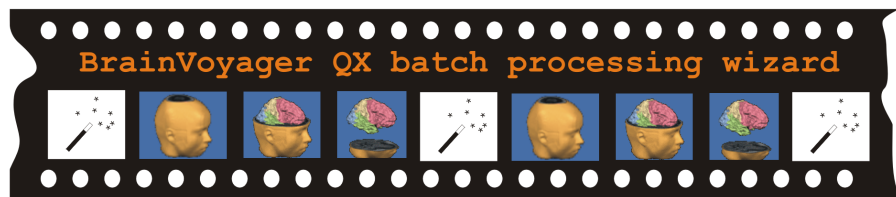


# manual for batch processing wizard for 4d data in brainvoyager qx



Hester Breman

October 13, 2010



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Notes per wizard version . . . . .	1
1.1.1	Planned features . . . . .	1
1.1.2	Notes on version 1.0 for BrainVoyager QX 2.2 . . . . .	1
1.1.3	Notes on version 0.8 for BrainVoyager QX 2.1 . . . . .	1
1.1.4	Notes on version 0.5 for BrainVoyager QX 2.1 . . . . .	1
1.1.5	Notes on version 0.6 . . . . .	1
1.1.6	Notes on version 0.5 . . . . .	1
1.1.7	Notes on version 0.4 . . . . .	2
1.1.8	Notes on version 0.3 . . . . .	3
1.1.9	Notes on version 0.1 . . . . .	3
1.2	Installation of the wizard . . . . .	4
1.2.1	Location of the plugin and script files . . . . .	4
1.3	Acknowledgements . . . . .	5
<b>2</b>	<b>Launching the wizard and global options</b>	<b>6</b>
2.1	Description of the functions . . . . .	6
2.2	Global options . . . . .	7
2.3	Selecting files for processing . . . . .	8
2.3.1	Introduction . . . . .	8
2.3.2	Selection of files via a file dialog . . . . .	8
2.3.3	Selection of files via a file list . . . . .	8
2.4	Processing order . . . . .	10
2.4.1	Order of processing of data sets . . . . .	10
2.4.2	Order of processing steps . . . . .	10
2.5	Reports . . . . .	10
2.5.1	BrainVoyager QX Log tab . . . . .	10
<b>3</b>	<b>Creating FMR or DMR projects</b>	<b>11</b>
3.1	Specifying parameters . . . . .	11
3.1.1	Creating FMR projects . . . . .	11
3.1.2	Creating DMR projects . . . . .	11
3.1.3	Storage order of the slices . . . . .	12
3.1.4	Set project parameters manually . . . . .	13
<b>4</b>	<b>Slice scan time correction</b>	<b>14</b>
4.1	Specifying parameters . . . . .	14
4.1.1	Slice order . . . . .	14
4.1.2	Interpolation method . . . . .	15
4.2	Background: slice scan time correction in BrainVoyager . . . . .	16
4.2.1	From User's Guide of BrainVoyager QX 1.9 . . . . .	16
4.2.2	From User's Guide for BrainVoyager 2000 . . . . .	20

<b>5</b>	<b>Performing motion correction and intra-session alignment</b>	<b>23</b>
5.1	Specifying parameters . . . . .	23
<b>6</b>	<b>Temporal filtering</b>	<b>25</b>
6.1	Specifying parameters . . . . .	25
6.2	Background: temporal filtering in BrainVoyager . . . . .	26
6.2.1	Introduction . . . . .	26
6.2.2	High pass filtering . . . . .	26
6.2.3	Gaussian temporal smoothing . . . . .	27
<b>7</b>	<b>Spatial filtering</b>	<b>28</b>
7.1	Specifying parameters . . . . .	28
7.2	Background: spatial smoothing in BrainVoyager . . . . .	29
7.2.1	Introduction . . . . .	29

## Abstract

**Outline** This document is created in order to provide an introduction to the procedures in the wizard for 4-dimensional data processing for BrainVoyager QX.

**Document version** : 0.6. For this manual it is assumed that the reader is already familiar with the functionality in BrainVoyager. If, however, the meaning of certain parameters may be unclear, please consult the documentation for BrainVoyager QX written by prof. dr. Rainer Goebel. For the reader's convenience, some of the information about preprocessing options in BrainVoyager has been included in this manual.  
See for example the online BrainVoyager QX User's Guide

<http://www.brainvoyager.com/bvqx/doc/UsersGuide/WebHelp/BrainVoyagerQXUsersGuide.htm>

and "Rainer's BV Blog":

<http://www.brainvoyager.com/resources/bvblog/bvblog.html>

*Hester Breman, Brain Innovation B.V., 2010*

# Chapter 1

## Introduction

This document provides information about the use of the wizard to process multiple 4D data sets in BrainVoyager QX.

### 1.1 Notes per wizard version

#### 1.1.1 Planned features

Creation of VTCs and VDWs.

#### 1.1.2 Notes on version 1.0 for BrainVoyager QX 2.2

This version includes intra-session alignment for multiple files.

#### 1.1.3 Notes on version 0.8 for BrainVoyager QX 2.1

This version also contains creation of FMR and DMR projects.

#### 1.1.4 Notes on version 0.5 for BrainVoyager QX 2.1

Because the scripting module changed due to changes in the Qt Library <sup>1</sup> that is used in BrainVoyager, the batch processing wizard had to be rewritten. All features in the 4d batch processing wizard will gradually re-appear in this rewritten version. We have tried to include as many as possible options for the current version.

#### 1.1.5 Notes on version 0.6

This version now includes the creation of DMR and VDW files. Also, it is possible to rename DICOM files in multiple directories. This version is for BrainVoyager QX 1.10.4 (because of the new scripting command to create VDW files).

#### 1.1.6 Notes on version 0.5

##### Version 0.5 beta 2

**Run all steps in one go** The main change in the transition from v0.4 to v0.5 is that it is now possible to run all processing steps sequentially. The wizard will still

---

<sup>1</sup>The Qt library for C++ can be found at: <http://www.qtsoftware.com/>

write the text files with names of the produced files after each processing stage. This version is for BrainVoyagerQX 1.9.10. The script project can only be used in BrainVoyager QX 1.9 or higher; in 1.9.10 there are more interpolation options than previous 1.9 versions. A wizard for an older version of BrainVoyager can be created on request. The function to create DMRs has been adopted and modified by Pim Pullens and is now separately available (on the wiki).

**Tabs for filenames** Whenever it is possible to select parameters for list of files, the dialogs are using tabs now, so that the size of the dialogs stays more decent. Please go through all of the tabs before proceeding to the next dialog.

**Default directory** When running the wizard, one can indicate a default directory. This saves a lot of clicking when selecting files. Also, the resulting log files can all be found now in this default directory.

**Intra-session alignment** In the previous version of the wizard, it was assumed that the same parameters would be applied for each FMR project. In the current version one can select a target FMR project if needed and indicate the target volume for each FMR project.

### **Version 0.5 beta 3**

**Set inter-slice time manually** This option has been added in the beta 3 version (29.04.08). See section 3.1.4.

**Single image file loops** This option has been added in the beta 3 version (29.04.08). See section 3.1.3.

**Improvements and bug fixes** The intra-session alignment procedure uses now the number of the target file in the list instead of the name of the target file, since the name is due to changes throughout the preprocessing procedure. The order of processing has been made similar to the order of initial manual file selection.

The “linear trend removal only” option was applied after the full high-pass filtering option. This has been changed to performing linear trend removal only when this option is checked.

## **1.1.7 Notes on version 0.4**

The functions can be started via the BrainVoyager QX “Scripts” menu. The functions can be used directly, without modification of the scripts in the script project. The main functions will give a text file with FMR project names as output, that can be used for further preprocessing steps. The projects will be written to the same directory as the source files. All main functions have the same structure:

1. select filenames or list with filenames
2. specify parameters or read parameters file (only motion correction and vtc creation)
3. perform the processing
4. write a list with processed filenames to disk

The help functions are provided to write a filenames or parameters file, to visually inspect the processing result and to convert between filter units. For an explanation of the processing steps, please consult the BrainVoyager QX Getting Started Guide or the Users Guide. These documents can be found via <http://www.brainvoyager.com/>. For information on scripting in BrainVoyager QX, please consult the Automation and Scripting guide that is available on the wiki at <http://wiki.brainvoyager.net/>. This version is for BrainVoyagerQX 1.9. The script project is not backward compatible. A new function to create DMR files is included.

Thanks to Koene van Dijk and Michael Capalbo for suggestions.

### **1.1.8 Notes on version 0.3**

Because of QSA differences, there are now two versions of the batch processing project, one for 1.8 and one for 1.9. In this version, it is now possible to provide filenames for the FMR project. Also, it is not necessary any more to indicate the number of projects in advance.

In this documentation, background information on the image processing options has been added (from the topic "FMR Preprocessing" → "Update" topic in the BrainVoyager 2000 User's Guide at:

<http://www.brainvoyager.de/BV2000OnlineHelp/BrainVoyagerOnlineHelp.html>).

For an up-to-date explanation of the preprocessing techniques, see also Goebel et al (2006, [1]). Date: May 2007.

### **1.1.9 Notes on version 0.1**

Date: December 2006.

## 1.2 Installation of the wizard

### 1.2.1 Location of the plugin and script files

The shared library (plugin) `batchprocessingwizard_vxx.dylib/so/dll`, the script `batchprocessingwizard_vxx.js` and the interface `batchprocessingwizard_vxx.ui` should be located in the folder:

`/ (My) Documents/BVQXExtensions/Plugins_32` (on Mac: also `_64`)/  
(see figure 1.1).

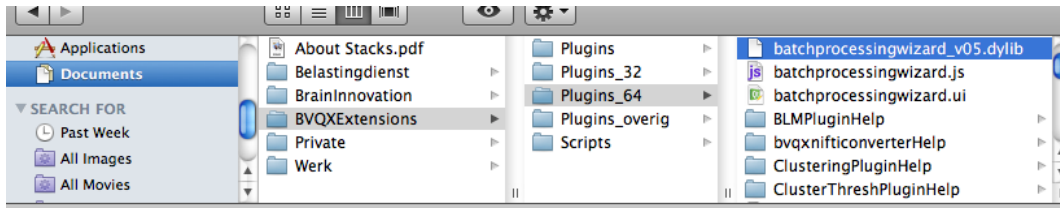


Figure 1.1: Location for the wizard

When the plugin and script files are in the proper location, BrainVoyager QX can be started. The `batchprocessingplugin` will appear in the BrainVoyager QX "Plugins" menu.

### **1.3 Acknowledgements**

Thanks to Dany d'Souza, Christoph Oberthuer, Hedy Kober, Johnston, Jochen Seitz and Bettina Sorger for their suggestions for earlier versions of the wizard. Thanks to Henk Jansma for his Human Brain Mapping poster which inspired me for the logo of the batch processing wizard.

## Chapter 2

# Launching the wizard and global options

### 2.1 Description of the functions

Wizard v0.6, called `batchprocessingwizard.dylib/dll/so`, covers most FMR processing steps that can be scripted in BrainVoyager QX, from project creation to spatial preprocessing; also, diffusion weighted image projects (DMR projects) can be created with this wizard.

The functions can be used in one procedure, where the output filenames serve as input for the next step, or separately, when it is for example necessary to perform temporal high-pass filtering on a bunch of functional data files. The processing that can be performed is:

**creating FMR projects:** Input: files selected via dialog.

Output: project files `<original name>.fmr`.

**creating DMR projects:** Input: files selected via dialog.

Output: project files `<original name>.dmr`.

**slice scan time correction:** Input: filenames from previous step or files selected via dialog.

Output: `Project1_SC*.fmr` to `ProjectN_SC*.fmr`.

**motion correction:** Input: filenames from previous step or files selected via dialog.

Output: `*_3DMC.fmr` files.

**temporal filtering:** Input: filenames from previous step or files selected via dialog.

Output: `*_T*.fmr` files.

**spatial filtering:** Input: filenames from previous step or files selected via dialog.

Output: `*_SGS*.fmr` files.

The FMR and DMR files are written to the same directory as where the source files reside. If several steps are performed sequentially, the files only need to be selected once.

The text files with lists of produced files are written to the main directory that can be indicated after the project parameters have been entered (see figure ??). In case a text file with the same name already exists, this might be overwritten. So in case there are important file list files, it might be helpful to transfer these to another directory.

## 2.2 Global options

The wizard can be started via the function 'batchprocessingplugin' via the BrainVoyager 'Plugins' menu (see figure 2.1).

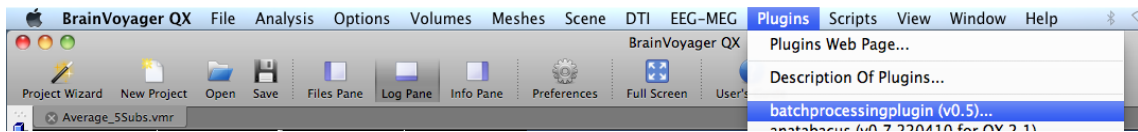


Figure 2.1: Launch the wizard via the BrainVoyager 'Plugins' menu

A main dialog will appear indicating all possible processing steps that can be performed at a time (see figure 2.2). The wizard will only accept parameters of options that have been selected on this global options tab; all other values will be ignored.

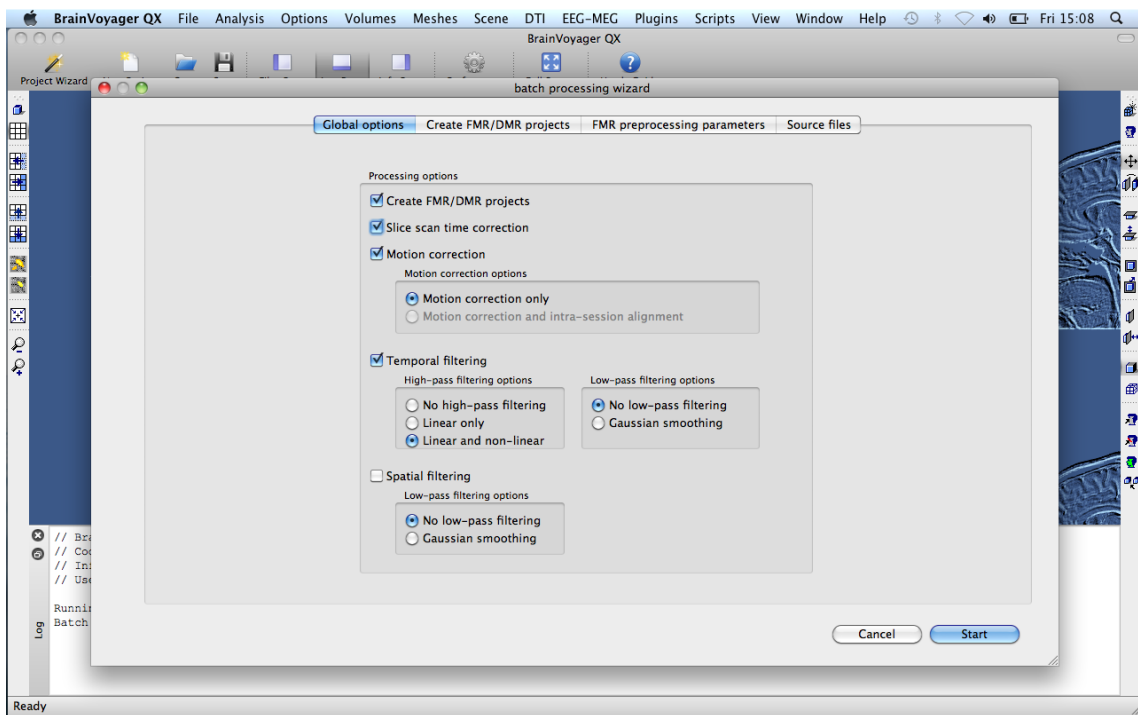


Figure 2.2: Global options of the wizard

## 2.3 Selecting files for processing

### 2.3.1 Introduction

In this wizard, the files to be processed have only to be selected once. To select these files, it is possible to select the files one by one via a file dialog as shown in figure 2.3.

After each step that has been performed, a text file with the resulting names of the processed files is written to the main, default directory that has been indicated earlier in the project.

In case you would only perform part of the processing, this text file with names can be used to continue the process, instead of manually selecting all files again.

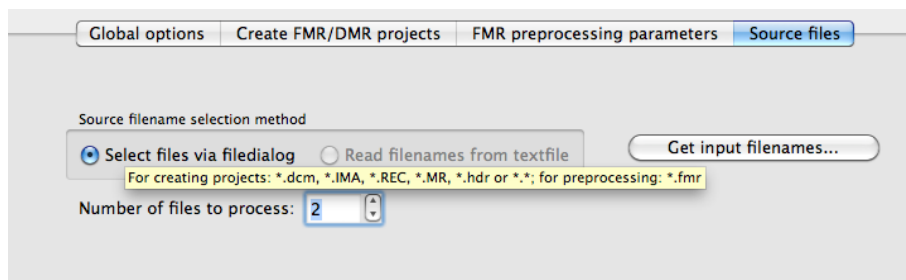


Figure 2.3: Choice for file selection method

### 2.3.2 Selection of files via a file dialog

When all parameters have been entered via the main dialog, the files for the processing can be selected via the “Get input filenames” button. A file dialog will appear for the number of files indicated on the tab.

In case the option “Motion correction and intra-session alignment” has been chosen, after each file the target FMR can be selected via the file dialog. Even if the source files are in other format, for example DICOM, the target filenames should be from FMR projects, which means one should first process the runs that only need to be motion corrected and will serve as target, then the runs that need to be aligned to other runs.

One needs to indicate the number of files that have to be processed; in the example we have two runs, therefore the number 2 has been entered on the Source filenames tab (see figure 2.3). In case intra-session alignment has been selected, the total number of files to be selected will be doubled (sourcefiles and targetfiles).

### 2.3.3 Selection of files via a file list

When all parameters have been entered via the main dialog, the textfile containing the filenames for the processing can be selected via the “Get input filenames” button. A file dialog will appear once. The textfile should have the following format:

```
<nr of files>  
<filename 1>  
<filename 2>  
...  
<filename n>
```

In case the option “Motion correction and intra-session alignment” has been chosen, the textfile needs to contain the same number of target files after the number of source files in the following format:

```
<nr of files>
<source filename 1>
<source filename 2>
...
<source filename n>
TARGETFMRs
<target filename 1>
<target filename 2>
...
<target filename n>
```

The filenames should contain the full file path. An example file is shown in figure 2.4.

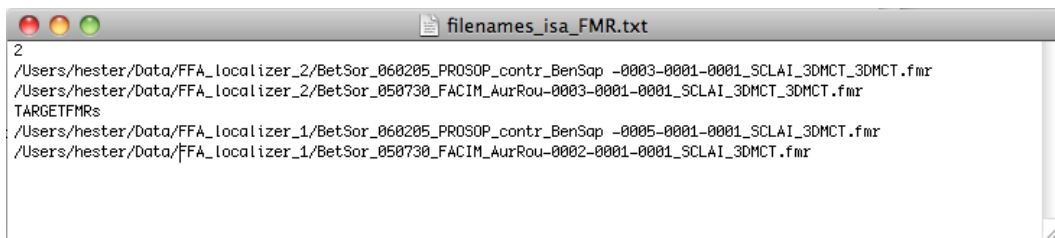


Figure 2.4: Example file produced after use of motion correction and intra-session alignment function

The file selection order determines the order of processing of the data sets (see section 2.4).

## **2.4 Processing order**

### **2.4.1 Order of processing of data sets**

The order of file selection determines the order of processing. So in case you think one data set should be processed before another one, follow the same temporal order as the wizard should follow later on.

### **2.4.2 Order of processing steps**

When multiple data sets are selected, each step will be processed for all data sets before proceeding to the next step. So first all FMR projects will be created before any slice scan time correction will occur. For the FMR preprocessing, the wizard assumes that slice scan time correction precedes any other preprocessing step. The processing steps are applied in the following order:

1. Project creation
2. Slice scan time correction
3. Motion correction
4. Temporal filtering
5. Spatial filtering

When you prefer another order, apply the steps individually.

## **2.5 Reports**

### **2.5.1 BrainVoyager QX Log tab**

The selected files and processing options are printed to the BrainVoyager QX Log tab. After selecting the text, a right-click provides the possibility to copy the text. This text can be pasted and saved in a simple text file.

## Chapter 3

# Creating FMR or DMR projects

### 3.1 Specifying parameters

#### 3.1.1 Creating FMR projects

The options to create FMR projects are similar to the ones when creating a single FMR project via 'File' → 'New' in BrainVoyager, although the difference is that the headers of the files are not read automatically. This means that the acquisition parameters should be entered carefully; the parameters that are displayed in figure 3.1 are simply default parameters.

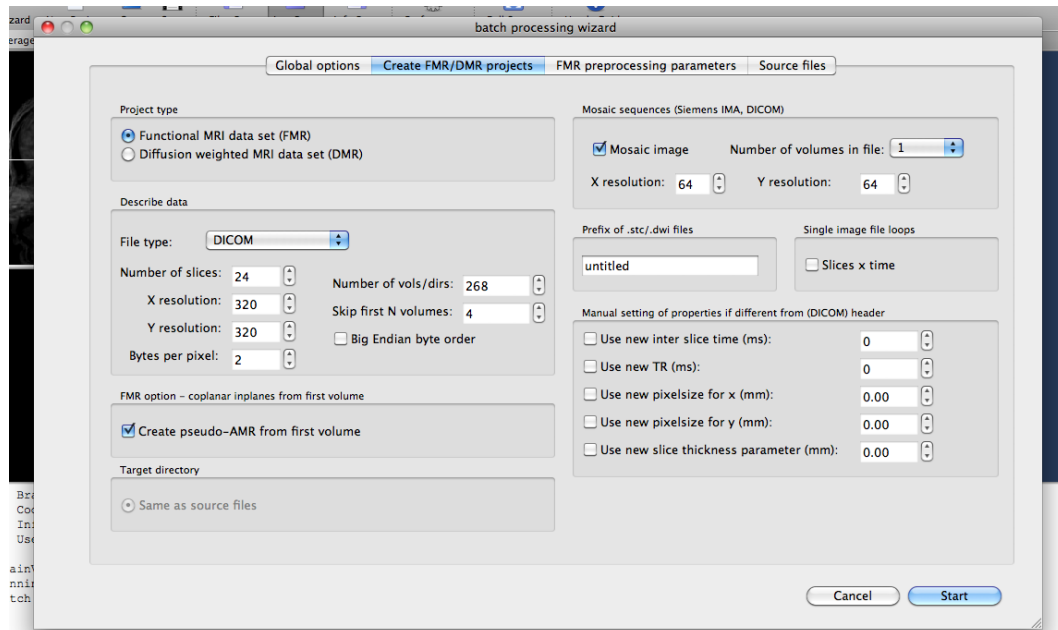


Figure 3.1: Options for creating FMR or DMR projects

#### 3.1.2 Creating DMR projects

To create multiple DMR projects, can use here the same parameters as when creating DMR files via 'File' → 'Create Project Wizard'; just like with creating DMRs via

this batch processing wizard, headers of the files are not read automatically, therefore please take care to enter the acquisition parameters that are applicable to your files. In creating DMR projects, one can use the following three filetypes: DICOM (\*.dcm or \*.IMA), Philips (\*.PAR and \*.REC) or Analyze (\*.hdr and \*.img). The settings for creating the BrainVoyager example DTI dataset are displayed in figure 3.2.

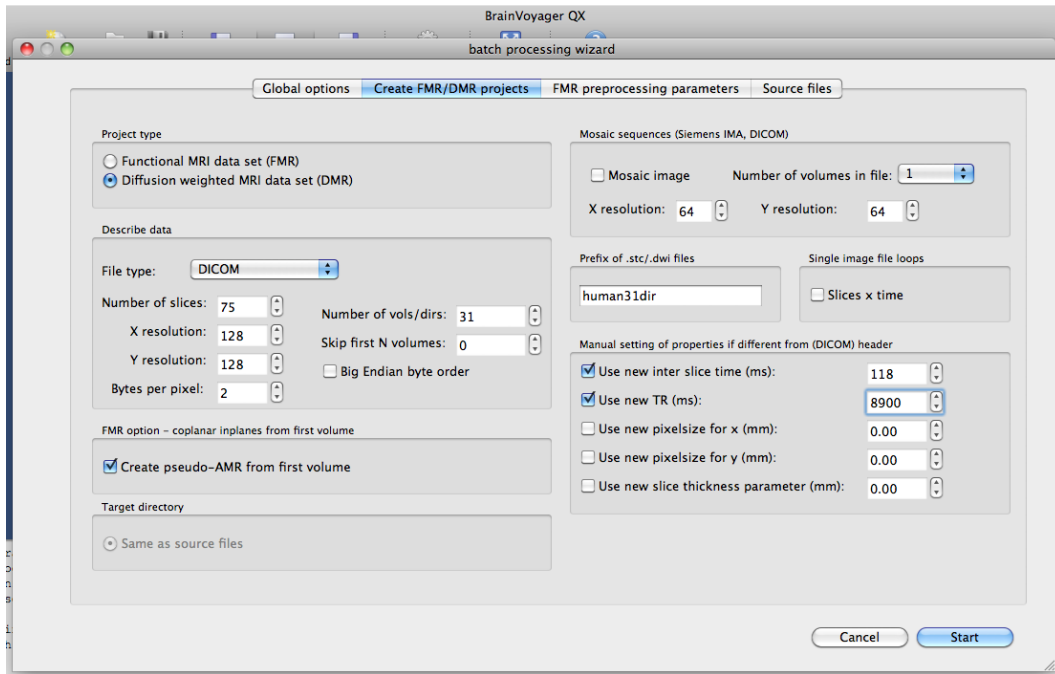


Figure 3.2: Create example DMR project

For further processing of DMR files, please see the DTI Getting Started Guide by Pim Pullens. This guide is installed with BrainVoyager (can be found in the folder `/BrainVoyager QX/GettingStartedGuides/` or can be downloaded from [http://wiki.brainvoyager.com/DTI\\_in\\_BVQX](http://wiki.brainvoyager.com/DTI_in_BVQX)).

### 3.1.3 Storage order of the slices

Usually all slices of a volume acquired at a timepoint are stored before a new volume is stored: slice 1, 2..n at timepoint 1, slice 1, 2..n at timepoint 2, to slice 1, 2..n at timepoint n. In some cases, all timepoints of a slice are stored together before a new slice is stored; so in this case the slices are not stored volume-wise, but slice-time wise.

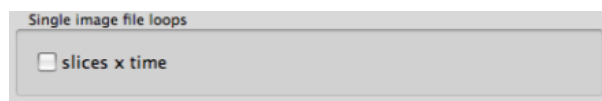


Figure 3.3: Set slice-wise storage order of slices (in contrast to volume-wise storage order)

If your functional MRI data are stored in the slice-time fashion, use the “slices × time” option to ensure that the data are read properly (see figure 3.3).

### 3.1.4 Set project parameters manually

If the interslice time is not set explicitly as a property, it will be computed by BrainVoyager. The computed interslice time is  $TR/\text{number of slices}$ . In the wizard it is possible to enter an interslice time that is different from the computed interslice time. This can be necessary in cases when the TR is longer than the time of actual data acquisition, for example when the MRI experiment is combined with another modality in the scanner.

In the wizard, it is possible to alter the following parameters:

- TR
- inter slice time
- pixel resolution on x-axis
- pixel resolution on y-axis
- slice thickness

If the actual interslice time is different from the computed interslice time, check the “Use new inter slice time” box and enter the actual interslice time (see figure 3.4). Please take care that the new interslice time is preserved throughout the processing (on some dialogs, BrainVoyager displays the computed interslice time instead of the interslice time that is saved in the FMR project).

If the interslice time of your data is different from the computed interslice time, use this option otherwise the slice scan time correction might need be performed correctly. For the example diffusion weighted data set, we have to reset the slice time and TR.

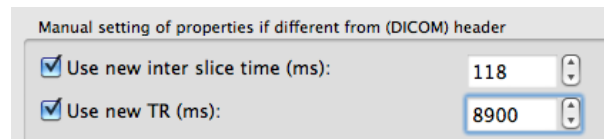


Figure 3.4: Set inter-slice time and TR manually

The DMR project will be saved with the same name as the raw data file (DICOM in this case, but with new \*.dmr/\*.dwi extension) and with the new parameter values.

## Chapter 4

# Slice scan time correction

### 4.1 Specifying parameters

For slice scan time correction, only two parameters need to be indicated (see figure 4.1). These are the slice order and the interpolation method.

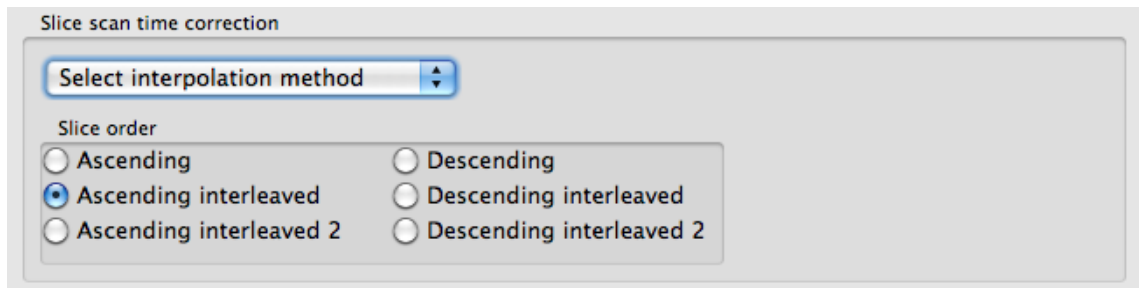


Figure 4.1: Options for slice scan time correction

#### 4.1.1 Slice order

The file names contain the characteristic indicating by SC that the FMR project was slice scan time corrected, and an additional marker to indicate the slice order. A is used for ascending, D for descending, I for interleaved and 2 is added if there are interleaved in a special way (Siemens-specific).

- SCLA: ascending
- SCLAI: ascending interleaved
- SCLAI2: ascending interleaved 2
- SCLD: descending
- SCLDI: descending interleaved
- SCLDI2: descending interleaved 2

Concerning the interpolation method, \*\_SCL-- .stc is an abbreviation of linear interpolation, \*\_SCC-- .stc of cubic interpolation and \*\_SCS-- .stc of SINC

interpolation. Information about the meaning of slice order in BrainVoyager from the User's Guide can be found in section 4.2.2.

#### **4.1.2 Interpolation method**

One can currently choose from 3 interpolation methods for slice scan time correction, which are linear interpolation, cubic spline interpolation and (windowed) SINC.

For a comparison of different interpolation methods, see the new section in the BrainVoyager User's Guide written by prof. Goebel that has been included below (see section 4.2.1). For more information about interpolation methods, see for example Meijering et al [2] or Thevenaz et al [3].

## 4.2 Background: slice scan time correction in BrainVoyager

### 4.2.1 From User's Guide of BrainVoyager QX 1.9

An important step of preprocessing, especially for event-related designs, is the proper treatment of differences in the time when individual slices are recorded. The problem of different slice scanning times stems from the fact that a functional volume (e.g. whole brain) is usually not covered at once but with a series of successively measured 2D slices. For a functional volume of 30 slices and a volume TR of 3 seconds, for example, the data of the last slice is measured almost 3 seconds later than the data of the first slice. Despite the sluggishness of the hemodynamic response, an imprecise specification of time in the order of 3 seconds will lead to suboptimal statistical analysis, especially in event-related designs.

One way to cope with slice scanning time differences would be to shift the expected BOLD time course in time to compute proper statistical results. In this *predictor-shifting* approach, the reference time courses for a slice are shifted in time proportionally to the temporal difference in scan time with respect to the reference (e.g. first) slice. Another possibility is to shift the data of a slice in time to the same time point as when the reference slice was scanned. This changes the data in a way as if the whole volume would have been measured at the same moment in time. Note that in the latter case, the same predictors can be used throughout the volume, i.e. slice-specific shifts of the predictors are no longer necessary. For this reason, the latter approach is normally used in fMRI data analysis, allowing to use the same predictors also after transforming the slice-based representation of the functional data (FMR/STC) to a 3D representation of the data (VTC) in an arbitrary (e.g. AC-PC or Talairach) space. It also allows to compare and integrate event-related responses from different brain regions correctly with respect to temporal parameters such as onset latency. The predictor-shifting approach would require that each voxel would keep a label indicating the slice from which it originates so that the correct shifted set of predictors could be used. Note that the *data-shifting* approach requires interpolation (resampling) of the slice time courses (see figure 4.2).

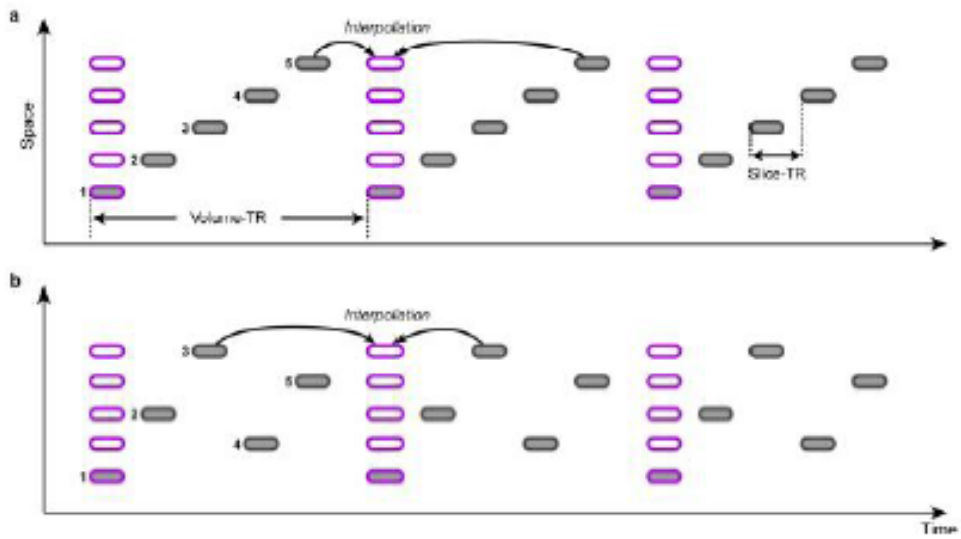


Figure 4.2: The data-shifting approach requires interpolation (resampling) of the slice time courses

### Slice Scanning Order

In order to correct for different slice scan timings, the time series of individual slices are temporally “shifted” to match a reference time point, e.g. the first or middle slice of a functional volume. This temporal shift depends on the order in which the individual slices of a volume are scanned. Besides an ascending or descending order, slices are often scanned interleaved, i.e. the odd slice numbers are recorded first followed by the even slice numbers. The figure above shows an example of ascending scanning (top) and interleaved scanning order (bottom) for five slices. The time points of original scan times are indicated with gray rounded rectangles, while the shifted data are depicted with pink rectangles. The resampled data can be treated as if all slices of one functional volume would have been scanned at the same time point.

### Temporal Interpolation

As described above, correcting differences in slice timing using the data-shifting approach requires to re-sample the data at time points falling in-between measured data points. The values at the non-measured time points can be estimated by using measured data points “in the neighborhood” i.e. from time points measured in close proximity. In BrainVoyager QX, three interpolation methods are available, linear, cubic spline and windowed sinc interpolation. The linear interpolation method uses only the neighbor on the left and on the right side to calculate the value at the resampled point. It implements an average of these values weighted by the relative distance:  $xt_{new} = (1 - \delta) * xt_{-1} + \delta * xt$ . The linear interpolation method is fast to compute but has the disadvantage that it smooths the data. Furthermore, the introduced smoothing depends on the slice, i.e. it is negligible if points are resampled close to a measured point but smoothing is rather strong if points are resampled in the middle between two measured points (see below). The two other interpolation schemes, cubic spline and sinc interpolation, avoid this problem by using more points in the neighborhood leading to a very accurate resampling.

In the following, the different interpolation methods are compared using the first 50 volumes of the “Objects” sample data set (the first two volumes have been skipped, i.e. volumes 3 - 52 are used from the original data). The sample data consists of 25 slices, which are scanned within 2 seconds (TR) in an interleaved ascending order.

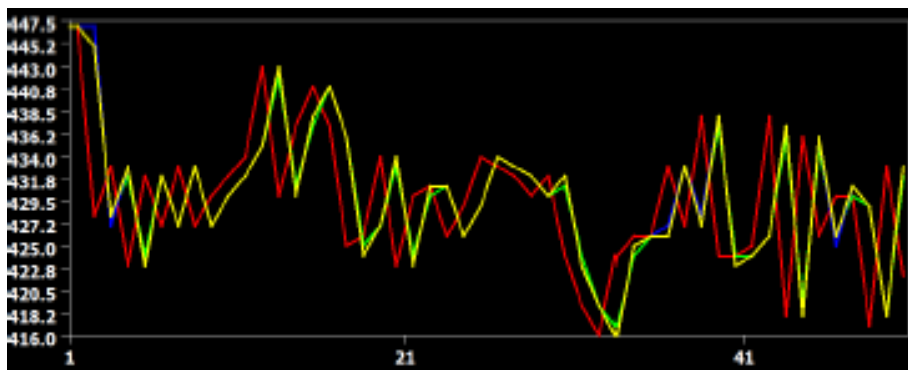


Figure 4.3: The original time course of a voxel in red from slice 22

Figure 4.3 shows the original time course of a voxel in red from slice 22, which

is the slice before the last slice scanned within a volume. In this case, the data has to be shifted forward almost for one full TR and the figure shows that all interpolation methods are performing this shift very well without significant deviations from each other. The curve from the linear interpolation method is shown in green, the cubic spline method in blue and the sinc method in yellow. Since the sinc curve is plotted last, the linear and cubic spline curve are hidden behind due to the similarity in the interpolation result.

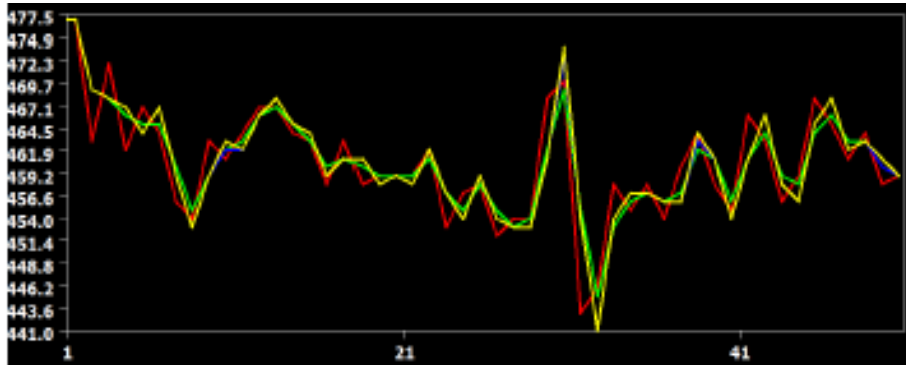


Figure 4.4: The raw data (red) and the result from the three interpolation methods are shown for slice 23

In figure 4.4 the raw data (red) and the result from the three interpolation methods are shown for slice 23, which is the slice scanned prior to the middle slice. In this situation, the time course must be shifted about half a TR, i.e. time points must be “recreated” in the middle between two measured time points. In this situation, the cubic spline (blue) and sinc (yellow) interpolation method are performing very similar but the linear interpolation method introduces visible smoothing. This is shown in more detail in the two subsequent figures.

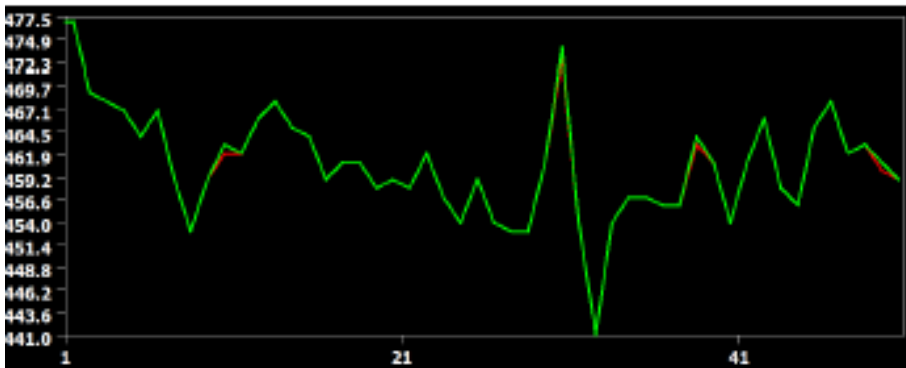


Figure 4.5: Comparison of the cubic spline (red) and sinc (green) interpolation method: they produce identical results (the green curve hides the red curve)

Figure 4.5 compares the cubic spline (red) and sinc (green) interpolation method, showing that they produce identical results (the green curve hides the red curve). Figure 4.6 below compares the interpolation result from the cubic spline (red) and the linear (green) interpolation method. Here it becomes clear that the linear inter-

polation method smooths the data as is apparent in weaker up and down deflections in the green as compared to the red curve.

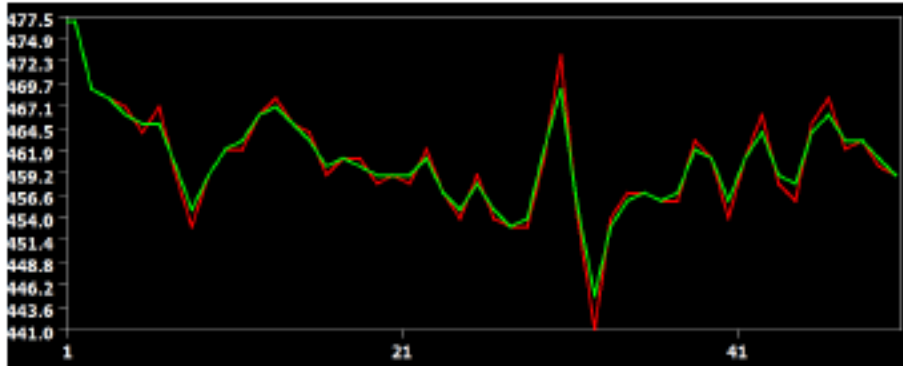


Figure 4.6: Comparison of the cubic spline (red) and linear (green) interpolation method: the linear interpolation method smooths the data

The examples confirm the theoretically expected behavior of the interpolation methods demonstrating slice-dependent smoothing of the linear method. Therefore cubic spline or windowed sinc interpolation should be used for slice scan time correction. Cubic spline interpolation is set as the default method in the Preprocessing dialog (see snapshot of the right upper part of the Preprocessing dialog in figure 4.7 below).

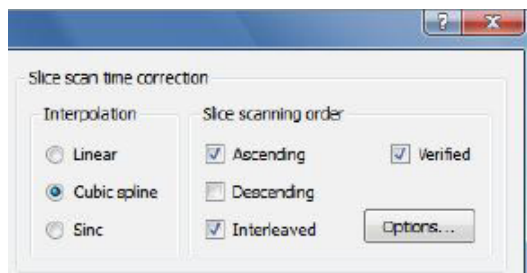


Figure 4.7: Slice scan time correction: in right upper part of the Preprocessing dialog

#### 4.2.2 From User's Guide for BrainVoyager 2000

The slices comprising one functional volume are scanned sequentially, i.e. at different moments in time. For functional analysis, i.e. in event-related designs, a whole functional volume is, however, treated as one data point, as if all slices were measured at the same time. To make this treatment of the data valid (i.e. for interpreting time the same way across a functional volume), the sequentially scanned slices have to be interpolated in time. For a correct temporal interpolation of the slices, the slice scanning order has to be known, which can be ascending or descending and either interleaved or not. The scanning order of the slices together with the TR and inter slice time can then be used to perform an appropriate temporal interpolation. You can either use sinc (default) or linear interpolation by selecting the Linear or Sinc option, respectively, in the Interpolation field within

the Slice scan time correction field (see snapshot below).

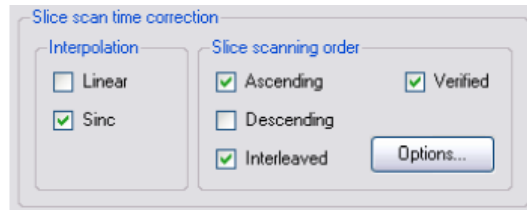


Figure 4.8: Slice scan time correction: in right upper part of the Preprocessing dialog (BV2000)

The specification of the slice order has to be performed with care because different manufacturers define "ascending" differently, i.e. either from the top of the head to the bottom or vice versa. In addition, BrainVoyager allows to reorder the slices (Reverse order button in the Edit FMR File Specifications dialog). To avoid any misconception, BrainVoyager now provides a Slice Scanning Order dialog to aid in the correct slice order specification. To invoke this dialog, click the Options... button in the Slice scan time correction field. The slice order dialog separates two pieces of information: the display order and the scan order of the slices. The display order is shown as a sequence of thumbnail images and is identical to the order shown in the FMR project window (see snapshot below).

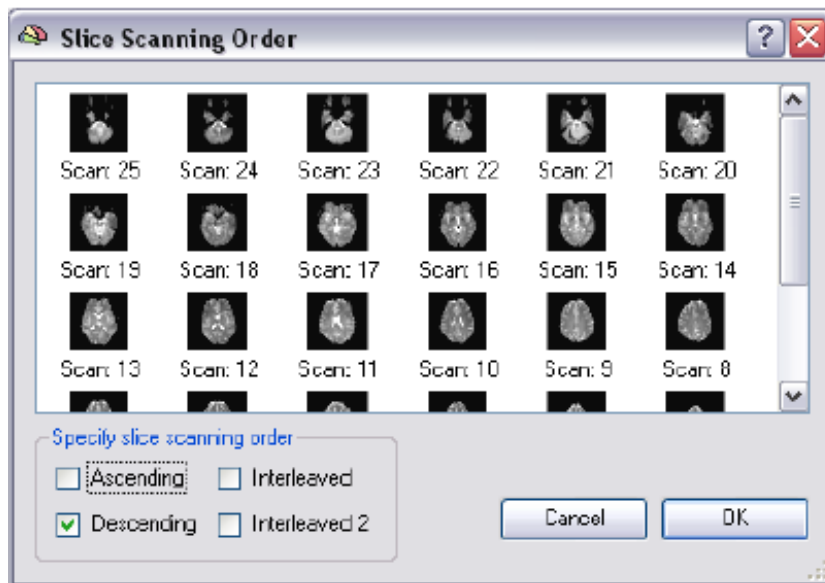


Figure 4.9: Slice order: descending

The correct scan order of the slices has to be specified below the thumbnails with respect to the display order. In the example data set shown above, the images are displayed from the top of the brain (left upper slice shown) to the bottom of the brain (right lower slice shown). If the slices would have been scanned also from the top to the bottom, the correct selection would be the Ascending option in the Specify slice scanning order field. In this example, the slices were, however,

scanned from the bottom to the top so that the correct selection is the Descending option (see snapshot above). The labels below the thumbnails ("Scan: 1", "Scan: 2" etc.) reflect now the real scan order, i.e. the bottom slice was scanned first. Note that this specification is correct irrespective of whether the bottom-to-top scan order was labelled "ascending" or "descending" by the manufacturer. What counts is the specification of the scan order with respect to the display order. Scanning is often performed in "interleaved" mode in which first the odd (slice 1, 3, 5...) and even (slice 2, 4, 6...) numbered slices are scanned. The Interleaved option can be combined with the Ascending and Descending option. As an example, "ascending, interleaved" has been specified in the snapshot below. With respect to the display order, the "Scan: <x>" labels below the thumbnails show accordingly that the first slice displayed was scanned first, the third slice displayed was scanned second, the second slice displayed was acquired as scan 11 and so on.

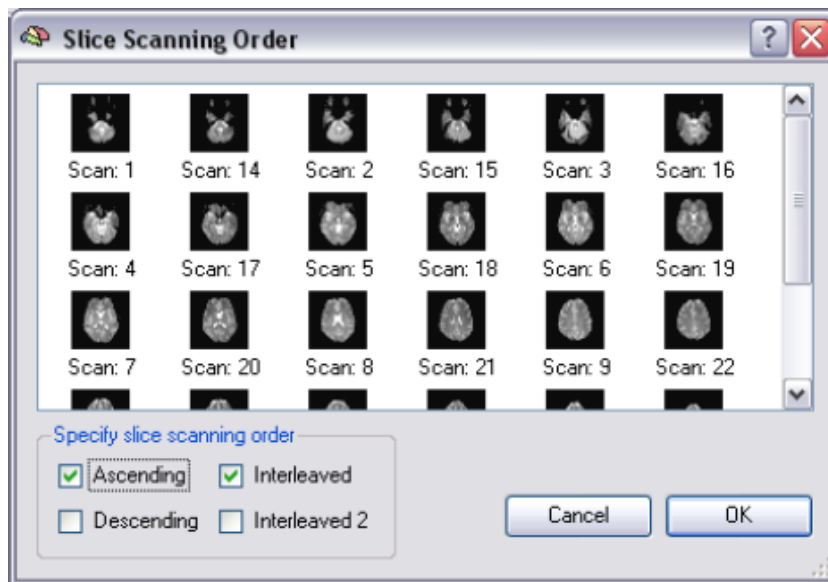


Figure 4.10: Slice order: ascending

An option Interleaved 2 is additionally available for Siemens scanners with the Numaris 4 software. In this latest software, "interleaved" is defined differently with regard to an odd or even number of slices. BrainVoyager takes care about this particular definition when using the "interleaved 2" option.

The file name segment reflecting slice scan time correction is `_SC [SL][A—D][I—I2]—`. The `_SC` part is always present to denote "slice scan time correction". This is followed by either letter "S" denoting "sinc interpolation" or letter "L" denoting "linear interpolation". This is then followed by either letter "A" denoting "ascending" or letter "D" denoting "descending". Optionally this is followed by the letter "I" or the letters "I2" denoting "interleaved" scanning. As an example, the string `_SCSAI` in an FMR file name tells us that slice scan time correction had been performed using sinc interpolation and with an ascending interleaved scan order specification.

## Chapter 5

# Performing motion correction and intra-session alignment

### 5.1 Specifying parameters

For motion correction, the global parameters can be provided via the ‘3D Motion correction and intra-session alignment’ section on the preprocessing tab of the wizard (see figure 5.1). These are the same parameters as can be entered via the “Pre-processing” dialog of the BrainVoyager menu.

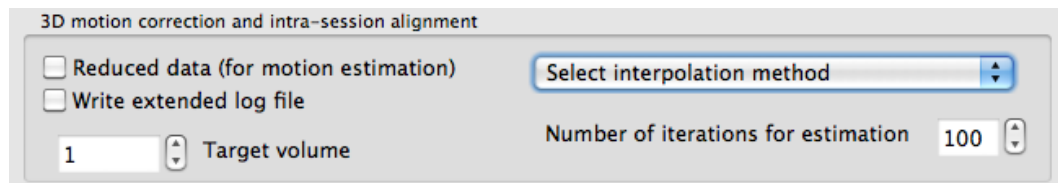


Figure 5.1: Options for motion correction

Interpolation options are 1. trilinear motion estimation and correction 2. trilinear motion estimation and SINC motion correction 3. SINC motion estimation and correction.

The difference between the motion correction log (\*\_3DMC.log) and extended log (\*\_3DMC\_verbose.log) is that for the default log, the rotation and translation parameters per volume are given in millimeters and degrees:

```
-> volume:    2   n_its:    3
dx: 0.0000 mm  dy: 0.0000 mm  dz: 0.0000 mm
rx: 0.0000 degs  ry: 0.0000 degs  rz: 0.0000 degs
```

while in the extended log file the gradients for translations and rotations, and the error and  $\lambda$  are provided per iteration for each volume:

```
volume: 2 iteration: 1   Error: 54.015769   Lambda: 1.000000e-04
GradientX: -0.393465   TranslX: 0.001868
GradientY: 0.835942   TranslY: 0.005183
GradientZ: 30.727486   TranslZ: -0.000186
GradRotX: 1.723598   RotX: -0.007159
GradRotY: 0.364815   RotY: 0.001076
GradRotZ: -0.006980   RotZ: -0.001683
```

The target volume refers to the volume in the run that the other volumes are aligned to. In motion correction only this refers to a volume in the same run; in

motion correction and intra-session alignment the target volume refers to the volume in the target run.

The generation of movies is set to true, so motion movies will always be generated.

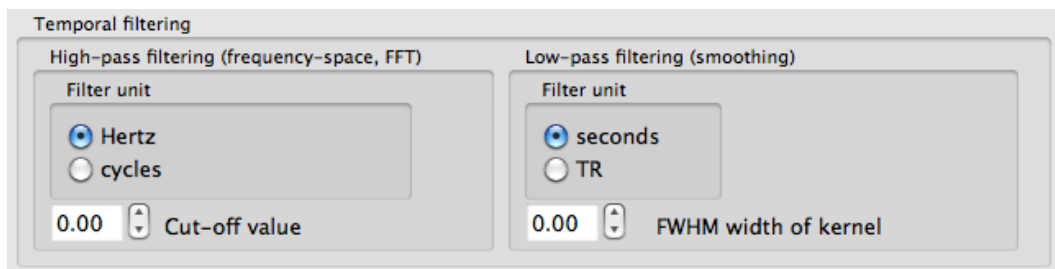
The intra-session alignment option is available from wizard version 1.0. To apply intra-session alignment, processed FMRs must be available to serve as target FMRs. The filenames of the target FMRs can be selected after selecting each source file via the file dialog, or placed in a list after the list of source filenames in a text file (see section 2.3.2).

## Chapter 6

# Temporal filtering

### 6.1 Specifying parameters

High-pass and low-pass temporal filter parameters can be entered via the ‘Temporal filtering’ section on the ‘Preprocessing’ tab of the wizard (see figure 6.1). The filter value can be provided in spectral units, either cycles per second (Hertz: ‘Hz’) or cycles per time course (‘cycles’).



The image shows a software interface titled "Temporal filtering" with two main sections:

- High-pass filtering (frequency-space, FFT):**
  - Filter unit:  Hertz,  cycles
  - 0.00 [up/down arrows] Cut-off value
- Low-pass filtering (smoothing):**
  - Filter unit:  seconds,  TR
  - 0.00 [up/down arrows] FWHM width of kernel

Figure 6.1: Parameters for temporal filtering

## 6.2 Background: temporal filtering in BrainVoyager

The information below is borrowed from the BrainVoyager 2000 User's Guide, just to provide general information about the filter units.

### 6.2.1 Introduction

Temporal filtering. Temporal filtering consists of two main functions: *removal of drifts* and *temporal gaussian smoothing*.

### 6.2.2 High pass filtering

The removal of drifts is very important and is highly recommended whereas temporal smoothing is less important and not checked as default.

The drifts observed in voxel time series are often linear but can be also nonlinear (see time course below).

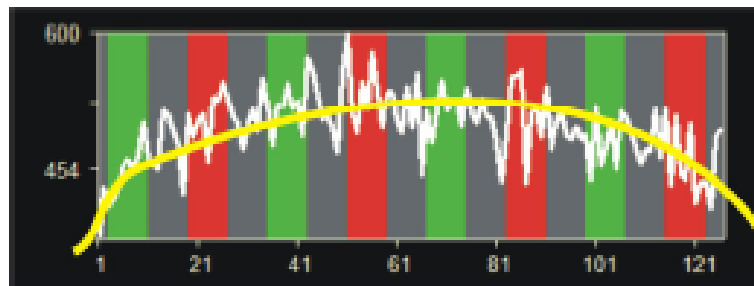


Figure 6.2: The drifts observed in voxel time series are often linear but can be also nonlinear

The minimum you should do is a linear trend removal and, thus, the Linear trend removal option is checked as default. To remove also nonlinear drifts, a temporal high pass filter is recommended. This filter converts the time series at each voxel in the frequency domain (using a Fast Fourier Transform or FFT) and then removes low frequencies (= high pass). The frequency domain representation is then converted back in the time domain (inverse FFT), which looks then as before except that low frequency drifts are no longer visible. Since this filter is desirable in most cases, the option High pass filter: is checked as default. Prior to a high pass filter, linear trend removal is always performed because this improves the result of high pass filtering in case of a large linear drift component. You can specify which (low) frequencies should be removed either as cycles in time course, cycles/point or Hz. The unit selected as default is "cycles in time course" and the preset value is 3. This value needs not to be changed except in the case that your paradigm uses only a few (i.e. 3 or less) blocks or trials. In this case, you should reduce the value to 2 or 1 or turn high pass filtering off. Note, however, that it is highly recommended to use (much) more than 3 blocks or trials per run in order to be able to separate fluctuations due to true activation effects from nonlinear drifts due to scanner and physiological noise. If you chose to enter the high pass filter in *cycles per point* or *Hz*, the program converts the *cycles in time course* units for you. Note that using the unit *Hz* requires a correct specification of the TR value.

### 6.2.3 Gaussian temporal smoothing

Besides removal of drifts, you can also apply a gaussian temporal smoothing filter. Since temporal gaussian smoothing blurs timing information across neighboring data points, it is not recommended as default. Temporal smoothing improves, however, the signal-to-noise ratio by removing high frequency fluctuations. If you want to apply temporal smoothing, check the Gaussian option in the Temporal smoothing parameters field. The width of the kernel can now be specified in seconds. Note that the specification in seconds is only correct if the TR value has been specified correctly.

The kernel width is specified in the Gaussian - FWHM: text box with the default value of 2.8 seconds. If you want to specify the width of the kernel in units of data points (TR's), check the data points option instead of the secs option.

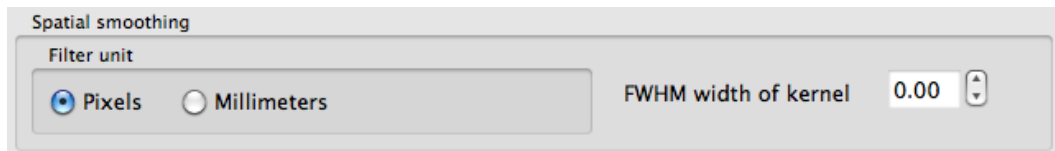
The file name segment reflecting temporal smoothing consists of up to three parts. If linear trend removal and/or high pass filtering is performed, the string `_LTR` (linear trend removal) is appended first. If a high pass filter is applied, the substring `_THP<filter><unit>` is appended. This is followed by the filter value and the unit of that value, which is either "c" (cycles in time course), "cp" (cycles per point) or "Hz". If temporal smoothing is performed, the substring `_TDTS<FWHM><unit>` is finally appended. The substring "TD" is short for "time domain", the substring "TS" is short for "temporal smoothing". This substring is followed by the width of the gaussian kernel (`<FWHM>`) followed by the unit of that value (`unit`), which is either "s" (seconds) or "dp" (data points). As an example, the string `_TDTS2.8s` in an FMR file name tells us that temporal smoothing had been performed in the time domain with a gaussian kernel of 2.8 seconds FWHM.

## Chapter 7

# Spatial filtering

### 7.1 Specifying parameters

Spatial low-pass (Gaussian) filter parameters can be entered via the ‘Spatial smoothing’ section on the “Preprocessing” tab of the wizard dialog (see figure 7.1). The width (FWHM, see below) of the filter kernel is specified in spatial units - either millimeters or pixels.



The image shows a software dialog box titled "Spatial smoothing". Inside the dialog, there is a section labeled "Filter unit" with two radio button options: "Pixels" (which is selected) and "Millimeters". To the right of these options, there is a label "FWHM width of kernel" followed by a text input field containing the value "0.00" and a small vertical spinner control.

Figure 7.1: Parameters for spatial filtering

## 7.2 Background: spatial smoothing in BrainVoyager

The information below is from the BrainVoyager 2000 User's Guide, just to provide general information about the filter units.

### 7.2.1 Introduction

Spatial smoothing. Spatial smoothing uses a 3D gaussian kernel. The width of the kernel can now be specified in millimeters. Note that the millimeter specification is only correct if the "voxel resolution" has been specified correctly (or more conveniently if it was extracted from the file header during project creation). The kernel width is specified in the Gaussian filter - FWHM: text box (FWHM = full width at half maximum). If you want to specify the width of the kernel in pixels, check the Pixel option in the Spatial smoothing parameters field. If you want to run gaussian smoothing only within the slices, check the 2D option instead of the 3D default option. Spatial smoothing is executed in the space domain (image space) as default, which is fast for small kernels. For large kernels (more than 15 mm), smoothing in the frequency domain is often faster. If you want to use frequency domain gaussian smoothing, check the Freq. domain option.

To exploit the full spatial resolution of fMRI, we do not recommend to spatially smooth the data for single subject analysis. This is the reason why the Spatial data smoothing option is turned off as default in the Preprocessing option field. If you want to smooth your data for single subject analysis, we recommend to use a small kernel of about "4 mm". For multi-subject analysis, we recommend a kernel between "8" and "12" mm. We suggest to apply spatial smoothing for multi-subject analysis after having first run single-subject analyses with no or modest spatial smoothing. The recommended moment for multi-subject spatial smoothing is at the stage where you want to run multi-subject GLM analyses. At this stage, you will have one or more VTC files per subject. To run spatial smoothing at this stage, you must use the version for VTC files, which is available in the 3D Data Preprocessing dialog (nb: VTC smoothing is now available via scripting and will be placed in a wizard soon). The file name segment reflecting spatial smoothing is

```
*_[SD|FD][3D|2D]SS<FWHM><unit>
```

. The substring "SD" is short for "space domain", the alternative string "FD" is short for "frequency domain". The next substring is either "3D" or "2D" followed by "SS", which is short for "spatial smoothing". This substring is followed by the width of the gaussian kernel (<FWHM>) followed finally by the unit of that value, which is either "mm" (millimeter) or "px" (pixels). As an example, the string \*\_SD3DSS4.00mm in an FMR file name tells us that 3D spatial smoothing had been performed in the space domain with a gaussian kernel of 4 mm FWHM.

# Bibliography

- [1] R. Goebel, F. Esposito, and E. Formisano. Analysis of fiac data with brainvoyager qx: From single-subject to cortically aligned group glm analysis and self-organizing group ica. *Human Brain Mapping*, 27(5):392–401, 2006.
- [2] E.H.W. Meijering, W.J. Niessen, and M.A. Viergever. Quantitative evaluation of convolution-based methods for medical image interpolation. *Medical Image Analysis*, 5(2):111–126, 2001.
- [3] P. Thevenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE Transactions on Medical Imaging*, 19(7):739–758, 2000.