameters at application start up.

Reading in the data can take a few seconds to a few minutes, depending on the dataset file size and possibly the network speed. A progress bar gives an indication about the reading progress of the dataset. One dataset at the same time can be loaded (including a mosaic datacube). When a new dataset is opened, the current one is closed (but not deleted).

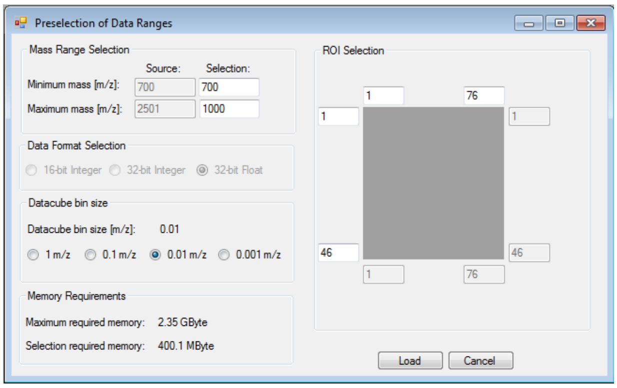
Open function

Datacubes can be opened using Menu -> File -> Open (Ctrl+O or the File Open icon). Select the .ini file which belongs to the datacube (there is also a corresponding .dat file in the data cube dataset, but this is not shown in the Open dialog box). ImzML datasets can be read into the Datacube Explorer, by selecting the .imzml file of the imzML dataset, while Biomap datasets can be read by selecting the .img file of the Biomap dataset.

Mosaic datacubes can be opened like standard datacubes. By opening the mosaic definition file, which also has the extension .ini, the application automatically composes a new datacube out of the individual datacubes.

Open VOI function

With the 'Open VOI' function, a so-called volume-of-interest (VOI) can be loaded. The VOI selection can be used to load a portion of the single datacube file, or parts from a set of mosaic datacubes. The VOI selection is especially useful when the original dataset would not fit into the memory of the PC entirely, but a selected VOI would.



Mass Range Selection

	Source:	Selection:
Minimum mass [m/z]:	700	700
Maximum mass [m/z]:	2501	1000

Data Format Selection

○ 16-bit Integer ○ 32-bit Integer ● 32-bit Float

Datacube bin size

Datacube bin size [m/z]: 0.01

○ 1 m/z ○ 0.1 m/z ● 0.01 m/z ○ 0.001 m/z

Memory Requirements

Maximum required memory: 2.35 GByte

Selection required memory: 400.1 MByte

ROI Selection: 1, 76, 1, 46 (selection); 1, 76, 1, 46 (source)

Figure 4.2: Preselection of the Volume-of-interest

In Figure 4.2, information about the preselection options is shown. For both the mass range and the ROI, a selection can be made, where the values in the grey background indicate the original values, and the values in the editable white background are the user selected values. In the lower-left corner the memory requirements of the cube are shown every time the VOI selection is changed.

For datacubes, the mass bin size stays fixed to the original bin size of the datacube. For both imzML and Biomap files, the mass bin size is user-selectable between 1 and 0.001 m/z.

A VOI selection can be any spatial or spectral volume, independent of the boundaries of the individual mosaic datacubes. After a VOI selection has been read into memory, it is handled as a new datacube by the Datacube Explorer. Therefore, this new datacube is not saved to disk yet.

The preselection of the data ranges is available for standard datacubes, mosaic datacubes, and imzML and Biomap datasets.

Drag-and-drop
A user can use the drag-and-drop function to open a dataset. By dragging and dropping a datacube (or mosaic) .ini file, a .imzML file, or a Biomap .img file into the Datacube Explorer application window, that particular dataset will be opened.

Command line parameters
The Datacube Explorer application ('Datacube Explorer.exe') can also be started from the command line. By adding the location and name of the dataset to load, the application will start and directly load the specified dataset.

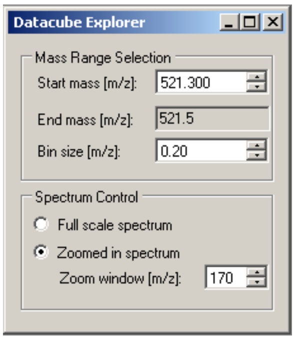
4.2.2 Dataset Analysis

The application displays the information in four different windows:

1. The Datacube Explorer Window
2. The Mass Slice Image Window
3. The Mass Spectrum Window
4. The Properties Window

4.2.2.1 Datacube Explorer Window

The 'Datacube Explorer' window (Figure 4.3) let you scroll through the masses of the dataset. The 'Mass Range Selection' lets you select the mass range (spectrally binned) to display in the 'Mass Slice Image' window. The 'Start mass' and the (spectral) 'Bin size' can be changed, the 'End mass' will be set to the corresponding value automatically. Values can be typed in, or changed using the small up/down buttons at the right side of the values.



Mass Range Selection
Start mass [m/z]: 521.300
End mass [m/z]: 521.5
Bin size [m/z]: 0.20

Spectrum Control
○ Full scale spectrum
● Zoomed in spectrum
Zoom window [m/z]: 170

Figure 4.3: Datacube Explorer Window

To make scrolling through the dataset more convenient, you can use the following short cut keys (only when the focus is in the 'Start mass' field):

SHIFT + Arrow Left/Right	Up/down 0.001 Da.
CTRL + Arrow Left/Right	Up/down 0.005 Da.
Arrow Left/Right	Up/down 0.01 Da.
Arrow Up/Down	Up/down 0.1 Da.
Page Up/Down	Up/down 1 Da.
Home/End	Up/down 10 Da.

Also see the description of the 'Mass Spectrum Window' where mass selection by means of the mass spectrum is explained.

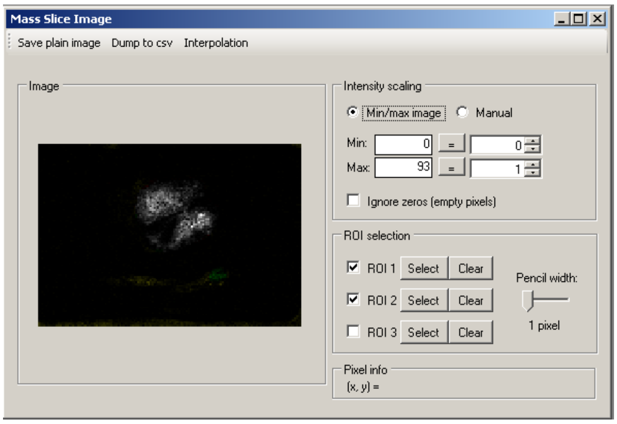
When entering the window with your mouse, the program automatically sets the focus to the 'Start mass' field, so you do not have to click in the 'Start mass' field to activate the use of the short cut keys.

The 'Spectrum Control' controls the viewed mass range in the 'Mass Spectrum' window:

Full scale spectrum The full range of the dataset is used on the x axis.
Zoomed in spectrum A zoomed in part around the selected mass is used on the x axis.
Zoom window [m/z] The zoom window, when 'Zoomed in spectrum' is selected.

4.2.2.2 Mass Slice Image Window

In the 'Mass slice image' window (Figure 4.4) the selected mass range image is displayed. All spectral values in the selected mass range are binned together. White means high signal, black means no signal. A linear scaling is applied to the intensities from black to white. You can resize the window, which resizes the image on the screen.



● Min/max image ○ Manual
Min: 0 = 0
Max: 93 = 1
□ Ignore zeros (empty pixels)
☑ ROI 1
☑ ROI 2
□ ROI 3
Pencil width: 1 pixel
(x, y) =

Figure 4.4: Mass Slice Image Window

The ‘Intensity scaling’ controls the intensity of the image:

Min/max image — Image intensity is scaled up between the minimum and the maximum ion count values. This results always in the maximum black-to-white intensity range, but gives no information about the absolute ion intensity.

Manual — Image intensity is scaled up between the minimum and maximum ion intensities specified in the 'Min' and 'Max' fields. Ion count values above Max will be white, values below Min will be black.
The values displayed on the left indicate the minimum and maximum values that would have been used in the 'Min/max image' mode. These values can be copied to the manual field by using the '=' buttons.

Ignore zeros — In the case when 'Min/max image' scaling is used for datasets for which not all pixels contain data, the minimum intensity value will always be zero (because there is no intensity available for a pixel which is not measured), decreasing the dynamic range of the image intensity of the image . When 'Ignore zeros' is selected, the minimum value for 'Min/max image' scaling is the 'minimum value above zero'. This results in a maximum dynamic range.
For datasets where all pixels contain data, no difference will be observed when selecting and deselecting this option.

The 'Pixel info' box shows the coordinates of a pixel (for example '(x,y) = 10, 34') when the mouse pointer is pointer on top of a pixel. No additional clicking in the image is required.

The 'Mass Slice Image' window has the following menu options:

Save plain image — An image of the currently displayed image is saved, in the PNG image format. The stored image is independent of the size on the screen. The image is stored without any interpolation, also when it *is* applied to the image on the screen.

Dump to csv — The image data is dumped into a CSV (comma separated values) file.

Interpolation — Is used to apply an interpolation mode (also called smoothing) to the image on the screen. You can choose between the following modes, based on the available interpolation modes of the Microsoft .Net framework (see ‘InterpolationMode Enumeration’ of .Net): 'Nearest neighbour' (normal mode, pixels are visualized as squares, independent of the zoom rate of the image), 'Interpolation', 'Bilinear' and ‘Bicubic' (best quality, smoothest, but slowest interpolation algorithm).

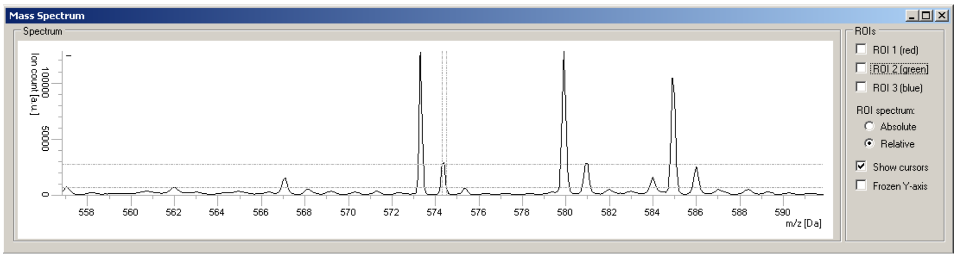
A description of the region-of-interest (ROI) functionality is described after the 'Mass Spectrum Window' section.

4.2.2.3 Mass Spectrum Window

In the 'Mass Spectrum' window the spectrum is displayed (Figure 4.5). This is the summed spectrum over all pixels of the image. You can resize the window to resize the spectrum on the screen.

The mass spectrum contains two cursors, which indicate the start and the end value of the selected mass. With the 'Show cursors' option, these cursors can be enabled or disabled.

The 'Frozen Y-axis' option will enable a fixed Y-axis range, at the time of enabling.



☐ ROI 1 (red)
☐ ROI 2 (green)
☐ ROI 3 (blue)

ROI spectrum:
○ Absolute
● Relative

☑ Show cursors
☐ Frozen Y-axis

Figure 4.5: Mass Spectrum Window

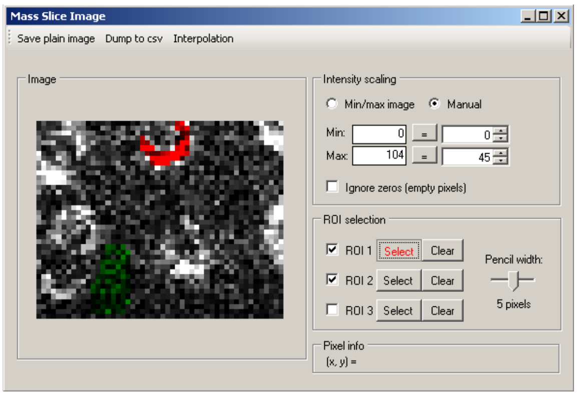
The selected mass can also be chosen with a single mouse click in the spectrum. An image with this selected mass image will be displayed in the Mass Slice Image Window. A single click will centre the spectrum around the selected mass, where as the single mouse click is performed while the control (Ctrl) key is down, the spectrum view will not be re-centred.

When zoomed in into a part of the full spectrum of the dataset, moving to an adjacent part of the spectrum is possible by clicking at the left or right part of the window, just outside the spectrum. At the point when a mouse click would cause a move to the next adjacent part of the spectrum, a text “Spectrum Up” or “Spectrum Down” will appear.

Region of interest (ROI) functionality

The ROI functionality enables you to show the spectra of a selected area in the image. A ROI can be selected in the ‘Mass Slice Image’ window (Figure 4.6). Pressing the ‘Select’ button activates the selection of the corresponding ROI. Moving the mouse, with the left mouse key down, over the area, selects the ROI. Clicking one time selects/deselects one pixel for the ROI. The ‘Clear’ buttons clears the corresponding ROI selection. The 'Pixel width' option can be used to select more than one pixel at a time, where a selection of 1x1, 3x3, 5x5, 7x7 or 9x9 pixels at a time can be made.

The ROI checkboxes show/hide the corresponding ROI, both on the ‘Mass Slice Image’ window as well as in the ‘Mass Spectrum’ window. Hiding does not mean deleting.



○ Min/max image ● Manual

Min: 0 = 0

Max: 104 = 45

□ Ignore zeros (empty pixels)

☑ ROI 1

☑ ROI 2

□ ROI 3

Pencil width: 5 pixels

(x, y) =

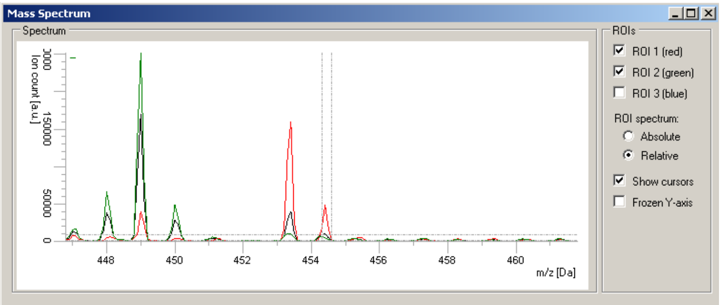
Figure 4.6: Selection of the Region-of-Interest

Up to three different ROI areas are available, displayed in different colours with the intensity of the original (white) pixel.

ROI 1	Red
ROI 2	Green
ROI 3	Blue
Total image	Black (only applicable in the spectrum)

When two or three ROIs overlap in the image, the colours of these ROIs are mixed (by adding their RGB colours together), with the colour intensities scaled to the relative intensity for every individual pixel.

The ROI intensity in the spectrum can be shown as absolute value (option ‘Absolute’) or relative (option ‘Relative’), see Figure 4.7. Relative means the ion intensity is corrected relative to the area for the ROI (if 5% of the image is selected for that ROI, the ion intensity is multiplied by 20).



ROIs

- ☑ ROI 1 (red)
- ☑ ROI 2 (green)
- □ ROI 3 (blue)

ROI spectrum:

- ○ Absolute
- ● Relative

- ☑ Show cursors
- □ Frozen Y-axis

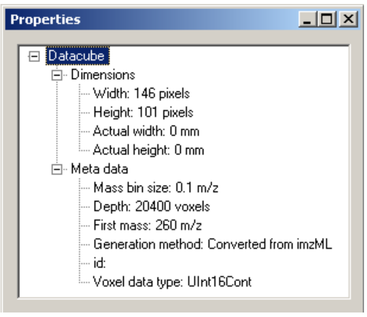
Figure 4.7: Region-of-interest as shown in the spectrum

4.2.2.4 Properties Window

In the 'Properties' window all meta information about the datacube is displayed (Figure 4.8). For imzML datasets also all imzML meta data can be explored directly.

Via Menu -> Tools -> Edit Properties, some of these properties are user editable. The user editable properties of a datacube can be added, modified and/or saved. The following user editable properties are available:

Actual width [mm]	The width of the sample in mm
Actual Height [mm]	The height of the sample in mm
Description	Any description to store together with the datacube
Generation method	Information about how the datacube was generated
Id	An identifier for the cube.



```
⊟ Datacube
   ⊟ Dimensions
      Width: 146 pixels
      Height: 101 pixels
      Actual width: 0 mm
      Actual height: 0 mm
   ⊟ Meta data
      Mass bin size: 0.1 m/z
      Depth: 20400 voxels
      First mass: 260 m/z
      Generation method: Converted from imzML
      id:
      Voxel data type: UInt16Cont
```

Figure 4.8: Properties Window

4.2.2.5 Main Window

The main window menu contains a number of items that are not mentioned above:

Menu -> File -> Save As

Stores the data cube data file under a different name, and/or stores it in another data format, when possible. Only loss-less conversions are possible.

Menu -> Tools -> Save Configuration

Saves the configuration (the ROI definitions) of the datacube.

Tools - Cube Format Analyser

Analyses the current data cube dataset, and mentions the most efficient datacube data format to store this dataset.

Menu -> Tools -> Flip Cube

This function will mirror the datacube. 'Left <-> Right' will mirror the pixels in such a way that it changes the horizontal position, but leave the pixels on the same vertical position. 'Top <-> Bottom' will mirror the pixels in such a way that it changes the vertical position, but leave the pixels on the same horizontal position.

For large datacubes, flipping can take some time, please wait until the flip operation has finished. Applying twice the same flip will result in the original datacube again.

After a flip, the datacube has to be saved to file again (the application will ask if you want to save the datacube when closing the Datacube Explorer). The ROIs of a cube will be flipped correspondingly, and need to be saved as well (by saving the configuration, which is also asked when closing the application).

Menu -> Export -> ROI spectrum to CSV file

This function will export the spectral data to a user-specified CSV file in mass-intensity pairs. Spectra from all pixels of a selected region-of-interest (1, 2 or 3) are summed together for this export. The intensity values are corrected for the number of pixels relative to the total amount of pixels (if 5% of the image is selected for that ROI, the ion intensity is multiplied by 20).

In the case when ‘All Pixels’ are selected, spectra from all pixels are summed and exported as one spectrum.

Self-organizing map functionality (Menu -> Tools -> Self-organizing map)

When a dataset has been loaded, a self-organizing map image classification functionality can be performed on the dataset. This analysis is classified as unsupervised competitive learning, and is also known as the Kohonen (neural) network [16]. A detailed description can be found in Chapter 7.

5. DATACUBE FILE FORMAT SYNTAX

5.1 Standard Datacube File Definitions

The Datacube file format has been developed by the FOM Institute AMOLF and is intended to store imaging mass spectrometry data in a uniform way. It is a simple data format, so with this documentation any programmer is able to read the datacube file format, or generate his own datacubes.

The name ‘cube’ refers to the 3-dimensional structure of the data, the first two dimensions are the spatial (x, y), and the third one is the mass (m/z) dimension.

Two-file datacube file format

For performance reasons this format contains two files per ‘datacube’:

- A text based file containing meta data of a datacube, called the properties file.
- A binary file containing the actual data of a datacube, called the data file.

File names

These two corresponding files have the same file name, but with different file extensions. The properties file has the .ini file extension; the data file the .dat file extension.

Content of properties file

A properties file contains multiple property/value pairs. The content of a typical properties file looks like:

```
# File created at 31-10-07 02:03:35 PM
#
Datatype = UInt16Cont
DatacubeDescription = Some useful description
DatacubeMassbinsize = 0.1
DatacubeId = 0
DataCubeGenerationMethod = SIMS converter
DatacubeOffset = 100
DatacubeWidth = 120
DatacubeHeight = 71
DatacubeDepth = 19000
DatacubeActualWidth = 24
DatacubeActualHeight = 14.2
```

- The order in which these properties appear in the file is not important.
- The property names are case-insensitive.
- Lines starting with # indicate comments and are not processed.
- Blank lines are not processed.

The following properties are present in a properties file:

`Datatype`

The data type for one value in the data file. Valid types are:

UInt16Cont	: 16 bit unsigned integers
UInt32Cont	: 32 bit unsigned integers
SingleCont	: 32 bit floating point values

When no datatype property is available, UInt32Cont must be used.

`DatacubeDescription`

Any useful description

`DatacubeMassbinsize`

The m/z distance between two ‘slices’ in the datacube data. The distance between each slice is the same. Floating point value.

`DatacubeId`

Unique identifier of the cube. Not used yet.

`DataCubeGenerationMethod`

Information about how this Datacube was generated.

`DatacubeOffset`

Offset in m/z [Da] of the first value. Integer value.

`DatacubeWidth`

Width of the Datacube. The width is defined in terms of number of positions [# image pixels].

`DatacubeHeight`

Height of the Datacube. The height is defined in terms of number of positions [# image pixels].

`DatacubeDepth`

Depth (range) of the Datacube. The depth in terms of the total number of mass bin slices.

`DatacubeActualWidth`

The real width of the dataset, in mm.

`DatacubeActualHeight`

The real Height of the dataset, in mm.

Content of the data file

The data file contains the raw data. The data represents the intensities for one pixel (x, y) for a specific m/z [Da] mass bin value (Figure 5.1). The integer data types are stored in the network byte order, the high-order bytes first. For the 32 bit floating point the low-order bytes will be written first.

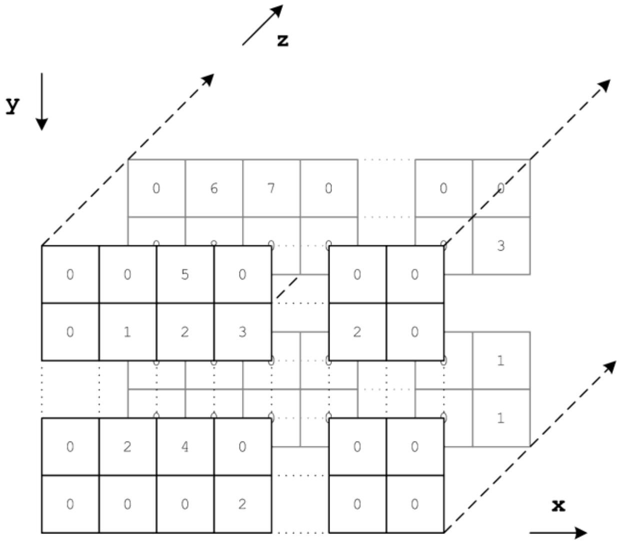


Figure 5.1: Data file organization

The format, in which an intensity value is stored, is defined in the properties file by the `Datatype` property.

Data cubes have the following internal data representation (the increasing numbers indicate the index in the data file, width = 4, height = 3):

```
       depth = 0        depth = 1        depth = 2

          width
    h  0  1  2  3     12 13 14 15      24 .. ..
    e
    i  4  5  6  7     16 17 18 19      ..
    g
    h  8  9 10 11     20 21 22 23
    t
```

The definitions of the X, Y and Z axis are also shown in figure 5.1.

5.2 Mosaic Datacube File Definitions

Each datacube which is part of the mosaic datacube collection is exactly the same as a normal datacube, except for two extra lines of information in the settings (.ini) file. The two extra lines contain the following two definitions

`DatacubeMosaicOffsetX`

Offset in the x direction. The offset is defined in terms of positions.

`DatacubeMosaicOffsetY`

Offset in the y direction. The offset is defined in terms of positions.

The offsets represent the position of the cube in the mosaic. For example, when a datacube is the third cube horizontally, and the two cubes on the left have both a width of 40 pixels, the offset for the third cube will be 80. When the datacube is the first cube in the vertical direction, the offset will be 0. An example of the settings file of this datacube looks like:

```
# File created at 31-05-11 11:22:35 AM
#
Datatype = SingleCont
DatacubeDescription = Maldi experiment on Ftms
DatacubeMassbinsize = 0.1
DatacubeId = 4
DataCubeGenerationMethod = Part of mosaic
DatacubeOffset = 100
DatacubeWidth = 40
DatacubeHeight = 50
DatacubeDepth = 19000
DatacubeMosaicOffsetX = 80
DatacubeMosaicOffsetY = 0
```

As long as the datacubes do not overlap with each other, any mosaic pattern can be defined. In the case of overlapping datacubes, the result will be the summed intensity values of the overlapping cubes, but without any further corrections. Spaces between adjacent datacubes are allowed, but the resulting datacube will contain zero values (because there is no data defined there).

Besides the individual datacube, a mosaic definition file must be present, also with the .ini file extension. This definition file only contains the number of cubes and their names, there is no data file (.dat) associated with the mosaic definition file. An example of a mosaic definition file is shown below:

```
# File created at 31/05/2011 11:00:00 AM
#
NumberOfCubesInMosaic = 5
Cube01 = MaldiFtms01.ini
Cube02 = MaldiFtms02.ini
Cube03 = MaldiFtms03.ini
Cube04 = MaldiFtms04.ini
Cube05 = MaldiFtms05.ini
```

The following 'two' properties must be present in the mosaic definition file:

`NumberOfCubesInMosaic`

The number of cubes in the mosaic definition.

`Cube##`

The name of a cube. Filenames are relative to the location of the mosaic definition file. ## should be a sequence number, although the number itself is ignored (every name starting with `Cube` is valid).

The number of cubes must be equal to the number of lines that define the file names.

5.3 Datacube Configuration File Definition

A datacube configuration file contains the definitions for the three user defined regions-of-interest (ROI), and a total-ion count (TIC) image of the entire datacube. The total-ion count image is not used yet in version 0.9 of the Datacube Explorer, but will be in future version of the Datacube Explorer.

Both the ROI and TIC data is stored using Base64 encoding, in order to store binary data in the configuration text file. The values of roi1, roi2, and roi3 contain the data for ROI 1, 2 and 3, and the value of tic contains the data for the TIC image.

The content of a small configuration file looks like:

```
roi1 = BgAAAAcAAAABAAAAAAA=
roi2 = BgAAAAcAAAAAAAD//wM=
roi3 = BgAAAAcAAAAAAPwAAAA=
tic = BgAAAAcAAAD/AAAAAAAA/////wAA/////wAA
      /////wAA/////wAA/////wAAAAAAAAA=
```

For all data value hold the following data format:

```
<4 bytes UInt32 width><4 bytes UInt32 height><data>
```

where the low-order bytes are stored first (Microsoft .NET standard).

ROI data

For the ROI data, the ROI selected by the user is converted into a bit array, where true means the position is in the ROI, and false it is not. The array is created by adding all rows together for increasing height. The sequence of ROI bit data of a datacube with a width of 4 and a height of 3 is like

```
0   1   2   3

4   5   6   7

8   9  10  11
```

The bytes formed by this sequence of bit create the data part, where the length of the array is always an integer number of bytes (undefined bits are false). In this 4x3 example the number of bytes for the ROI data is 2. The total amount of bytes for the ROI data is 4 + 4 + 2 bytes here.

TIC data

The TIC image is created by adding spectral data values together per pixel, and scale all values between 0 and 255 for the minimum and maximum value of all summed pixels. These values are stored as an unsigned byte array in the same sequence where the ROI is stored in. The length of this byte array is equal to the number of pixels in the datacube.

The total amount of bytes for the TIC data in the example above is 4 + 4 + 12 bytes.

The byte arrays for roi1, roi2, roi3 and tic, and is then converted into a Base64 String individually.

When reading back the ROI and TIC values, it is advised to check the size of the data, both against the width and height in the ROI and TIC values as well as against the properties of the datacube itself.

6. REFERENCES

[1] Thunig, J., Hansen, S.H., and Janfelt, C., *Analysis of Secondary Plant Metabolites by Indirect Desorption Electrospray Ionization Imaging Mass Spectrometry.* Analytical Chemistry, 2011. **83**(9): p. 3256-3259.

[2] Gustafsson, J.O.R., Oehler, M.K., Ruszkiewicz, A., McColl, S.R., and Hoffmann, P., *MALDI Imaging Mass Spectrometry (MALDI-IMS)—Application of Spatial Proteomics for Ovarian Cancer Classification and Diagnosis.* International Journal of Molecular Sciences. **12**(1): p. 773-794.

[3] Kaspar, S., Peukert, M., Svatos, A., Matros, A., and Mock, H., *MALDI-imaging mass spectrometry - An emerging technique in plant biology.* Proteomics.

[4] MacAleese, L., Duursma, M.C., Klerk, L.A., Fisher, G., and Heeren, R.M.A., *Protein identification with Liquid Chromatography and Matrix Enhanced Secondaty Ion Mass Spectrometry (LC-ME-SIMS).* J. Proteomics, 2011. **74**(7): p. 993-1001.

[5] Aizikov, K., Smith, D.F., Chargin, D.A., Ivanov, S., Lin, T.Y., Heeren, R.M.A., and O'Connor, P.B., *Vacuum compatible sample positioning device for matrix assisted laser desorption/ionization Fourier transform ion cyclotron resonance mass spectrometry imaging.* Review of Scientific Instruments, 2011. **82**(5): p. 054102 1-8.

[6] Smith, D.F., Aizikov, K., Duursma, M.C., Giskes, F., Spaanderman, D.J., McDonnell, L.A., O'Connor, P.B., and Heeren, R.M.A., *An External Matrix-Assisted Laser Desorption Ionization Source for Flexible FT-ICR Mass Spectrometry Imaging with Internal Calibration on Adjacent Samples.* Journal of the American Society for Mass Spectrometry, 2011. **22**(1): p. 130-137.

[7] van Hove, E.R.A., Blackwell, T.R., Klinkert, I., Eijkel, G.B., Heeren, R.M.A., and Glunde, K., *Multimodal Mass Spectrometric Imaging of Small Molecules Reveals Distinct Spatio-Molecular Signatures in Differentially Metastatic Breast Tumor Models.* Cancer Research, 2010. **70**(22): p. 9012-9021.

[8] van Hove, E.R.A., Smith, D.F., and Heeren, R.M.A., *A concise review of mass spectrometry imaging.* Journal of Chromatography A, 2010. **1217**(25): p. 3946-3954.

[9] Chughtai, K. and Heeren, R.M.A., *Mass Spectrometric Imaging for Biomedical Tissue Analysis.* Chemical Reviews, 2010. **110**(5): p. 3237-3277.

[10] Heeren, R.M.A., Smith, D.F., Stauber, J., Kukrer-Kaletas, B., and MacAleese, L., *Imaging Mass Spectrometry: Hype or Hope?* Journal of the American Society for Mass Spectrometry, 2009. **20**(6): p. 1006-1014.

[11] Heeren, R.M.A., Taban, I.M., Altelaar, A.F.M., Van der Burgt, Y.E.M., McDonnell, L.A., Fuchser, J., and Baykut, G., *Imaging of peptides in the rat brain using MALDI-FTICR mass spectrometry.* Journal of the American Society for Mass Spectrometry, 2007. **18**(1): p. 145-151.

[12] Heeren, R.M.A., Altelaar, A.F.M., Luxembourg, S.L., McDonnell, L.A., and Piersma, S.R., *Imaging mass spectrometry at cellular length scales.* Nature Protocols, 2007. **2**(5): p. 1185-1196.

[13] Smith, D.F., Kharchenko, A., Konijnenburg, M., Klinkert, I., Pasa-Tolic, L., and Heeren, R.M.A., *Advanced Mass Calibration and Visualization for FT-ICR Mass Spectrometry Imaging.* Journal of the American Society for Mass Spectrometry, 2012. **23**(11): p. 1865-1872.

[14] Klinkert, I., Chughtai, K., Ellis, S.R., and Heeren, R.M.A., *Methods for full resolution data exploration and visualization for large 2D and 3D mass spectrometry imaging datasets.* International Journal of Mass Spectrometry, 2014. **362**: p. 40-47.

[15] Hamacher, M., Eisenacher, M., Stephan, C., Römpp, A., Schramm, T., Hester, A., Klinkert, I., Both, J.-P., Heeren, R.M.A., Stöckli, M., and Spengler, B., *imzML: Imaging Mass Spectrometry Markup Language: A Common Data Format for Mass Spectrometry Imaging*, in *Data Mining in Proteomics*, J.M. Walker, Editor. 2011, Humana Press. p. 205-224.

[16] Kohonen, T., *The self-organizing map* Proceedings of the IEEE 1990. **78**(9): p. 1464 - 1480

7. APPENDIX: SELF-ORGANIZING MAP

When a dataset has been loaded, a self-organizing map image classification functionality can be performed on the dataset. This analysis is classified as unsupervised competitive learning, and is also known as the Kohonen (neural) network [16]. After correctly parameterized analysis, an overview of typical image patterns becomes available. These typical image patterns are the typical ion distributions determined over all selected ion images which are fed into the classification algorithm. This feature is available from Menu -> Tools -> Self-organizing map.

For performing the analysis, the image dataset is converted into a set of images, defined by a start mass and an end mass, the mass bin per image and the step size to move to a next image. A threshold value is used to distinguish between images containing noise and images containing real image data.

Figure 7.1 shows the model of the Kohonen network used in the Datacube Explorer. The image classification is performed on a two dimensional set of output images, for example 4 by 4, which makes 16 images. The images, marked as the 'Input layer' in Figure 7.1, are fed into the network one by one. For every image fed into the network, a winning output image is determined on the basis of which of the output images have their associated weight vectors closest to the input vector of the input image, measured by a distance function. The distance is defined as the sum from all root mean squares of the differences in the pixel intensities of all pixel places of the input image and the output image. The winning output image is where the ‘distance’ of the fed image to the output images is minimal.

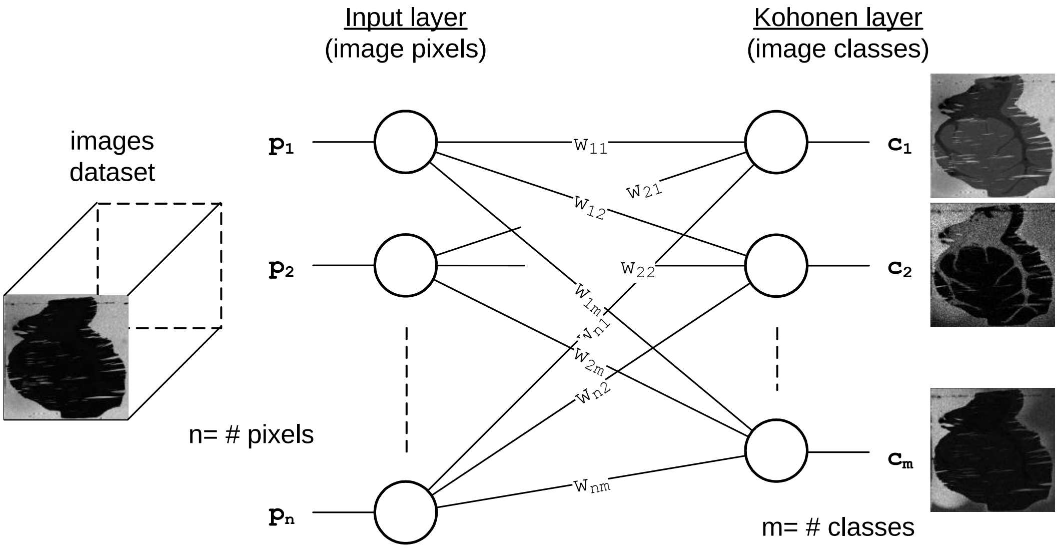


Figure 7.1: Model of Kohonen network used

The winning output image weight factors (and the images around it defined by the 2D neighbourhood value) are adapted to the input image with a learning rate factor. During the entire process both the neighbourhood value and the learning rate are reduced (to respectively 1 and 0).

The final images can be shown in an external image viewing program, for example Irfanview (http://www.irfanview.com/). In Figure 7.2 an example of the results (4 by 4) is shown.

When performing the self-organizing map analysis on a dataset, care should be taken to select the optimal parameters, which often is an iterative process.

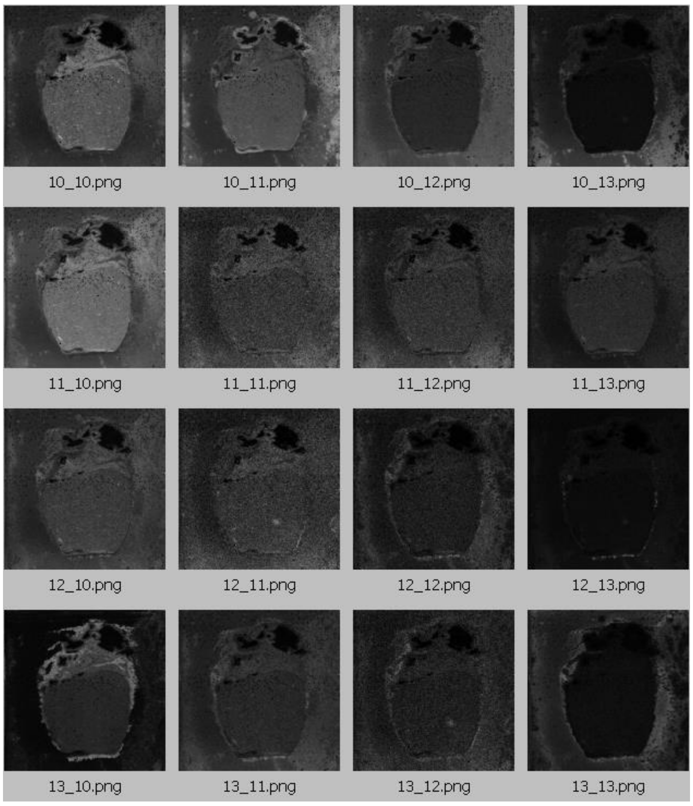


Figure 7.2: Example results of a Self-organizing map analysis

The Self-organizing map requires the following user entries.

Results output directory Directory to store the output images.

Start mass [Da] Start of images range.
End mass [Da] End of images range.
Mass bin size [Da] Mass range for spectral binning of every image.
Iteration step size [Da] Step size between successive images.
Threshold Minimum pixel threshold value in the image to take an image into account for the analysis.

Network output size (2D) The dimension of the output network.

Initial radius Radius of the network to start with.

Initial learning rate Learning rate to start with.

Number of iterations The number of iterations to perform. In one iteration all images are fed to the network once.

Manual scaling When disabled, the images are fed to the network with maximum scaling (minimum value black, maximum scaling white). When enabled the minimum and maximum values can be manually specified.

8. APPENDIX: KEYBOARD SHORTCUTS DATACUBE EXPLORER

General	
CTRL + O	Open new dataset
CTRL + P	Open new dataset as Volume-of-Interest
Datacube Explorer Window	
SHIFT + Arrow Left/Right	Up/down 0.001 Da.
CTRL + Arrow Left/Right	Up/down 0.005 Da.
Arrow Left/Right	Up/down 0.01 Da.
Arrow Up/Down	Up/down 0.1 Da.
Page Up/Down	Up/down 1 Da.
Home/End	Up/down 10 Da.
Mass Spectrum Window	
Click IN spectrum	Show m/z image, centre spectrum around this m/z.
CTRL + Click IN spectrum	Show m/z image, NO spectrum centring around m/z.
Click OUTSIDE spectrum,	
- in "Spectrum Up" area	Spectrum moves half the range up.
- in "Spectrum Down" area	Spectrum moves half the range down.
Area select with mouse	Zoom into selected spectrum area.
Backspace	Previous zoom level.
Right mouse click	Spectrum menu.