

BXH/XCEDE Tools manual

BXH/XCEDE Tools manual

Table of Contents

1. Overview	1
1.1. Image Wrapping	1
1.1.1. BXH/XCEDE creation tools	1
1.1.2. BXH/XCEDE conversion tools	1
1.1.3. BXH/XCEDE manipulation tools	1
1.2. BXH/XCEDE QA tools	2
1.3. XML events	2
1.3.1. bxh_eventstats and related image processing and statistical tools	2
1.3.2. XML events file creation tools	3
1.3.3. XML events file manipulation tools	3
2. QA tools	4
2.1. BIRN Agar Phantom QA	4
A. BXH/XCEDE Tools usage	7
A.1. Usage for afni2bxh	7
A.2. Usage for analyze2bxh	8
A.3. Usage for batch_showplay2xml	9
A.4. Usage for bxh2analyze	9
A.5. Usage for bxh2pgm	10
A.6. Usage for bxh2ppm	11
A.7. Usage for bxh_brainmask	12
A.8. Usage for bxh_correlate	13
A.9. Usage for bxh_epochavg	14
A.10. Usage for bxh_event2table	17
A.11. Usage for bxh_eventmerge	18
A.12. Usage for bxh_eventresp	18
A.13. Usage for bxh_eventstats	19
A.14. Usage for bxh_eventstats_standardize	20
A.15. Usage for bxh_mean	20
A.16. Usage for bxh_tfilter	20
A.17. Usage for bxh_ttest	21
A.18. Usage for bxhabsorb	22
A.19. Usage for bxhreorient	25
A.20. Usage for bxhselect	26
A.21. Usage for bxhsetorient	26
A.22. Usage for dicom2bxh	27
A.23. Usage for dumpheader	28
A.24. Usage for eprime2xml	28
A.25. Usage for eventstable2xml	31
A.26. Usage for extractimagedata	33
A.27. Usage for extractxyztdata	34
A.28. Usage for ffile2bxh	34
A.29. Usage for fmriqa_count	34
A.30. Usage for fmriqa_generate.pl	35
A.31. Usage for fmriqa_minmax	36
A.32. Usage for fmriqa_oediff	36
A.33. Usage for fmriqa_phantomqa.pl	36
A.34. Usage for fmriqa_phantomqa	37
A.35. Usage for fmriqa_spikiness	37
A.36. Usage for iowa-signafive2bxh	38
A.37. Usage for pfile2bxh	39
A.38. Usage for presentation2xml	40

A.39. Usage for printfrags	42
A.40. Usage for showplay2xml	43
A.41. Usage for signafive2bxh	43
A.42. Usage for xcede_extract_schedules.pl	44
A.43. Usage for ximg2bxh	45
A.44. Usage for fmriqa_volmeasures	46

Chapter 1. Overview

There are four main components to the tools. The following sections describe each of these components in detail.

1.1. Image Wrapping

The "Image Wrapping" component contains tools that read, write and manipulate BXH or XCEDE header files that "wrap" image files.

1.1.1. BXH/XCEDE creation tools

These tools will create a BXH (or XCEDE, if the `--xcede` option is specified) XML file that "wraps" the image data in the various supported formats. The supported formats and programs are:

- Autodetect - `bxabsorb` (Section A.18)
- AFNI - `afni2bxh` (Section A.1)
- Analyze7.5/SPM/NIfTI-1 - `analyze2bxh` (Section A.2)
- DICOM - `dicom2bxh` (Section A.22)
- GE P-file - `pfile2bxh` (Section A.37)
- GE Signa 5.x - `signafive2bxh` (Section A.41)
- GE XImg - `ximg2bxh` (Section A.43)

Some of the important metadata in the image headers are extracted into the XML file using a standard naming scheme.

In a typical installation, most of these tools are symbolically linked to the same executable -- the behavior of the tool is switched based on the name of the link. This executable, **bxhabsorb**, attempts to autodetect the format of the input images, whereas the other tools assume a given input format.

1.1.2. BXH/XCEDE conversion tools

These tools convert from BXH or XCEDE into other image formats:

- `bxh2analyze` (Section A.4)
- `bxh2pgm` (Section A.5)
- `bxh2ppm` (Section A.6)

1.1.3. BXH/XCEDE manipulation tools

These tools manipulate the BXH or XCEDE file in various ways. Any single non-XML image file can also be sent as an input, in place of a BXH/XCEDE file, as long as `bxhabsorb` can recognize it (if you send a single DICOM file, it will act as if the `--search-for-others` option is specified). Note that you cannot specify multiple image files (say, a set of 3-D NIfTI-1 files) using this option -- in that case, you need to use one of the Image Wrapping tools listed in Section 1.1.

- `bxhreorient` (Section A.19)
- `bxhselect` (Section A.20)
- `bxhsetorient` (Section A.21)
- `dumpheader` (Section A.23)
- `extractimagedata` (Section A.26)
- `extractxyzdata` (Section A.27)
- `printfrags` (Section A.39)

1.2. BXH/XCEDE QA tools

These tools perform QA (quality assurance) calculations and produce images, graphs, and/or XML data as output. **`fmriqa_phantomqa.pl`** and **`fmriqa_generate.pl`** produce an HTML report with various QA measures. **`fmriqa_phantomqa.pl`** was designed for fMRI images of the BIRN stability phantom, and **`fmriqa_generate.pl`** has been used for human fMRI data. These two tools depend on various subsidiary tools (listed below) to perform various tasks, which can be used individually.

As mentioned in \ref manipulation, any single non-XML image file can also be sent as an input to all of the tools that don't end in `.pl` (i.e. the tools that are not perl scripts).

- `fmriqa_phantomqa.pl` (Section A.33)
- `fmriqa_generate.pl` (Section A.30)
- `fmriqa_count` (Section A.29)
- `fmriqa_minmax` (Section A.31)
- `fmriqa_oediff` (Section A.32)
- `fmriqa_phantomqa` (Section A.34)
- `fmriqa_spikiness` (Section A.35)
- `fmriqa_volmeasures` (Section A.44)

1.3. XML events

1.3.1. `bxh_eventstats` and related image processing and statistical tools

`bxh_eventstats` is an event-based epoch averaging tool. It collects event-synchronized "snippets" of the fMRI response, averages them, and optionally correlates them to a template hemodynamic response or to other averaged "snippets". The times of the chosen fragments are selected by querying XML events files for events matching given characteristics. **`bxh_eventstats`** uses various subsidiary tools that may also be used directly.

- `bxh_eventstats` (Section A.13)
- `bxh_brainmask` (Section A.7)

- `bxh_correlate` (Section A.8)
- `bxh_epochavg` (Section A.9)
- `bxh_mean` (Section A.15)
- `bxh_tfilter` (Section A.16)
- `bxh_ttest` (Section A.17)

Various screeds on topics related to **bxh_eventstats** can be found in ???.

1.3.2. XML events file creation tools

These tools are used to create the XML events files used by **bxh_eventstats** and other tools. Event data can currently be extracted from text files generated by E-Prime, Presentation, CIGAL, and other software that generates tabular text data.

- `eprime2xml` (Section A.24)
- `presentation2xml` (Section A.38)
- `showplay2xml` (Section A.40)
- `eventstable2xml` (Section A.25)

1.3.3. XML events file manipulation tools

These tools are used to manipulate XML events files:

- `bxh_event2table` (Section A.10)
- `bxh_eventmerge` (Section A.11)
- `bxh_eventresp` (Section A.12)

Chapter 2. QA tools

This package provides several tools to perform QA (quality assurance) calculations. They produce images, graphs, and/or XML data as output. **fmriqa_phantomqa.pl** and **fmriqa_generate.pl** are the main tools and produce an HTML report with various QA measures. **fmriqa_phantomqa.pl** was designed for fMRI images of the BIRN stability phantom, and **fmriqa_generate.pl** has been used for human fMRI data.

2.1. BIRN Agar Phantom QA

The tool **fmriqa_phantomqa.pl** implements several recommendations of the fBIRN Calibration working group in analyzing fMRI data from a spherical agar-filled phantom. A subset of these recommendations come from the following publication:

Notes are included below for any metrics whose implementation differs from those described in that report.

To run the tool, provide a minimum of two arguments -- first, the image data as wrapped by a BXH or XCEDE file, and second, the name of the directory where the outputs should go. This directory should not exist, unless the `--overwrite` option is specified. So, for example, if you have a set of DICOM files, you could do something like:

```
dicom2bxh *.dcm WRAPPED.bxh
fmriqa_phantomqa.pl WRAPPED.bxh OUTPUTQADIR
```

Several options to the QA tool are available, and documentation of these are available by running **fmriqa_phantomqa.pl --help**.

To avoid cutting and pasting wholesale from the text of the paper, we refer to the Friedman, Glover (2006) paper for documentation and justifications of several metrics. The differences between (and additions to) the metrics described in the paper are described below:

Initially discarded volumes

If the number of timepoints in the data is even, then the first 2 timepoints are discarded, as recommended in the paper. However, since several metrics depend on an even number of volumes, if the number of timepoints is odd, then 3 timepoints are discarded.

ROI

The ROI used in these computation (when specified) is an NxN square in the center of the middle slice of the volume. This tool uses N=15 voxels for a 64x64 voxel image by default, but can be adjusted with the `--roisize` option. The current recommendation from the paper for a 64x64 voxel image is an ROI of 21x21 voxels, but the default remains at 15x15 to maintain compatibility with earlier revisions of the protocol.

Drift

In the paper, the signal *drift* is calculated on the second-order polynomial fit of a signal composed of the mean intensities for each volume across the ROI. This value is listed as **driftfit** in the output of this QA tool. The value **drift** as calculated by this tool,

however, is calculated on the unfitted (i.e. raw) signal.

Fourier analysis

The y-axis in the frequency spectrum images generated by this tool is scaled by the mean of the raw mean ROI signal (see above), and matches what the paper suggests will be implemented in a "future version" of the protocol. To limit any confusion, the y-axis of the plot is annotated as "mean scaled".

Additions

Several newer analyses beyond those described in the paper have been added to the QA report (and are documented below).

The GSL function **gsl_multifit_linear()** is used to perform the polynomial fit for the metrics described in the Friedman, Glover paper.

The output directory will contain several images and data files, including an XML file 'summaryQA.xml' which encodes many of the QA metrics into XCEDE2 format. The images and several acquisition and summary values can be displayed by opening the file **index.html** in any web browser.

At the top of index.html are a series of images based on recommendations from the Friedman, Glover paper:

Raw signal

This image plots the mean intensity of the raw signal across the ROI for each timepoint, as well as the second-order polynomial fit of the same signal. Displayed in the graph are the summary values *percent fluctuation*, *drift* and *driftfit* (see note above). These values are also available in **summaryQA.xml** as "percentFluc", "drift", and "driftfit".

Magnitude spectrum

This image plots a Fourier transform of the signal, with the y-axis scaled to be a percentage of the mean of the raw signal intensity across the ROI. Some other summary values are also displayed here: "mean", "SNR", and "SFNR" (also in **summaryQA.xml** with the same names).

Weiskoff analysis

This plots the observed and theoretical coefficients of variation for voxels within various ROI sizes up to the maximum ROI size.

Several more images follow:

FWHM (full-width half-maximum)

If AFNI tools were installed when running the QA, three images plotting the FWHM values over time (in each of X, Y, and Z dimensions). FWHM is calculated on the motion-corrected, 2nd-order polynomial detrended, masked data. The order of operations follows this sequence (note that 2 or 3 volumes [by default] are removed from the input before executing):

```
3dvolreg --prefix REG INPUT
3dDetrend --polort 2 --prefix
DETREND REG
```

```
3dTstat --mean --prefix MEAN REG
3dAutomask --q --prefix MASK MEAN
3dFWHMx --dset DETREND --mask MASK --
out FWHMVALS
```

Center of Mass

The center of mass in the X, Y, or Z dimension is the weighted mean intensity over each volume, where the weights are the X, Y, or Z indices of each voxel. If the phantom is always fully within the field of view, this can be a simple indication of whether the object being imaged somehow changed position during scanning. If AFNI tools were installed when running the QA, the more sophisticated output of 3dvolreg is also plotted here (shifted along the y-axis to match the first value of the center-of-mass).

Ghost analysis

For EPI sequences, this image plots the "ghostiness" of the data. The ghost metric is calculated for each volume by calculating a dilated mask ("original mask") of the motion-corrected, detrended data, and shifting it by N/2 voxels in the appropriate axis to create a "ghost mask". The mean intensities of those voxels in the original mask and not in the ghost mask, and of voxels in the ghost mask and not in the original mask are calculated. The mean intensity of the top 10 percent of ghost-only voxels ("meanBrightGhost") is also calculated. The ghost values are plotted as a percentage of the mean intensity of non-ghost voxels. The dilated mask is generated using the following sequence of AFNI commands:

```
3dvolreg --prefix REG INPUT
3dDetrend --polort 2 --prefix
DETREND REG
3dTstat --mean --prefix MEAN REG
3dAutomask --q --dilate 4 --prefix
MASK MEAN
```

Four more images from the original Friedman, Glover paper follow:

Odd-even difference

This corresponds to the "Static Spatial Noise Image" in the paper.

Mean

This corresponds to the "Signal Image" in the paper.

Standard Deviation

This corresponds to the "Temporal Fluctuation Noise Image" in the paper.

SFNR

This corresponds to the "Signal-toFluctuation-Noise Ratio (SFNR) Image" in the paper.

After this follows a table of the acquisition parameters, if available, extracted from the original input data files.

A.1. Usage for afni2bxh

Usage :

```
afni2bxx [opts] afnifile.HEAD output.bxx
```

This program creates an XML wrapper for AFNI images.

```
--inputsfromfile <str>
--inputsfromfile=<str>
    Read list of input files from this file.
--version
    Print version string and exit.
--hintsize x <size_t>
--hintsize x=<size_t>
--hintsize y <size_t>
--hintsize y=<size_t>
--hintsize z <size_t>
--hintsize z=<size_t>
--hintsize t <size_t>
--hintsize t=<size_t>
--hintorigin x <double>
--hintorigin x=<double>
--hintorigin y <double>
--hintorigin y=<double>
--hintorigin z <double>
--hintorigin z=<double>
--hintorigin t <double>
--hintorigin t=<double>
--hintspacing x <double>
--hintspacing x=<double>
--hintspacing y <double>
--hintspacing y=<double>
--hintspacing z <double>
--hintspacing z=<double>
--hintspacing t <double>
--hintspacing t=<double>
--hintgap x <double>
--hintgap x=<double>
--hintgap y <double>
--hintgap y=<double>
--hintgap z <double>
--hintgap z=<double>
--hintgap t <double>
--hintgap t=<double>
    These options will provide a hint to the program of the 'size',
    'origin', 'spacing', or 'gap' of the specified dimension. Some image
    types will not use all these values. In particular, sizex and sizey
    are assumed correct in most image headers, but they, as well as sizez
    and sizet options may be useful with image type 'pfile'. Origin and
    spacing hints will be used by most image types.
--forceorientation <str>
--forceorientation=<str>
    This option will force the labeled orientation of the image to match
    the given three letter orientation code. Each letter must come from
    the following groups in any order: R(ight) or L(ef); A(nterior) or
    P(osterior); S(uperior) or I(nferior). Only one letter from each group
    is allowed.
--xcede
    Write XCEDE-style XML files.
--xcede2
```

Write XCEDE 2-style XML files.

A.2. Usage for analyze2bxh

Usage:

```
analyze2bxh [opts] [analyzefiles...] output.bxh
```

This program creates an XML wrapper for Analyze7.5/SPM/NIfTI-1 images.

```
--orientation <str>
--orientation=<str>
    Orientation of image, letters indication which way theX, Y, and Z
    dimensions (in that order) are pointing (e.g. LPS, IRP). Default is
    RAS (i.e. orientation used by SPM), or that specified in accompanying
    SPM .mat files. This option overrides all info in SPM .mat files
    and/or the Analyze 'orient' field (if --strictanalyze is specified).
    Equivalent to (and overrides) --forceorientation.
--strictanalyze
    Don't use SPM .mat files and use Analyze convention for orientation.
    The 'orient' field in the analyze header is parsed, and the default
    case (i.e. it is zero) means 'LAS'.
--avwbyteorder <str>
--avwbyteorder=<str>
    Specify byte order for AVW files (which don't store this info). This
    field should be 'l' for little-endian or 'b' for big-endian.
--inputsfromfile <str>
--inputsfromfile=<str>
    Read list of input files from this file.
--version
    Print version string and exit.
--hintsizeX <size_t>
--hintsizeX=<size_t>
--hintsizeY <size_t>
--hintsizeY=<size_t>
--hintsizeZ <size_t>
--hintsizeZ=<size_t>
--hintsizeT <size_t>
--hintsizeT=<size_t>
--hintoriginX <double>
--hintoriginX=<double>
--hintoriginY <double>
--hintoriginY=<double>
--hintoriginZ <double>
--hintoriginZ=<double>
--hintoriginT <double>
--hintoriginT=<double>
--hintspacingX <double>
--hintspacingX=<double>
--hintspacingY <double>
--hintspacingY=<double>
--hintspacingZ <double>
--hintspacingZ=<double>
--hintspacingT <double>
--hintspacingT=<double>
--hintgapX <double>
--hintgapX=<double>
--hintgapY <double>
--hintgapY=<double>
--hintgapZ <double>
--hintgapZ=<double>
--hintgapT <double>
--hintgapT=<double>
    These options will provide a hint to the program of the 'size',
```

```

'origin', 'spacing', or 'gap' of the specified dimension. Some image
types will not use all these values. In particular, sizex and sizey
are assumed correct in most image headers, but they, as well as sizez
and sizet options may be useful with image type 'pfile'. Origin and
spacing hints will be used by most image types.
--forceorientation <str>
--forceorientation=<str>
    This option will force the labeled orientation of the image to match
    the given three letter orientation code. Each letter must come from
    the following groups in any order: R(ight) or L(eft); A(nterior) or
    P(osterior); S(uperior) or I(nferior). Only one letter from each group
    is allowed.
--xcede
    Write XCEDE-style XML files.
--xcede2
    Write XCEDE 2-style XML files.

```

A.3. Usage for batch_showplay2xml

```

Can't locate JSON/PP.pm in @INC (@INC contains: /builddir/build/BUILD/RELEASE_1_10_7/
bxh_xcede_tools/utils /usr/local/lib64/perl5 /usr/local/share/perl5 /usr/lib64/perl5/
vendor_perl /usr/share/perl5/vendor_perl /usr/lib64/perl5 /usr/share/perl5 .) at /
builddir/build/BUILD/RELEASE_1_10_7/bxh_xcede_tools/utils/BXHPerlUtils/XMLMetadata.pm
line 3.
BEGIN failed--compilation aborted at /builddir/build/BUILD/RELEASE_1_10_7/
bxh_xcede_tools/utils/BXHPerlUtils/XMLMetadata.pm line 3.
Compilation failed in require at /builddir/build/BUILD/RELEASE_1_10_7/bxh_xcede_tools/
utils/BXHPerlUtils.pm line 13.
BEGIN failed--compilation aborted at /builddir/build/BUILD/RELEASE_1_10_7/
bxh_xcede_tools/utils/BXHPerlUtils.pm line 13.
Compilation failed in require at ../../utils/batch_showplay2xml line 12.
BEGIN failed--compilation aborted at ../../utils/batch_showplay2xml line 12.

```

A.4. Usage for bxh2analyze

```

Usage:
  bxh2analyze [opts] input.bxh outputprefix

  xcede2analyze [opts] input.bxh outputprefix

```

Both programs create Analyze 7.5, SPM, or NifTI-1 images from BXH- or XCEDE-
 wrapped images (both programs support both formats). NOTE: XCEDE is only
 supported if this program was compiled with XSLT support (which in this case,
 it has; congratulations!).

```

--version
    Print version string and exit.
--overwrite
    Overwrite files if they exist.
--bxh
    This option forces output XML file to be BXH (this option is ignored by
    [and is the default behavior of] bxh2analyze)
--xcede
    This option forces output XML file to be XCEDE (this option is ignored
    by [and is the default behavior of] xcede2analyze)
--xcede2
    This option forces output XML file to be XCEDE2.
-b
    This option suppresses the writing of a BXH/XCEDE header for every

```

```

        Analyze header and image file.
-s
    This option suppresses the writing of an SPM .mat file for every
    Analyze header and image file.
-i
    This option suppresses the writing of the image (.img) files.
-h
    This option suppresses the writing of the Analyze header (.hdr) files.
-B
    If writing BXH/XCEDE headers, instead of writing one header, this
    option forces the writing of several BXH/XCEDE headers, one per Analyze
    header and image file.
-v
    This option suppresses the splitting a time series into separate
    volumes. If not specified, then each image file will contain the data
    for one volume. If specified, then each time series will be output as
    one large file.
--niftihdr
    Generate NIfTI-1 format header (default is to attempt to generate a
    maximally compatible header).
--nii
    Use NIfTI-1 one-file convention (header and data in same file). Output
    image files will have the .nii extension. This option automatically
    turns on --niftihdr and -v.
--niigz
    Same as (and overrides) --nii, but the output will be compressed and
    will have the .nii.gz extension.
--spmhdr
    Assume header will only be read by SPM, using SPM-specific values when
    possible (default is to attempt to generate a maximally compatible
    header).
--analyzehdr
    Generate "pure" Analyze 7.5 header, whatever that means. (default is
    to attempt to generate a maximally compatible header).
--preferanalyzetypes
    Prefer the use of only Analyze 7.5 pixel types if we are writing SPM or
    NIfTI headers; non-Analyze 7.5 pixel types will be used if
    overflow/underflow is detected. This option requires either the
    --spmhdr or --niftihdr options.
--analyzetypes
    Force the use of only Analyze 7.5 pixel types, even if we are writing
    SPM or NIfTI headers. If necessary, this will cause conversion to a
    data type with higher range.
--nosform
    For NIfTI headers, do not write orientation information into the sform
    fields. Default is to write both qform and sform.
--spatialunits <str>
--spatialunits=<str>
    Force spatial units to a given value. Must be 'm', 'mm', or 'um'.
    Output metadata will not be converted if this does not correctly
    represent the input metadata.
--temporalunits <str>
--temporalunits=<str>
    Force temporal units to a given value. Must be 's', 'ms', or 'us'.
    Output metadata will not be converted if this does not correctly
    represent the input metadata.
--zeroorigin
    Force the origin fields in the output images to be [0,0,0] (this is
    useful if your analysis software uses 0,0,0 to mean 'unspecified', and
    you don't want to specify the origin).

```

A.5. Usage for bxh2pgm

Usage:

```
bxh2pgm input.bxh output.pgm
```

This program converts images wrapped with a BXH or XCEDE header into PGM format. 3-D or higher dimensionality images are represented as a sequence of 2-D images.

```
--version
    Print version string and exit.
--colorbar <str>
--colorbar=<str>
    Write a horizontal colormap 'bar' to this PGM file.
--colorbarorient <str>
--colorbarorient=<str>
    Orientation of colorbar, either 'horizontal' (default) or 'vertical'.
--barwidth <uint>
--barwidth=<uint>
    Width (in pixels) of colormap 'bar' (default 16).
--barlength <uint>
--barlength=<uint>
    Length (in pixels) of colormap 'bar' (default 256).
--maxval <double>
--maxval=<double>
    By default, if the input element type is floating-point or if the
    maximum input value is greater than 65535, the maximum value in the
    input will be mapped to 65535 (the highest possible PGM value) in the
    output PGM image. --maxval specifies an alternative maximum input
    value. Input values greater than this will be clipped.
--minval <double>
--minval=<double>
    By default, the minimum value in the input will be mapped to 0 (the
    lowest possible PGM value) in the output PGM image. --minval specifies
    an alternative minimum input value. Input values smaller than this
    will be clipped.
--dimorder <str>
--dimorder=<str>
    Specify dimension order as a comma-separated list of dimension names.
--timeselect <str>
--timeselect=<str>
    Comma-separated list of timepoints to use (first timepoint is 0). Any
    timepoint can be a contiguous range, specified as two numbers separated
    by a colon, i.e. 'START:END'. An empty END implies the last timepoint.
    The default step of 1 (one) in ranges can be changed using
    'START:STEP:END', which is equivalent to
    'START,START+STEP,START+(2*STEP),...,END'.
--xselect <str>
--xselect=<str>
    Just like timeselect, but for the 'x' dimension.
--yselect <str>
--yselect=<str>
    Just like timeselect, but for the 'y' dimension.
--zselect <str>
--zselect=<str>
    Just like timeselect, but for the 'z' dimension.
```

A.6. Usage for bxh2ppm

Usage:

```
bxh2ppm input.bxh output.ppm
```

This program converts images wrapped with a BXH or XCEDE header into PPM format. 3-D or higher dimensionality images are represented as a sequence of 2-D images.

```

--version
    Print version string and exit.
--colorbar <str>
--colorbar=<str>
    Write a colormap 'bar' to this PPM file.
--nobracket
    If --colorbar specified, then --nobracket disables the 'bracket' in the
    colorbar that shows the color range displayed by this image.
--colorbarorient <str>
--colorbarorient=<str>
    Orientation of colorbar, either 'horizontal' (default) or 'vertical'.
--barwidth <uint>
--barwidth=<uint>
    Width (in pixels) of colormap 'bar' (default 16).
--barlength <uint>
--barlength=<uint>
    Length (in pixels) of colormap 'bar' (default 256).
--maxval <double>
--maxval=<double>
    By default, if the input element type is floating-point or if the
    maximum input value is greater than 65535, the maximum value in the
    input will be mapped to the color corresponding to the highest value in
    the output PPM image. --maxval specifies an alternative maximum input
    value. Input values greater than this will be clipped.
--minval <double>
--minval=<double>
    By default, the minimum value in the input will be mapped to the color
    corresponding to the lowest value in the output PPM image. --minval
    specifies an alternative minimum input value. Input values smaller
    than this will be clipped.
--colormap <str>
--colormap=<str>
    Use this colormap for converting input values to colors. Valid
    colormaps are 'hot', 'bluered', 'grayhot', and 'gray' (default).
--dimorder <str>
--dimorder=<str>
    Specify dimension order as a comma-separated list of dimension names.
--timeselect <str>
--timeselect=<str>
    Comma-separated list of timepoints to use (first timepoint is 0). Any
    timepoint can be a contiguous range, specified as two numbers separated
    by a colon, i.e. 'START:END'. An empty END implies the last timepoint.
    The default step of 1 (one) in ranges can be changed using
    'START:STEP:END', which is equivalent to
    'START,START+STEP,START+(2*STEP),...,END'.
--xselect <str>
--xselect=<str>
    Just like timeselect, but for the 'x' dimension.
--yselect <str>
--yselect=<str>
    Just like timeselect, but for the 'y' dimension.
--zselect <str>
--zselect=<str>
    Just like timeselect, but for the 'z' dimension.

```

A.7. Usage for bxh_brainmask

Usage:
 bxh_brainmask [opts] inputfile outputfile

This program will attempt to create a simple (thresholded) brain mask given a
 BXH- or XCEDE-wrapped input image. Output is also a BXH- or XCEDE-wrapped

input image. Calculation of the threshold is modified using various options.

```

--version
    Print version string and exit.
--timeselect <str>
--timeselect=<str>
    Comma-separated list of timepoints to use (first timepoint is 0). Any
    timepoint can be a contiguous range, specified as two numbers separated
    by a colon, i.e. 'START:END'. An empty END implies the last timepoint.
    The default step of 1 (one) in ranges can be changed using
    'START:STEP:END', which is equivalent to
    'START,START+STEP,START+(2*STEP),...,END'. Default is all timepoints
    (:).
--xselect <str>
--xselect=<str>
    Just like timeselect, but for the 'x' dimension.
--yselect <str>
--yselect=<str>
    Just like timeselect, but for the 'y' dimension.
--zselect <str>
--zselect=<str>
    Just like timeselect, but for the 'z' dimension.
--overwrite
    Overwrite existing output files (otherwise error and exit).
--method <str>
--method=<str>
    Method to use for creating the brain mask.
    'threshold' marks those voxels whose mean value over time are not less
    than a given threshold (provided by --filterthresh).
    'rank' chooses the largest threshold that allows at least the n
    highest-valued voxels (as determined by the voxel's mean value over
    time) where n is specified by --filterrank.
    'localmin' fits a nth-order polynomial (order optionally specified by
    --filterorder) to an intensity histogram of the minimum value of each
    voxel over time, and chooses the first local minimum (disregarding the
    first histogram bucket) as the threshold. This method assumes the data
    follows an intensity distribution with at least two "humps", the first
    (lower) of which reflects noise.
    Default is 'rank'.
--filterorder <uint>
--filterorder=<uint>
    Order of the polynomial used for --method localmin. Default is 5.
--filterthresh <str>
--filterthresh=<str>
    Threshold used for --method threshold. If value ends with the percent
    sign (%), then this is taken as a percent of maximum intensity.
    Default is '50%'.
--filterrank <str>
--filterrank=<str>
    Threshold used for --method rank. If value ends with the percent sign
    (%), then this is taken as a percent of the number of total voxels.
    Default is '20%'.
--debug
    Print out debugging messages.

```

A.8. Usage for bxh_correlate

Usage:

```

bxh_correlate [opts] --template T1,T2,T3... inputxmlfile out_rfile
[out_tfile]

```

This program correlates the time series of each voxel in a 4-D time series of volumes (inputxmlfile) with a given "template" vector (specified with

```

--template option). Output (in out_rfile) is a 3-D data set storing the
correlation coefficient (r). The optional third argument (out_tfile) is where
to write the 3-D data set storing the corresponding t-statistic (derived from
r).

--version
    Print version string and exit.
--optsfromfile <str>
--optsfromfile=<str>
    Program options (i.e. those starting with '--') will come from this
    file. If this option is specified, then the options in the file will
    be applied after all command-line options. The options (and their
    arguments) should be specified one per line, with the leading '--'
    omitted.
--overwrite
    Overwrite existing output files (otherwise error and exit).
--timeselect <str>
--timeselect=<str>
    Comma-separated list of timepoints to use (first timepoint is 0). Any
    timepoint can be a contiguous range, specified as two numbers separated
    by a colon, i.e. 'START:END'. An empty END implies the last timepoint.
    The default step of 1 (one) in ranges can be changed using
    'START:STEP:END', which is equivalent to
    'START,START+STEP,START+(2*STEP),...,END'. Default is all timepoints
    (:).
--xselect <str>
--xselect=<str>
    Just like timeselect, but for the 'x' dimension.
--yselect <str>
--yselect=<str>
    Just like timeselect, but for the 'y' dimension.
--zselect <str>
--zselect=<str>
    Just like timeselect, but for the 'z' dimension.
--template <str>
--template=<str>
    A comma-separated list of numbers making up the template vector to
    correlate with the data. This option or --templatevoxel is required.
--templatevoxel <str>
--templatevoxel=<str>
    A comma-separated x,y,z coordinate (indices start at 0) indicating
    which voxel in the dataset to which to do the correlation. The value
    at that voxel in the output will be 1.0. This option or --template is
    required.
--maskfile <str>
--maskfile=<str>
    Use this 3-D mask (should be an XML file) before doing calculations.

```

A.9. Usage for bxh_epochavg

Usage:

```

bxh_epochavg [opts] outputprefix imgfile1 eventfile1[,eventfile1b,...]
[imgfile2 eventfile2[,eventfile2b,...] ...]

```

This program "queries" a 4-D data set (with corresponding event lists) and produces averages of all time courses surrounding each event that match the query. Multiple independent queries may be specified, and the width and position and duration of each time course relative to the event is also user-specified. Multiple event files corresponding to the same image data can be specified separated by commas (the filenames/paths themselves are therefore prohibited from containing commas).

```

--version

```

```

    Print version string and exit.
--optsfromfile <str>
--optsfromfile=<str>
    Program options (i.e. those starting with '--') will come from this
    file. If this option is specified, then the options in the file will
    be applied after all command-line options. The options (and their
    arguments) should be specified one per line, with the leading '--'
    omitted.
--overwrite
    Overwrite existing output files (otherwise error and exit).
--maskfile <str>
--maskfile=<str>
    Use this 3-D mask (should be an XML file) before doing calculations.
--querylanguage <str>
--querylanguage=<str>
    The language used for all queries. Valid values are 'XPath' and
    'event'. Case is irrelevant. Default is 'XPath'.
--query <str>
--query=<str>
    A query string as an XPath boolean expression. Will be applied as a
    predicate filter to each event. Each event node may or may not have
    onset, duration, type, and value elements (as well as others).
    Examples:
    --query "value[@name='color']='red'"
    --query "value[@name='color']='red' or value[@name='color']='blue'"
    --query "(value[@name='color']='red' or value[@name='color']='blue')
    and not value[@name='field']='upper' and onset>12000"
    Note that some characters in queries may need to be protected from the
    shell with quotes (as in the above examples). Separate instances of
    the --query option will result in independent queries, with separate
    outputs. Empty queries match all events. NOTE: At least one query
    must be specified!
--queryfilter <str>
--queryfilter=<str>
    If this option is specified, it is an XPath query (like --query) that
    is applied to a list of pseudo-events, each pseudo-event corresponding
    to an event matching the original query. Each pseudo-event is a
    merging of all events that are simultaneously in effect at the time of
    the onset of the real event. If this query matches the pseudo-event,
    the real event passes through the filter. The n-th instance of this
    option corresponds to the n-th specified query. If any --queryfilter
    options are specified, there should be exactly one --queryfilter per
    --query. Empty or missing filter queries match everything.
--queryepochexclude <str>
--queryepochexclude=<str>
    Like --query, --queryepochexclude specifies an XPath-based event query.
    However, any epoch that includes an event that matches this query will
    be excluded from the analysis. The epoch surrounding an event is
    specified using --ptsbefore and --ptsafter (or --secsbefore and
    --secsafter). The n-th instance of this option corresponds to the n-th
    specified query. If any --queryepochexclude options are specified,
    there should be exactly one --queryepochexclude per --query. Empty or
    missing epoch exclusion queries exclude nothing.
--querylabel <str>
--querylabel=<str>
    A textual label for the corresponding query. The first instance of
    this option corresponds to the first specified query. There should be
    at most one --querylabel per --query. Default label is the query
    number.
--ptsbefore <int>
--ptsbefore=<int>
    How many time points before the event to include in analysis. This
    option (or --secsbefore) is required.
--ptsafter <int>
--ptsafter=<int>
    How many time points after the event to include in analysis. This
    option (or --secsafter) is required.

```

```

--secsbefore <double>
--secsbefore=<double>
    How many seconds before the event to include in analysis. This option
    (or --ptsbefore) is required.
--secsafter <double>
--secsafter=<double>
    How many seconds after the event to include in analysis. This option
    (or --ptsafter) is required.
--basestartoffset <int>
--basestartoffset=<int>
    Where to start calculating mean baseline, in number of timepoints (TRs)
    relative to event time. A negative number refers to a timepoint before
    the event, 0 is at the time of the event, and a positive number is
    after the event. Default is 0.
--baseendoffset <int>
--baseendoffset=<int>
    Where to end calculating mean baseline, in number of timepoints (TRs)
    relative to event time. A negative number refers to a timepoint before
    the event, 0 is at the time of the event, and a positive number is
    after the event. Default is 0.
--startpt <uint>
--startpt=<uint>
    This number of time points at the start of the data will be ignored.
    Default is 0.
--endpt <uint>
--endpt=<uint>
    Time points after this point will be ignored. Default is last
    timepoint.
--forcetr <double>
--forcetr=<double>
    If specified, this value (in seconds) will replace the TR specified in
    the input image file, if any.
--nointerp
    If specified, no interpolation will be done -- events will be assumed
    to occur at the closest TR/image acquisition time.
--scalebl
    If specified, values in each epoch are additionally scaled by dividing
    by (after subtracting) the baseline. This affects the 'avg' and 'std'
    output images. Percent signal-change images are not written. WARNING:
    Know what you are doing before using this option.
--extracttrials
    If this option is specified, the program will write out epochs for
    *all* extracted trials to a file PREFIX_QUERY_trials.bxh. This file
    will be a 5-D image file where the 4th dimension goes across time
    points within an epoch, and the 5th dimension represents the global
    trial number.
--trialssummary <str>
--trialssummary=<str>
    This option enables the creation of "summaries" of trials before
    averaging, where summaries are new trials where each point is an
    average of some number of timepoints in the original trial. The string
    argument is of the form "QUERY-PTS", where QUERY is a query label, and
    PTS is a plus('+')-separated list of "index groups", an "index group"
    is a comma-separated list of indices or ranges (which are two indices
    separated by a colon). For example "red-0" will create a summary
    trials containing only the first point in each trial that matches the
    "red" query, and "red-0:3+4:7+8:11" or "red-0,1,2,3+4,5,6,7+8,9,10,11"
    (both are equivalent) will average together the 12 timepoints of each
    "red" trial in groups of 4. The outputs will be similar to other
    outputs but will look like PREFIX_QUERY_summary_PTS_avg.bxh etc., but
    with the colons (':') replaced with dashes ('-') due to problems some
    file systems have with colons. Note that timepoints are indexed from
    0. This option may be specified more than once.
--trialmax
    This is an EXPERIMENTAL option. If specified, a 'seed' timepoint and
    voxel is found within the optional ROI specified by --trialmaxroi. The
    seed timepoint is defined as the timepoint within the epoch average

```

that has the highest mean intensity. The seed voxel is then defined as the voxel with the highest value within the seed timepoint. Then, for each voxel, a 'trial sequence' is constructed containing the value of that voxel at the seed timepoint within each individual epoch (before averaging). The output is a 4-D series of volumes (one for each trial) named PREFIX_QUERY_trialmax.bxh that contains the volumes at the seed timepoint in each trial. The seed voxel coordinates are written to PREFIX_QUERY_trialmaxseed.txt.

```
--trialmaxroi <str>
--trialmaxroi=<str>
    The ROI used by --trialmax.
--trialmaxseed <str>
--trialmaxseed=<str>
    This specifies an explicit comma-separated coordinate X,Y,Z,T for the
    seed for --trialmax, to be applied to ALL queries. The T coordinate
    must be in the range [0,s-1] where s is the number of time points in
    the epoch. Note that timepoints are indexed from 0.
--extracttimingonly
    If specified, only the PREFIX_LABEL_timing.txt files will be written.
--memorylimit <double>
--memorylimit=<double>
    This specifies the number of megabytes of the input data to read at a
    time. Default is to read the entire data at once. If you are running
    out of memory due to high-resolution data, or large numbers of
    timepoints, this is one way to reduce memory usage. This is not an
    overall memory usage limit -- actual memory usage will surely be much
    higher than this.
```

A.10. Usage for bxh_event2table

Usage:

```
bxh_event2table [opts] eventfiles...
```

This program takes XML event files as input, selects events (given user-specified queries), and writes a table of these events and associated metadata to standard output. Each row is one event, and each column represents a different metadata element (like onset, duration, and other values specified in the events file).

```
--version
    Print version string and exit.
--optsfromfile <str>
--optsfromfile=<str>
    Program options (i.e. those starting with '--') will come from this
    file. If this option is specified, then the options in the file will
    be applied after all command-line options. The options (and their
    arguments) should be specified one per line, with the leading '--'
    omitted.
--querylanguage <str>
--querylanguage=<str>
    The language used for all queries. Valid values are 'XPath' and
    'event'. Case is irrelevant. Default is 'XPath'.
--query <str>
--query=<str>
    A query string to match events. This option is required.
--filterquery <str>
--filterquery=<str>
    A query string to filter matched events.
--colsep <str>
--colsep=<str>
    String to separate columns (default is tab).
```

A.11. Usage for bxh_eventmerge

Usage:

```
bxh_eventmerge [ --debug ] [ --eventpath XPATH ] [ --mergeeventpath XPATH ] [ --mergequery XPATH ] [ --grabincludes XPATH ] [ --grabexcludes XPATH ] INPUTQUERY GRABQUERY inputevents1.xml inputevents2.xml... mergeevents.xml
```

This program takes several input files (inputevents*.xml) and "merges" the information from another event file (mergeevents.xml) into each input file. Here is the algorithm:

1. Create sets of event nodes in the input and merge event files using the XPaths specified by --eventpath and --mergeeventpath. Default for --eventpath, if not specified, is //events/event (but namespace-agnostic), and default for --mergeeventpath is the specified or default value of the --eventpath option.
2. Each event node in the input event files will have a "match" value created by applying the XPath INPUTQUERY.
3. Each event node in the merge event file will have a "match" value created by applying the XPath specified by the --mergequery option (which is set to INPUTQUERY by default).
4. For each event node in the input event file whose "match" value is not an empty string, and which matches the "match" value of an event node in the merge event file:
 - a. Apply GRABQUERY to the matching merge event, and recursively copy every node in the result set, *but*:
 - i. if --grabincludes is specified, only include those nodes that are also in the set created by applying the XPath specified by --grabincludes to the merge event.
 - ii. if --grabexcludes is specified, exclude those nodes that are also in the set created by applying the XPath specified by --grabexcludes to the merge event.
5. All non-matching events in the input files will be output without change.

The output files will be named the same as the inputs, but starting with the prefix "merged-".

A.12. Usage for bxh_eventresp

Usage:

```
bxh_eventresp [opts] eventfiles... outputfile
```

This program takes event files as input, and selects stimulus and response events (given user-specified queries). The responses are then merged into the closest stimulus event within a given time interval from the response.

```
--version
    Print version string and exit.
--optsfromfile <str>
--optsfromfile=<str>
    Program options (i.e. those starting with '--') will come from this file. If this option is specified, then the options in the file will be applied after all command-line options. The options (and their arguments) should be specified one per line, with the leading '--' omitted.
--overwrite
    Overwrite existing output files (otherwise error and exit).
--querylanguage <str>
--querylanguage=<str>
```

The language used for all queries. Valid values are 'XPath' and 'event'. Case is irrelevant. Default is 'XPath'.

```
--stimquery <str>
--stimquery=<str>
    A query string to match stimulus events. This option is required.
--stimfilterquery <str>
--stimfilterquery=<str>
    A query string to filter stimulus events.
--respquery <str>
--respquery=<str>
    A query string to match response events. This option is required.
--respfilterquery <str>
--respfilterquery=<str>
    A query string to filter stimulus events.
--maxresptime <double>
--maxresptime=<double>
    Specifies the longest time interval (in the same units as the onsets in
    the input file) within which a response can be associated with a
    stimulus. A negative value represents infinity (default).
--respdelayname <str>
--respdelayname=<str>
    The name to be used to label the value for response delay (time of
    response minus time of stimulus). Default is not to add this value.
--embeddedrespdelayvalues <str>
--embeddedrespdelayvalues=<str>
    If the actual response delay is embedded within (and relative to) an
    event that is not strictly a response event, this option lists the
    names of the <value> elements (separated by commas) in the
    (pseudo-)response events that would store the response delay. Only one
    value within each event may match this list. This value will be added
    to the default response delay (response event time minus stimulus event
    time) to calculate the actual response time/delay.
--movevalue <str>
--movevalue=<str>
    By default, all values are moved from matched responses to matched
    stimuli. If this option is specified one or more times, only the
    values specified by instances of this option will be moved. Other
    values will be left alone.
--reversemerge
    This option reverses the merging process -- instead of moving response
    event values into matching stimulus events, it will move the matching
    stimulus event's values into the response event. The response delay
    value (if --respdelayname is specified) is also put into the response
    event. Make sure this is what you really want to do!
```

A.13. Usage for bxh_eventstats

```
Can't locate JSON/PP.pm in @INC (@INC contains: /builddir/build/BUILD/RELEASE_1_10_7/
bxh_xcede_tools/utils /usr/local/lib64/perl5 /usr/local/share/perl5 /usr/lib64/perl5/
vendor_perl /usr/share/perl5/vendor_perl /usr/lib64/perl5 /usr/share/perl5 .) at /
builddir/build/BUILD/RELEASE_1_10_7/bxh_xcede_tools/utils/BXHPerlUtils/XMLMetadata.pm
line 3.
BEGIN failed--compilation aborted at /builddir/build/BUILD/RELEASE_1_10_7/
bxh_xcede_tools/utils/BXHPerlUtils/XMLMetadata.pm line 3.
Compilation failed in require at /builddir/build/BUILD/RELEASE_1_10_7/bxh_xcede_tools/
utils/BXHPerlUtils.pm line 13.
BEGIN failed--compilation aborted at /builddir/build/BUILD/RELEASE_1_10_7/
bxh_xcede_tools/utils/BXHPerlUtils.pm line 13.
Compilation failed in require at ../utils/bxh_eventstats line 13.
BEGIN failed--compilation aborted at ../utils/bxh_eventstats line 13.
```

A.14. Usage for bxh_eventstats_standardize

```
Can't locate JSON/PP.pm in @INC (@INC contains: /builddir/build/BUILD/RELEASE_1_10_7/
bxh_xcede_tools/utils /usr/local/lib64/perl5 /usr/local/share/perl5 /usr/lib64/perl5/
vendor_perl /usr/share/perl5/vendor_perl /usr/lib64/perl5 /usr/share/perl5 .) at /
builddir/build/BUILD/RELEASE_1_10_7/bxh_xcede_tools/utils/BXHPerlUtils/XMLMetadata.pm
line 3.
BEGIN failed--compilation aborted at /builddir/build/BUILD/RELEASE_1_10_7/
bxh_xcede_tools/utils/BXHPerlUtils/XMLMetadata.pm line 3.
Compilation failed in require at /builddir/build/BUILD/RELEASE_1_10_7/bxh_xcede_tools/
utils/BXHPerlUtils.pm line 13.
BEGIN failed--compilation aborted at /builddir/build/BUILD/RELEASE_1_10_7/
bxh_xcede_tools/utils/BXHPerlUtils.pm line 13.
Compilation failed in require at ../../utils/bxh_eventstats_standardize line 15.
BEGIN failed--compilation aborted at ../../utils/bxh_eventstats_standardize line 15.
```

A.15. Usage for bxh_mean

Usage:
 bxh_mean [opts] inputs.bxh... output.bxh

This program calculates per-voxel averages across a selected dimension, and produces an output dataset 'collapsed' across that dimension. If --dimension 'dataset' is specified, then corresponding voxels in each input dataset are averaged to create an output dataset of the same dimensionality; in this case, all of the dimensions in all input datasets must match. If multiple input datasets are provided and --dimension 'dataset' is not specified, then they are concatenated along the last (slowest-moving) dimension; i.e. if one specifies an XYZT 64x64x27x120 time series and an XYZT 64x64x27x130 time series as inputs, they will be considered together as a single 64x64x27x250 time series. In this case, all dimensions except the last dimension must match in all data sets.

```
--version
    Print version string and exit.
--stddev <str>
--stddev=<str>
    Calculate standard deviation too, and put the output in this file.
--sumonly
    Calculate only the sum of the data, and don't divide by the number of
    inputs. This option can not be used with --stddev.
--dimension <str>
--dimension=<str>
    Select the dimension over which to average. The dimension must be one
    that exists in the input dataset, or must be 'dataset'. Default is the
    last (slowest-moving) dimension.
--outtype <str>
--outtype=<str>
    The output will be of this type. Valid types are: float64, float32,
    uint32, int32, uint16, int16, uint8, int8. Note: using this option may
    result in overflow/underflow or precision errors if the output type can
    not represent the output appropriately. Default is float64 if either
    of the inputs are float64, or float32 otherwise.
```

A.16. Usage for bxh_tfilter

Usage:

```
bxh_tfilter [opts] input.bxh output.bxh
```

This program runs, on a 4-D data set, a Chebyshev filter across each voxel's fourth dimension (e.g. time course) and writes the results to output.bxh.

```
--version
    Print version string and exit.
--overwrite
    Overwrite existing output files (otherwise error and exit).
--filtertype <str>
--filtertype=<str>
    This required option chooses the filter type. Valid choices are
    'lowpass', 'highpass', 'bandpass', or 'bandstop'. Each filter is
    parameterized by one or more instances of --period. 'lowpass' or
    'highpass' require one --period option, specifying the stop or start
    frequency respectively. 'bandpass' or 'bandstop' require two --period
    options, specifying the start and stop frequencies, in any order
    (larger period/smaller frequency is assumed to be start frequency for
    'bandpass' and stop frequency for 'bandstop').
--period <double>
--period=<double>
    This option specifies the frequency parameters for the filter in terms
    of the period (i.e. 1/frequency) in seconds per cycle. May be
    specified once for 'lowpass' and 'highpass' filter types, twice for
    'bandpass' and 'bandstop' filter types, and must be greater than 0.
--ripple <double>
--ripple=<double>
    This option specifies the percent ripple for the Chebyshev filter. If
    0 [zero], which is the default, then the filter is a Butterworth
    filter.
--order <uint>
--order=<uint>
    Order of the filter. Default is 6.
--forcetr <double>
--forcetr=<double>
    If specified, this value (in seconds) will replace the TR specified in
    the input image file, if any.
--keepdc
    Keep DC component (mean signal). Has no effect for lowpass and
    bandpass filter types (which already keep the DC component).
```

A.17. Usage for bxh_ttest

Usage:

```
bxh_ttest [opts] avgl.bxh std1.bxh n1.bxh avg2.bxh std2.bxh n2.bxh out_tfile
```

This program computes a per-voxel t-statistic between two datasets given their 3-D or 4-D mean, standard deviation, and n images. Output (in out_tfile) is a data set, with the same dimensions as the input, storing the t-statistic.

```
--version
    Print version string and exit.
--optsfromfile <str>
--optsfromfile=<str>
    Program options (i.e. those starting with '--') will come from this
    file. If this option is specified, then the options in the file will
    be applied after all command-line options. The options (and their
    arguments) should be specified one per line, with the leading '--'
    omitted.
--overwrite
    Overwrite existing output files (otherwise error and exit).
--timeselect <str>
```

```
--timeselect=<str>
    Comma-separated list of timepoints to use (first timepoint is 0). Any
    timepoint can be a contiguous range, specified as two numbers separated
    by a colon, i.e. 'START:END'. An empty END implies the last timepoint.
    The default step of 1 (one) in ranges can be changed using
    'START:STEP:END', which is equivalent to
    'START,START+STEP,START+(2*STEP),...,END'. Default is all timepoints
    (:).
--xselect <str>
--xselect=<str>
    Just like timeselect, but for the 'x' dimension.
--yselect <str>
--yselect=<str>
    Just like timeselect, but for the 'y' dimension.
--zselect <str>
--zselect=<str>
    Just like timeselect, but for the 'z' dimension.
--maskfile <str>
--maskfile=<str>
    Use this 3-D mask (should be an XML file) before doing calculations.
```

A.18. Usage for bxhabsorb

```
Usage:
  bxhabsorb [ --fromtype type ] [ type-specific-opts... ] inputfiles...
  bxhoutoutputfile
  bxhabsorb [ --fromtype type ] [ --inputsfromfile inputlistfile ] [
  type-specific-opts...] bxhoutoutputfile
```

This program creates a BXH or XCEDE wrapper for the specified input images. If the `--fromtype` option is not specified, it attempts to auto-detect the format of the input image files. If it cannot auto-detect the format, it will exit with an error.

```
--fromtype <str>
--fromtype=<str>
    Type of the input data (pfile, signa5, signafive, iowa-signa5,
    iowa-signafive, ximg, analyze, afni, nrrd, dicom). If this option is
    not specified, attempt to autodetect format of inputfiles.
--inputsfromfile <str>
--inputsfromfile=<str>
    Read list of input files from this file.
--version
    Print version string and exit.
--hintsize x <size_t>
--hintsize x=<size_t>
--hintsize y <size_t>
--hintsize y=<size_t>
--hintsize z <size_t>
--hintsize z=<size_t>
--hintsize t <size_t>
--hintsize t=<size_t>
--hintorigin x <double>
--hintorigin x=<double>
--hintorigin y <double>
--hintorigin y=<double>
--hintorigin z <double>
--hintorigin z=<double>
--hintorigin t <double>
--hintorigin t=<double>
--hintspacing x <double>
--hintspacing x=<double>
--hintspacing y <double>
--hintspacing y=<double>
```

```
--hintspacingz <double>
--hintspacingz=<double>
--hintspacinggt <double>
--hintspacinggt=<double>
--hintgapx <double>
--hintgapx=<double>
--hintgapy <double>
--hintgapy=<double>
--hintgapz <double>
--hintgapz=<double>
--hintgapt <double>
--hintgapt=<double>
    These options will provide a hint to the program of the 'size',
    'origin', 'spacing', or 'gap' of the specified dimension. Some image
    types will not use all these values. In particular, sizex and sizey
    are assumed correct in most image headers, but they, as well as sizez
    and sizet options may be useful with image type 'pfile'. Origin and
    spacing hints will be used by most image types.
--forceorientation <str>
--forceorientation=<str>
    This option will force the labeled orientation of the image to match
    the given three letter orientation code. Each letter must come from
    the following groups in any order: R(ight) or L(eft); A(nterior) or
    P(osterior); S(uperior) or I(nferior). Only one letter from each group
    is allowed.
--xcede
    Write XCEDE-style XML files.
--xcede2
    Write XCEDE 2-style XML files.
```

PFILE USAGE

```
bxhabsorb --fromtype pfile [opts] [pfilehdr imagedata1...] output.bxh
```

PFILE OPTIONS

```
--forceversion <float>
--forceversion=<float>
    Force version of P-file to be interpreted as this number.
--msbfirst
    Indicates that data is big-endian (default: little-endian).
--dimorder <str>
--dimorder=<str>
    Comma-separated names of dimensions from fastest-moving to
    slowest-moving (default: x,y,z,t).
--elementype <str>
--elementype=<str>
    Provide element type of image data (one of int8, uint8, int16
    [default], uint16, int32, uint32, float32, or float64).
--usemrorigin
    This option extracts the origin from the tlhc_[RAS] fields in the MR
    structure. This is the default.
--useslicetableorigin
    The origin coordinates are extracted from the slice table at the end of
    the P-file header.
```

SIGNA 5.X USAGE

```
bxhabsorb --fromtype signa5 [opts] [signa5files...] output.bxh
```

```
bxhabsorb --fromtype signafive [opts] [signa5files...] output.bxh
```

SIGNA 5.X OPTIONS

```
--dimzsize <size_t>
--dimzsize=<size_t>
    Specifies the size of the z dimension (i.e. number of slices per
    timepoint). Default is to use the number of input files. Equivalent
    to (and overrides) --hintsizez.
--dimtsize <size_t>
--dimtsize=<size_t>
    Specifies the size of the t dimension (i.e. number of timepoints).
    Default is number of input files divided by number of slices per
    timepoint (as specified by --dimzsize). Equivalent to (and overrides)
```

```

--hintsize t.

IOWA SIGNA 5.X USAGE
  bxhabsorb --fromtype iowa-signa5 imagedir output.bxh
  bxhabsorb --fromtype iowa-signafive imagedir output.bxh
  NOTE: the bxhabsorb option --inputsfromfile is not available for the
  iowa-signa5 format
  imagedir is a directory containing I.* images

XIMG USAGE
  bxhabsorb --fromtype ximg [opts] [ximgfiles...] output.bxh
XIMG OPTIONS
  --dimzsize <size_t>
  --dimzsize=<size_t>
    Specifies the size of the z dimension (i.e. number of slices per
    timepoint). Default is to use the number of input files. Equivalent
    to (and overrides) --hintsize z.
  --dimtsize <size_t>
  --dimtsize=<size_t>
    Specifies the size of the t dimension (i.e. number of timepoints).
    Default is number of input files divided by number of slices per
    timepoint (as specified by --dimzsize). Equivalent to (and overrides)
    --hintsize t.

ANALYZE/SPM USAGE
  bxhabsorb --fromtype analyze [opts] [analyzefiles...] output.bxh
ANALYZE/SPM OPTIONS:
  --orientation <str>
  --orientation=<str>
    Orientation of image, letters indication which way the X, Y, and Z
    dimensions (in that order) are pointing (e.g. LPS, IRP). Default is
    RAS (i.e. orientation used by SPM), or that specified in accompanying
    SPM .mat files. This option overrides all info in SPM .mat files
    and/or the Analyze 'orient' field (if --strictanalyze is specified).
    Equivalent to (and overrides) --forceorientation.
  --strictanalyze
    Don't use SPM .mat files and use Analyze convention for orientation.
    The 'orient' field in the analyze header is parsed, and the default
    case (i.e. it is zero) means 'LAS'.
  --avwbyteorder <str>
  --avwbyteorder=<str>
    Specify byte order for AVW files (which don't store this info). This
    field should be 'l' for little-endian or 'b' for big-endian.

AFNI USAGE
  bxhabsorb --fromtype afni [opts] afnifile.HEAD output.bxh

NRRD USAGE
  bxhabsorb --fromtype nrrd [opts] file.nrrd output.bxh
  bxhabsorb --fromtype nrrd [opts] file.nhdr output.bxh

DICOM USAGE
  bxhabsorb --fromtype dicom [opts] [dicomfiles...] output.bxh
general options:
  --debug
  -d
    debug mode, print debug information
input options:
  --force-concat
    If the input images have different orientation, Study UID, Series UID,
    ImageType, etc., then this option nevertheless forces them to be
    concatenated into the same volume. (They would otherwise be
    encapsulated within separate XML files.) This option may result in XML
    files that do not correctly describe the DICOM data -- use only if you
    know what you're doing!
  --filename-sort

```

```

    This program normally sorts input files by various fields in the DICOM
    headers. This option forces a sort by filename only. This can be
    useful in the case that the fields are unreliable.
--no-sort
    This program normally sorts input files by various fields in the DICOM
    headers. This option disables sorting and relies on the order in which
    files are provided on the command line. This can be useful in the case
    that the fields are unreliable.
    input file format:
--search-for-others
-s
    search for matching files in the same directory
--read-dataset
-f
    read data set without file meta information
input transfer syntax (only with --read-dataset):
--read-xfer-auto
-t
    use TS recognition (default)
--read-xfer-little
-te
    read with explicit VR little endian TS
--read-xfer-big
-tb
    read with explicit VR big endian TS
--read-xfer-implicit
-ti
    read with implicit VR little endian TS
output options:
converting:
--load-short
-M
    do not load very long values (e.g. pixel data)
error handling:
--ignore-errors
-E
    attempt to convert even if file is damaged

MGH/MGZ USAGE
    bxhabsorb --fromtype mgh [opts] file.mgh output.bxh
    bxhabsorb --fromtype mgh [opts] file.mgz output.bxh

BXH/XCEDE (as input) USAGE
    bxhabsorb --fromtype bxh [opts] bxhfile outputfile
    bxhabsorb --fromtype xcede [opts] xcedefile outputfile

```

A.19. Usage for bxhreorient

```

Usage:
    bxhreorient [options] inputfile [ outputfile [datafileout] ]

```

This program reorients the image data given by the input BXH or XCEDE file to an orientation specified by the user using the `--orientation` option. It is assumed that the orientation vectors in the BXH/XCEDE file are correct with respect to the image data. `outputfile` is required if not using `--inplace` option. Output is also a BXH or XCEDE file, pointing to an image data file (named by `datafileout` if specified).

```

--version
    Print version string and exit.
--overwrite
    Overwrite output files if they exist.
--orientation <str>

```

```
--orientation=<str>
    This option specifies the new orientation by R/L A/P S/I letters,
    upper- or lower-case, in X,Y,Z order, where R means that dimension
    starts on the left and goes TO THE RIGHT, A means the dimension goes
    from posterior TO ANTERIOR, etc. For example, IPR means X goes S->I, Y
    goes A->P, and Z goes L->R. Default is RAS (neurological axial, as
    used by SPM).
```

A.20. Usage for bxhselect

Usage:
 bxhselect [options] inputfile outputfile

This program copies a subset of the input image data based on the various selection options. The selected subset is written to the output file.

```
--version
    Print version string and exit.
--timeselect <str>
--timeselect=<str>
    Comma-separated list of timepoints to use (first timepoint is 0). Any
    timepoint can be a contiguous range, specified as two numbers separated
    by a colon, i.e. 'START:END'. An empty END implies the last timepoint.
    The default step of 1 (one) in ranges can be changed using
    'START:STEP:END', which is equivalent to
    'START,START+STEP,START+(2*STEP),...,END'.
--xselect <str>
--xselect=<str>
    Just like timeselect, but for the 'x' dimension.
--yselect <str>
--yselect=<str>
    Just like timeselect, but for the 'y' dimension.
--zselect <str>
--zselect=<str>
    Just like timeselect, but for the 'z' dimension.
--overwrite
    Overwrite output files if they exist.
```

A.21. Usage for bxhsetorient

Usage:
 bxhsetorient [options] orient inputfile [outputfile]

This program sets the orientation vectors in the BXH or XCEDE file. NOTE: this program does not reorient or otherwise touch the image data itself. This program is useful to fix incorrect or missing orientation vectors in a BXH/XCEDE file. bxhfileout is required if not using --inplace option (and vice-versa). 'orient' specifies the new orientation by R/L A/P S/I letters, upper- or lower-case, in X,Y,Z order, where R means that dimension starts on the left and goes TO THE RIGHT, A means the dimension goes from posterior TO ANTERIOR, etc. For example, IPR means X goes S->I, Y goes A->P, and Z goes L->R.

```
--version
    Print version string and exit.
--inplace
    Do the reorientation in-place, overwriting the original BXH file.
```

A.22. Usage for dicom2bxh

```

Usage:
  dicom2bxh [opts] [dicomfiles...] output.bxh

This program creates an XML wrapper for DICOM images.

general options:
  --debug
  -d
      debug mode, print debug information
input options:
  --force-concat
      If the input images have different orientation, Study UID, Series UID,
      ImageType, etc., then this option nevertheless forces them to be
      concatenated into the same volume. (They would otherwise be
      encapsulated within separate XML files.) This option may result in XML
      files that do not correctly describe the DICOM data -- use only if you
      know what you're doing!
  --filename-sort
      This program normally sorts input files by various fields in the DICOM
      headers. This option forces a sort by filename only. This can be
      useful in the case that the fields are unreliable.
  --no-sort
      This program normally sorts input files by various fields in the DICOM
      headers. This option disables sorting and relies on the order in which
      files are provided on the command line. This can be useful in the case
      that the fields are unreliable.
  input file format:
  --search-for-others
  -s
      search for matching files in the same directory
  --read-dataset
  -f
      read data set without file meta information
input transfer syntax (only with --read-dataset):
  --read-xfer-auto
  -t
      use TS recognition (default)
  --read-xfer-little
  -te
      read with explicit VR little endian TS
  --read-xfer-big
  -tb
      read with explicit VR big endian TS
  --read-xfer-implicit
  -ti
      read with implicit VR little endian TS
output options:
  converting:
  --load-short
  -M
      do not load very long values (e.g. pixel data)
error handling:
  --ignore-errors
  -E
      attempt to convert even if file is damaged
additional options:
  --inputsfromfile <str>
  --inputsfromfile=<str>
      Read list of input files from this file.
  --version
      Print version string and exit.
  --hintsize <size_t>

```

```

--hintsize=<size_t>
--hintsizey <size_t>
--hintsizey=<size_t>
--hintsizez <size_t>
--hintsizez=<size_t>
--hintsizez <size_t>
--hintsizez=<size_t>
--hintsizez <size_t>
--hintsizez=<size_t>
--hintoriginx <double>
--hintoriginx=<double>
--hintoriginy <double>
--hintoriginy=<double>
--hintoriginz <double>
--hintoriginz=<double>
--hintorigint <double>
--hintorigint=<double>
--hintspacingx <double>
--hintspacingx=<double>
--hintspacingy <double>
--hintspacingy=<double>
--hintspacingz <double>
--hintspacingz=<double>
--hintspacingz <double>
--hintspacingz=<double>
--hintspacingt <double>
--hintspacingt=<double>
--hintgapx <double>
--hintgapx=<double>
--hintgapy <double>
--hintgapy=<double>
--hintgapz <double>
--hintgapz=<double>
--hintgapt <double>
--hintgapt=<double>
    These options will provide a hint to the program of the 'size',
    'origin', 'spacing', or 'gap' of the specified dimension. Some image
    types will not use all these values. In particular, size and sizey
    are assumed correct in most image headers, but they, as well as sizez
    and sizet options may be useful with image type 'pfile'. Origin and
    spacing hints will be used by most image types.
--forceorientation <str>
--forceorientation=<str>
    This option will force the labeled orientation of the image to match
    the given three letter orientation code. Each letter must come from
    the following groups in any order: R(ight) or L(ef); A(nterior) or
    P(osterior); S(uperior) or I(nferior). Only one letter from each group
    is allowed.
--xcde
    Write XCEDE-style XML files.
--xcde2
    Write XCEDE 2-style XML files.

```

A.23. Usage for dumpheader

Usage:
 dumpheader inputfile

This program prints a simplistic summary of the BXH or XCEDE file given as input.

A.24. Usage for eprime2xml

Usage:


```
eprime2xml instructions.txt eprime.txt [outputevents.xml]
```

eprime2xml takes an E-Prime output file as exported as text from the E-Prime software and an instruction file, and creates an XML event file. The instruction file indicates which columns in the E-Prime file are of interest and what they should map to in the output event file. If the output file is not specified, the event data is written to standard output.

Options:

```
--xcede2
    Write in XCEDE2 format.
--xcede2dataid=ID
    ID for the XCEDE 2 data element (default: auto-generated based
    on hostname, process ID, and current time)
--extracttable
    Instead of creating XML as output, output as tabular text.
    This is a no-op for most formats, and is really only useful for
    converting E-Prime "recovery" logs into a tabular form.
--columnnames
    If this options is specified, only the columns of the table are
    printed (one per line) and the program exits.
--subtractonset SECS
    This option subtracts SECS from all onset times (default is 0).
    This is in addition to any other normalization that may occur
    (see use of 'firstmritime' below).
--colsep SEPARATOR
    This option specifies the column separator (default is tab).
```

The instruction file language is defined as follows:

```
COMMAND VALUESPEC [VALUESPEC...]
```

where COMMAND can be event, param, or block. VALUESPEC has one of the following formats:

```
[OUTVALUENAME=]COLUMNNAME[:UNITS]
OUTVALUENAME=@TEXT[:UNITS]
```

Each VALUESPEC defines values that should be passed through to the corresponding event, param, or block. In the first alternative listed above, the value comes from a column in the input file (optionally renamed to OUTVALUENAME) and in the second alternatives, the value is directly specified preceded by a '@' character. If OUTVALUENAME= is missing, then the VALUESPEC is equivalent to:

```
COLUMNNAME=COLUMNNAME[:UNITS]
```

OUTVALUENAME may not contain the equals sign ('='), at sign ('@'), quotes, or whitespace.

Either COLUMNNAME or TEXT may contain quoted substrings to protect special characters like colon (':'), equals sign ('='), at sign ('@'), or whitespace; otherwise these special characters are prohibited. A single quote will protect all characters until the next single quote, and likewise for double quote. The following examples show equivalent VALUESPECs:

```
description=DESC
description=D'E' "SC"

onset="Onset Time":secs
onset=Onset' 'Time:secs
onset=Onse't T'ime:secs
```

Unquoted spaces separate VALUESPECs.

```
-----
'event' command:
```

 Each 'event' command creates a class of events in the output event file, where the contents of the event are specified by the VALUESPECs. In general, for each matching row (more later), it creates an event with the following contents:

```
<event type="$type" units="$units">
  <onset>$onset</onset>
  <duration>$duration</duration>
  <name>$name</name>
  <description>$description</description>
  <value name="OUTVALUENAME1" units="UNITS1">VALUE1</value>
  <value name="OUTVALUENAME2" units="UNITS2">VALUE2</value>
  ...
</event>
```

VALUESPECs whose OUTVALUENAMES start with a dollar sign (\$) are "magic", and are interpreted in a value-specific way. VALUESPECs whose OUTVALUENAMES start with a percent sign (%) are explicitly non-magic. Any OUTVALUENAME not starting with a % or \$ is assumed to have an implicit % unless it matches a list of pre-defined magic values (below), in which case an implicit \$ is assumed.

Pre-defined magic values '\$type', '\$units', '\$onset', '\$duration', '\$name', and '\$description' are put in the appropriate child element or attribute of <event> (shown above). Only the '\$onset' VALUESPEC is required. Default value for '\$duration' is zero. All non-magic values are placed in <value> elements.

The pre-defined magic value '\$DURUNTIL' indicates that any row in the input used to create an event will have an ending time specified by the value of column COLUMNNAME in the current row. Likewise, the value '\$DURUNTILNEXTROW' does the same thing, but grabs the value from the next row. These are used to calculate the duration of this event. This may be specified more than once, and the first non-NULL column will be used. This option is used when a row does not have a duration column, and it must be calculated based on times in this or the subsequent row.

By default, only those rows whose '\$onset' column is non-empty and non-NULL will be processed as events. Certain magic OUTVALUENAMES further restrict the rows that are used for this event command. '\$MATCH' and '\$MATCHNONZERO' specify a column whose values indicate whether that row should be selected -- for '\$MATCH', the values must be non-empty and non-'NULL'; for '\$MATCHNONZERO', the values must also be non-zero. With '\$MATCHEQUAL', one specifies both a column and an actual value to match -- for the '\$MATCHEQUAL' value name (and only the '\$MATCHEQUAL' value name) the VALUESPEC syntax is extended in the following way:

```
$MATCHEQUAL=COLUMNNAME@MATCHVALUE
```

where COLUMNNAME and MATCHVALUE are the two relevant parameters.

 'block' command:

The block command has the same usage as the event command. The same magic values apply to block commands as event commands. An '\$onset' value is again required, and '\$duration' is optional (assumed to be zero [0] if missing).

 'param' command:

Each param command specifies a list of columns that should be

put in the <params> section of the event file. These represent parameters that are constant (or default) throughout the events file. Each VALUESPEC represents one item to put in the <params> element as such:

```
<params>
...
<value name="OUTVALUENAME1" units="UNITS1">VALUE1</value>
<value name="OUTVALUENAME2" units="UNITS2">VALUE2</value>
...
</params>
```

Only the first non-empty, non-NULL field in the column specified by a 'param' will be used. Be aware of this if this column does not have the same value in every row.

There is one magic OUTVALUENAME (maybe more later) '\$firstmritime', which will generate the following element:

```
<params>
  <firstmritime>0</firstmritime>
</params>
```

If '\$firstmritime' is specified, it (and all '\$onset' VALUESPECs) must have UNITS specified. All '\$onset' columns are normalized by this value, so their units and '\$firstmritime' units must match.

A.25. Usage for eventstable2xml

Usage:

```
eventstable2xml instructions.txt inputevents.txt [outputevents.xml]
```

eventstable2xml takes a text tabular events file and an instruction file, and creates an XML events file. The instruction file indicates which columns in the original events file are of interest and what they should map to in the output event file. If the output file is not specified, the event data is written to standard output.

Options:

```
--xcede2
    Write in XCEDE2 format.
--xcede2dataid=ID
    ID for the XCEDE 2 data element (default: auto-generated based
    on hostname, process ID, and current time)
--extracttable
    Instead of creating XML as output, output as tabular text.
    This is a no-op for most formats, and is really only useful for
    converting E-Prime "recovery" logs into a tabular form.
--columnnames
    If this options is specified, only the columns of the table are
    printed (one per line) and the program exits.
--subtractonset SECS
    This option subtracts SECS from all onset times (default is 0).
    This is in addition to any other normalization that may occur
    (see use of 'firstmritime' below).
--colsep SEPARATOR
    This option specifies the column separator (default is tab).
```

The instruction file language is defined as follows:

```
COMMAND VALUESPEC [VALUESPEC...]
```

where COMMAND can be event, param, or block. VALUESPEC has one of the following formats:

```
[OUTVALUENAME=]COLUMNNAME[:UNITS]
OUTVALUENAME=@TEXT[:UNITS]
```

Each VALUESPEC defines values that should be passed through to the corresponding event, param, or block. In the first alternative listed above, the value comes from a column in the input file (optionally renamed to OUTVALUENAME) and in the second alternatives, the value is directly specified preceded by a '@' character. If OUTVALUENAME= is missing, then the VALUESPEC is equivalent to:

```
COLUMNNAME=COLUMNNAME[:UNITS]
```

OUTVALUENAME may not contain the equals sign ('='), at sign ('@'), quotes, or whitespace.

Either COLUMNNAME or TEXT may contain quoted substrings to protect special characters like colon (':'), equals sign ('='), at sign ('@'), or whitespace; otherwise these special characters are prohibited. A single quote will protect all characters until the next single quote, and likewise for double quote. The following examples show equivalent VALUESPECs:

```
description=DESC
description=D'E"SC"

onset="Onset Time":secs
onset=Onset' 'Time:secs
onset=Onse't T'ime:secs
```

Unquoted spaces separate VALUESPECs.

```
-----
'event' command:
-----
```

Each 'event' command creates a class of events in the output event file, where the contents of the event are specified by the VALUESPECs. In general, for each matching row (more later), it creates an event with the following contents:

```
<event type="$type" units="$units">
  <onset>$onset</onset>
  <duration>$duration</duration>
  <name>$name</name>
  <description>$description</description>
  <value name="OUTVALUENAME1" units="UNITS1">VALUE1</value>
  <value name="OUTVALUENAME2" units="UNITS2">VALUE2</value>
  ...
</event>
```

VALUESPECs whose OUTVALUENAMES start with a dollar sign (\$) are "magic", and are interpreted in a value-specific way. VALUESPECs whose OUTVALUENAMES start with a percent sign (%) are explicitly non-magic. Any OUTVALUENAME not starting with a % or \$ is assumed to have an implicit % unless it matches a list of pre-defined magic values (below), in which case an implicit \$ is assumed.

Pre-defined magic values '\$type', '\$units', '\$onset', '\$duration', '\$name', and '\$description' are put in the appropriate child element or attribute of <event> (shown above). Only the '\$onset' VALUESPEC is required. Default value for '\$duration' is zero. All non-magic values are placed in <value> elements.

The pre-defined magic value '\$DURUNTIL' indicates that any row in the input used to create an event will have an ending time specified by the value of column COLUMNNAME in the current row. Likewise, the value '\$DURUNTILNEXTROW' does the same thing, but grabs the value from

the next row. These are used to calculate the duration of this event. This may be specified more than once, and the first non-NULL column will be used. This option is used when a row does not have a duration column, and it must be calculated based on times in this or the subsequent row.

By default, only those rows whose '\$onset' column is non-empty and non-NULL will be processed as events. Certain magic OUTVALUENAMES further restrict the rows that are used for this event command. '\$MATCH' and '\$MATCHNONZERO' specify a column whose values indicate whether that row should be selected -- for '\$MATCH', the values must be non-empty and non-'NULL'; for '\$MATCHNONZERO', the values must also be non-zero. With '\$MATCHEQUAL', one specifies both a column and an actual value to match -- for the '\$MATCHEQUAL' value name (and only the '\$MATCHEQUAL' value name) the VALUESPEC syntax is extended in the following way:

```
$MATCHEQUAL=COLUMNNAME@MATCHVALUE
```

where COLUMNNAME and MATCHVALUE are the two relevant parameters.

```
-----
'block' command:
-----
```

The block command has the same usage as the event command. The same magic values apply to block commands as event commands. An '\$onset' value is again required, and '\$duration' is optional (assumed to be zero [0] if missing).

```
-----
'param' command:
-----
```

Each param command specifies a list of columns that should be put in the <params> section of the event file. These represent parameters that are constant (or default) throughout the events file. Each VALUESPEC represents one item to put in the <params> element as such:

```
<params>
...
<value name="OUTVALUENAME1" units="UNITS1">VALUE1</value>
<value name="OUTVALUENAME2" units="UNITS2">VALUE2</value>
...
</params>
```

Only the first non-empty, non-NULL field in the column specified by a 'param' will be used. Be aware of this if this column does not have the same value in every row.

There is one magic OUTVALUENAME (maybe more later) '\$firstmritime', which will generate the following element:

```
<params>
<firstmritime>0</firstmritime>
</params>
```

If '\$firstmritime' is specified, it (and all '\$onset' VALUESPECs) must have UNITS specified. All '\$onset' columns are normalized by this value, so their units and '\$firstmritime' units must match.

A.26. Usage for extractimagedata

Usage:

```
extractimagedata [opts] xmlfile outputfile
```

This program extracts the image data pointed to by the input BXH or XCEDE file and writes it to outputfile.

```
--version
    Print version string and exit.
--msbfirst
    Extract data as big-endian (default: little-endian).
```

A.27. Usage for extractxyztdata

```
Usage:
  extractxyztdata [opts] xmlfile outputfile
```

This program extracts the image data pointed to by the input BXH or XCEDE file and writes it to outputfile. The data is reordered so that the dimensions labeled 'x', 'y', 'z', and 't' are in that order.

```
--version
    Print version string and exit.
--msbfirst
    Extract data as big-endian (default: little-endian).
```

A.28. Usage for ffile2bxh

```
Usage:
  ffile2bxh [ --dimorder "x,y,z,t" ] ffile [datafile1...] outputfile
```

This program takes a Stanford F-file and creates a BXH or XCEDE header using the metadata in the F-file, and points to the image data in the given datafiles.

```
--dimorder specifies the comma-separated names of the dimensions in the
datafiles(s) in order from fastest-moving to slowest-moving
Default is "x,y,z,t".
```

```
--xcede produces an XCEDE file as output.
```

A.29. Usage for fmriqa_count

```
Usage:
  fmriqa_count inputfile
```

This program outputs histograms or counts of voxels in a BXH- or XCEDE-wrapped dataset that match the given conditions. Output can be per-slice, per-volume, or for entire dataset (see --granularity). Histogram output requires the --histogram option. If histogram output is not chosen, output is as if there were one histogram 'bucket'. Conditions are specified as command-line options, described below. Default is to 'and' all conditions (but see --aggregate). Default condition for 'and' aggregate is to match all voxels. Default condition for 'or' aggregate is to match no voxels. Thus, with no options, this program prints out the number of voxels in the data.

```
--version
```

```

    Print version string and exit.
--granularity <str>
--granularity=<str>
    Print counts at this granularity. Acceptable values are 'timeseries'
    (default), 'volume', 'slice', and 'voxel'.
--aggregate <str>
--aggregate=<str>
    Conditions are aggregated by this operator, either 'and' (default) or
    'or'.
--gt <double>
--gt=<double>
    Match those voxels greater than this value.
--ge <double>
--ge=<double>
    Match those voxels greater than or equal to this value.
--lt <double>
--lt=<double>
    Match those voxels less than this value.
--le <double>
--le=<double>
    Match those voxels less than or equal to this value.
--timeselect <str>
--timeselect=<str>
    Match only those timepoints in this comma-separated list of timepoints
    (first timepoint is 0). Any timepoint can be a contiguous range,
    specified as two numbers separated by a colon, i.e. 'START:END'. An
    empty END implies the last timepoint. The default step of 1 (one) in
    ranges can be changed using 'START:STEP:END', which is equivalent to
    'START,START+STEP,START+(2*STEP),...,END'.
--xselect <str>
--xselect=<str>
    Just like timeselect, but for the 'x' coordinate.
--yselect <str>
--yselect=<str>
    Just like timeselect, but for the 'y' coordinate.
--zselect <str>
--zselect=<str>
    Just like timeselect, but for the 'z' coordinate.
--histogram
    Specifies that output should be histogram. See --histobuckets to
    specify number of buckets, --histobucketwidth to specify width of
    buckets, or --histobounds to specify bucket boundaries.
--histobuckets <int>
--histobuckets=<int>
    Valid only with --histogram option. Constructs this many evenly-spaced
    histogram buckets. This option is incompatible with --histobounds or
    --histobucketwidth.
--histobounds <str>
--histobounds=<str>
    Valid only with --histogram option. By default, histogram bucket
    boundaries are in multiples of standard deviations. This option
    specifies alternate boundaries for (N + 1) buckets as a space-separated
    list of N floating point numbers. For example, --histobounds "0.0 5.0
    10.0" will separate voxels with values -infinity < x < 0.0, 0.0 <= x <
    5.0, 5.0 <= x < 10.0, and 10.0 <= x < infinity. This option is
    incompatible with --histobucketwidth or --histobucketsize.
--histobucketwidth <double>
--histobucketwidth=<double>
    Valid only with --histogram option. Constructs evenly-spaced histogram
    buckets with this width. This option is incompatible with
    --histobounds or --histobuckets.

```

A.30. Usage for fmriqa_generate.pl

```
Can't locate BXHPerlUtils.pm in @INC (@INC contains: /builddir/build/BUILD/
RELEASE_1_10_7/bxh_xcede_tools/fmriqa /usr/local/lib64/perl5 /usr/local/share/perl5 /
usr/lib64/perl5/vendor_perl /usr/share/perl5/vendor_perl /usr/lib64/perl5 /usr/share/
perl5 .) at ../../fmriqa/fmriqa_generate.pl line 39.
BEGIN failed--compilation aborted at ../../fmriqa/fmriqa_generate.pl line 39.
```

A.31. Usage for fmriqa_minmax

```
Usage:
  fmriqa_minmax xmlfile(s)...
```

This program merely computes the minimum and maximum values in the input
BXH- or XCEDE-wrapped dataset and writes them to standard output.

A.32. Usage for fmriqa_oediff

```
Usage:
  fmriqa_oediff [opts] xmlfile outputfile
```

Given a 4-D BXH- or XCEDE-wrapped time series, this program calculates the
cumulative difference between the even images (where the first selected image
is 0) and the odd images. The input file must be BXH or XCEDE file, and the
output is a 3-D image in the same format, written to outputfile.

```
--version
    Print version string and exit.
--timeselect <str>
--timeselect=<str>
    Comma-separated list of timepoints to use (first timepoint is 0). Any
    timepoint can be a contiguous range, specified as two numbers separated
    by a colon, i.e. 'START:END'. An empty END implies the last timepoint.
    The default step of 1 (one) in ranges can be changed using
    'START:STEP:END', which is equivalent to
    'START,START+STEP,START+(2*STEP),...,END'.
--xselect <str>
--xselect=<str>
    Just like timeselect, but for the 'x' dimension.
--yselect <str>
--yselect=<str>
    Just like timeselect, but for the 'y' dimension.
--zselect <str>
--zselect=<str>
    Just like timeselect, but for the 'z' dimension.
```

A.33. Usage for fmriqa_phantomqa.pl

```
Can't locate BXHPerlUtils.pm in @INC (@INC contains: /builddir/build/BUILD/
RELEASE_1_10_7/bxh_xcede_tools/fmriqa /usr/local/lib64/perl5 /usr/local/share/perl5 /
usr/lib64/perl5/vendor_perl /usr/share/perl5/vendor_perl /usr/lib64/perl5 /usr/share/
perl5 .) at ../../fmriqa/fmriqa_phantomqa.pl line 32.
BEGIN failed--compilation aborted at ../../fmriqa/fmriqa_phantomqa.pl line 32.
```


A.34. Usage for fmriqa_phantomqa

Usage:

```
fmriqa_phantomqa [opts] xmlfile [outputbase]
```

This program is usually called by `fmriqa_phantomqa.pl`, and is not likely to be useful to users on its own. This program takes a 4-D BXH- or XCEDE- wrapped dataset and calculates and writes various QA measures, designed for fMRI images of the BIRN calibration phantom.

```
--version
    Print version string and exit.
--timeselect <str>
--timeselect=<str>
    Comma-separated list of timepoints to use (first timepoint is 0). Any
    timepoint can be a contiguous range, specified as two numbers separated
    by a colon, i.e. 'START:END'. An empty END implies the last timepoint.
    The default step of 1 (one) in ranges can be changed using
    'START:STEP:END', which is equivalent to
    'START,START+STEP,START+(2*STEP),...,END'. Default is to ignore first
    2 timepoints (2:), or 3 if the total number of timepoints is odd.
--zselect <str>
--zselect=<str>
    Chooses the slice number on which to compute the statistics. Must be a
    single unsigned integer within the range 0 <= x <= (numslices-1).
    Default is middle slice.
--summaryonly
    Don't generate ave, nave, std images.
--nofluct
    Don't run fluctuation analysis.
--noroi
    Don't run ROI-based analysis.
--roisize <uint>
--roisize=<uint>
    Override the default ROI size of 30x30 (for 128x128 slices) or 15x15
    (for everything else). Specify the length of the edge of the ROI box
    in voxels.
--maskfile <str>
--maskfile=<str>
    Use this mask (should be an XML file) instead of 30x30 (for 128x128
    slices) or 15x15 (or whatever is specified by --roisize). If 2-D, must
    match slice dimensions of input data. If 3-D, all three spatial
    dimensions must match (but only slice specified in zselect will be
    used).
--forcetr <double>
--forcetr=<double>
    If specified, this value (in seconds) will replace the TR specified in
    the input image file, if any.
```

A.35. Usage for fmriqa_spikiness

Usage:

```
fmriqa_spikiness [opts] xmlfile [outputbase]
```

This program is usually called by wrapper scripts, and may not be useful to users on its own. This program takes a 4-D BXH- or XCEDE- wrapped dataset and calculates a 'spikiness' metric. Various 'spikiness' metrics are available and are selected using options. The size and meaning of the output data is dependent on the metric being calculated.

```
--overwrite
    Overwrite output files if they exist.
--version
    Print version string and exit.
--verbose
    More diagnostic output.
--metric <str>
--metric=<str>
    Which metric to return after fitting/detrending data.
    'diff' returns (value-fit) per voxel.
    'zscore' returns (value-fit)/stddev per voxel.
    'abszscore' returns (value-fit)/stddev per voxel.
    'afni' returns abs(value-fit)/mstddev per voxel (i.e. same as returned
    by Robert Cox's AFNI 3dDespike) where mstddev is a modified standard
    deviation that is less influenced by outlier points.
    'abszscoreslice' returns average abs(value-fit)/stddev per slice.
    'jackknife' (default) takes the output of 'abszscoreslice' and finds
    the "jackknife" z-score of each slice (over the volume) where the
    current slice is ignored in calculating mean/stddev.
    'jackknife' and 'abszscoreslice' produce a 2-D result set, whereas
    every other metric produces a 4-D result set.
--braintresh <double>
--braintresh=<double>
    Only voxels with a value greater than its_braintresh are used in the
    calculation. Other voxels will return a metric of 0. Default is minus
    infinity or thereabouts.
--fit_method <str>
--fit_method=<str>
    Which fitting/detrending method to use.
    'mean' (default) simply uses the mean of each voxel's time-course.
    'linear' does a linear L1 fit of each voxel time-course.
    'afni' L1-fits the function used in Robert Cox's AFNI 3dDespike program
    to each voxel's time-course.
--timeselect <str>
--timeselect=<str>
    Comma-separated list of timepoints to use (first timepoint is 0). Any
    timepoint can be a contiguous range, specified as two numbers separated
    by a colon, i.e. 'START:END'. An empty END implies the last timepoint.
    The default step of 1 (one) in ranges can be changed using
    'START:STEP:END', which is equivalent to
    'START,START+STEP,START+(2*STEP),...,END'.
--xselect <str>
--xselect=<str>
    Just like timeselect, but for the 'x' dimension.
--yselect <str>
--yselect=<str>
    Just like timeselect, but for the 'y' dimension.
--zselect <str>
--zselect=<str>
    Just like timeselect, but for the 'z' dimension.
```

A.36. Usage for iowa-signafive2bxh

Usage:
iowa-signafive2bxh imagedir output.bxh

This program creates an XML wrapper for Univ. of Iowa-style GE Signa5 images.

imagedir is a directory containing I.* images

```
--xcede
    Write XCEDE-style XML files.
--xcede2
    Write XCEDE 2-style XML files.
```

A.37. Usage for pfile2bxh

Usage:

```
pfile2bxh [opts] [pfilehdr imagedata1...] output.bxh
```

This program creates an XML wrapper for GE P-files (and associated reconstructed image data).

```
--forceversion <float>
--forceversion=<float>
    Force version of P-file to be interpreted as this number.
--msbfirst
    Indicates that data is big-endian (default: little-endian).
--dimorder <str>
--dimorder=<str>
    Comma-separated names of dimensions from fastest-moving to
    slowest-moving (default: x,y,z,t).
--elemtype <str>
--elemtype=<str>
    Provide element type of image data (one of int8, uint8, int16
    [default], uint16, int32, uint32, float32, or float64).
--usemrorigin
    This option extracts the origin from the tlhc_[RAS] fields in the MR
    structure. This is the default.
--useslicetableorigin
    The origin coordinates are extracted from the slice table at the end of
    the P-file header.
--inputsfromfile <str>
--inputsfromfile=<str>
    Read list of input files from this file.
--version
    Print version string and exit.
--hintsize x <size_t>
--hintsize x=<size_t>
--hintsize y <size_t>
--hintsize y=<size_t>
--hintsize z <size_t>
--hintsize z=<size_t>
--hintsize t <size_t>
--hintsize t=<size_t>
--hintorigin x <double>
--hintorigin x=<double>
--hintorigin y <double>
--hintorigin y=<double>
--hintorigin z <double>
--hintorigin z=<double>
--hintorigin t <double>
--hintorigin t=<double>
--hintspacing x <double>
--hintspacing x=<double>
--hintspacing y <double>
--hintspacing y=<double>
--hintspacing z <double>
--hintspacing z=<double>
--hintspacing t <double>
--hintspacing t=<double>
--hintgap x <double>
--hintgap x=<double>
--hintgap y <double>
--hintgap y=<double>
--hintgap z <double>
--hintgap z=<double>
```

```
--hintgapt <double>
--hintgapt=<double>
    These options will provide a hint to the program of the 'size',
    'origin', 'spacing', or 'gap' of the specified dimension. Some image
    types will not use all these values. In particular, sizex and sizey
    are assumed correct in most image headers, but they, as well as sizez
    and sizet options may be useful with image type 'pfile'. Origin and
    spacing hints will be used by most image types.
--forceorientation <str>
--forceorientation=<str>
    This option will force the labeled orientation of the image to match
    the given three letter orientation code. Each letter must come from
    the following groups in any order: R(ight) or L(eft); A(nterior) or
    P(osterior); S(uperior) or I(nferior). Only one letter from each group
    is allowed.
--xcede
    Write XCEDE-style XML files.
--xcede2
    Write XCEDE 2-style XML files.
```

A.38. Usage for presentation2xml

Usage:

```
presentation2xml instructions.txt inputevents.txt [outputevents.xml]
```

presentation2xml takes a Presentation output file and an instruction file, and creates an XML events file. The instruction file indicates which columns in the Presentation file are of interest and what they should map to in the output event file. If the output file is not specified, the event data is written to standard output.

Options:

```
--xcede2
    Write in XCEDE2 format.
--xcede2dataid=ID
    ID for the XCEDE 2 data element (default: auto-generated based
    on hostname, process ID, and current time)
--extracttable
    Instead of creating XML as output, output as tabular text.
    This is a no-op for most formats, and is really only useful for
    converting E-Prime "recovery" logs into a tabular form.
--columnnames
    If this options is specified, only the columns of the table are
    printed (one per line) and the program exits.
--subtractonset SECS
    This option subtracts SECS from all onset times (default is 0).
    This is in addition to any other normalization that may occur
    (see use of 'firstmrtime' below).
--colsep SEPARATOR
    This option specifies the column separator (default is tab).
```

The instruction file language is defined as follows:

```
COMMAND VALUESPEC [VALUESPEC...]
```

where COMMAND can be event, param, or block. VALUESPEC has one of the following formats:

```
[OUTVALUENAME=]COLUMNNAME[:UNITS]
OUTVALUENAME=@TEXT[:UNITS]
```

Each VALUESPEC defines values that should be passed through to the corresponding event, param, or block. In the first alternative listed above, the value comes from a column in the input file

(optionally renamed to OUTVALUENAME) and in the second alternatives, the value is directly specified preceded by a '@' character. If OUTVALUENAME= is missing, then the VALUESPEC is equivalent to:

```
COLUMNNAME=COLUMNNAME[:UNITS]
```

OUTVALUENAME may not contain the equals sign ('='), at sign ('@'), quotes, or whitespace.

Either COLUMNNAME or TEXT may contain quoted substrings to protect special characters like colon (':'), equals sign ('='), at sign ('@'), or whitespace; otherwise these special characters are prohibited. A single quote will protect all characters until the next single quote, and likewise for double quote. The following examples show equivalent VALUESPECs:

```
description=DESC
description=D'E"SC"
```

```
onset="Onset Time":secs
onset=Onset' 'Time:secs
onset=Onse't T'ime:secs
```

Unquoted spaces separate VALUESPECs.

```
-----
'event' command:
-----
```

Each 'event' command creates a class of events in the output event file, where the contents of the event are specified by the VALUESPECs. In general, for each matching row (more later), it creates an event with the following contents:

```
<event type="$type" units="$units">
  <onset>$onset</onset>
  <duration>$duration</duration>
  <name>$name</name>
  <description>$description</description>
  <value name="OUTVALUENAME1" units="UNITS1">VALUE1</value>
  <value name="OUTVALUENAME2" units="UNITS2">VALUE2</value>
  ...
</event>
```

VALUESPECs whose OUTVALUENAMES start with a dollar sign (\$) are "magic", and are interpreted in a value-specific way. VALUESPECs whose OUTVALUENAMES start with a percent sign (%) are explicitly non-magic. Any OUTVALUENAME not starting with a % or \$ is assumed to have an implicit % unless it matches a list of pre-defined magic values (below), in which case an implicit \$ is assumed.

Pre-defined magic values '\$type', '\$units', '\$onset', '\$duration', '\$name', and '\$description' are put in the appropriate child element or attribute of <event> (shown above). Only the '\$onset' VALUESPEC is required. Default value for '\$duration' is zero. All non-magic values are placed in <value> elements.

The pre-defined magic value '\$DURUNTIL' indicates that any row in the input used to create an event will have an ending time specified by the value of column COLUMNNAME in the current row. Likewise, the value '\$DURUNTILNEXTROW' does the same thing, but grabs the value from the next row. These are used to calculate the duration of this event. This may be specified more than once, and the first non-NULL column will be used. This option is used when a row does not have a duration column, and it must be calculated based on times in this or the subsequent row.

By default, only those rows whose '\$onset' column is non-empty and

non-NULL will be processed as events. Certain magic OUTVALUENAMES further restrict the rows that are used for this event command. '\$MATCH' and '\$MATCHNONZERO' specify a column whose values indicate whether that row should be selected -- for '\$MATCH', the values must be non-empty and non-'NULL'; for '\$MATCHNONZERO', the values must also be non-zero. With '\$MATCHEQUAL', one specifies both a column and an actual value to match -- for the '\$MATCHEQUAL' value name (and only the '\$MATCHEQUAL' value name) the VALUESPEC syntax is extended in the following way:

```
$MATCHEQUAL=COLUMNNAME@MATCHVALUE
```

where COLUMNNAME and MATCHVALUE are the two relevant parameters.

```
-----
'block' command:
-----
```

The block command has the same usage as the event command. The same magic values apply to block commands as event commands. An '\$onset' value is again required, and '\$duration' is optional (assumed to be zero [0] if missing).

```
-----
'param' command:
-----
```

Each param command specifies a list of columns that should be put in the <params> section of the event file. These represent parameters that are constant (or default) throughout the events file. Each VALUESPEC represents one item to put in the <params> element as such:

```
<params>
...
  <value name="OUTVALUENAME1" units="UNITS1">VALUE1</value>
  <value name="OUTVALUENAME2" units="UNITS2">VALUE2</value>
...
</params>
```

Only the first non-empty, non-NULL field in the column specified by a 'param' will be used. Be aware of this if this column does not have the same value in every row.

There is one magic OUTVALUENAME (maybe more later) '\$firstmritime', which will generate the following element:

```
<params>
  <firstmritime>0</firstmritime>
</params>
```

If '\$firstmritime' is specified, it (and all '\$onset' VALUESPECs) must have UNITS specified. All '\$onset' columns are normalized by this value, so their units and '\$firstmritime' units must match.

A.39. Usage for printfrags

Usage:
 printfrags xmlfile

This program prints out the 'frags' in a BXH/XCEDE file.

A.40. Usage for showplay2xml

```
Can't locate XML/Parser.pm in @INC (@INC contains: /builddir/build/BUILD/
RELEASE_1_10_7/bxh_xcede_tools/utils /usr/local/lib64/perl5 /usr/local/share/perl5 /
usr/lib64/perl5/vendor_perl /usr/share/perl5/vendor_perl /usr/lib64/perl5 /usr/share/
perl5 .) at /builddir/build/BUILD/RELEASE_1_10_7/bxh_xcede_tools/utils/XML/XPath/
XMLParser.pm line 7.
BEGIN failed--compilation aborted at /builddir/build/BUILD/RELEASE_1_10_7/
bxh_xcede_tools/utils/XML/XPath/XMLParser.pm line 7.
Compilation failed in require at /builddir/build/BUILD/RELEASE_1_10_7/bxh_xcede_tools/
utils/XML/XPath.pm line 13.
BEGIN failed--compilation aborted at /builddir/build/BUILD/RELEASE_1_10_7/
bxh_xcede_tools/utils/XML/XPath.pm line 13.
Compilation failed in require at ../../utils/showplay2xml line 13.
BEGIN failed--compilation aborted at ../../utils/showplay2xml line 13.
```

A.41. Usage for signafive2bxh

Usage:
 signafive2bxh [opts] [signa5files...] output.bxh

This program creates an XML wrapper for GE Signa5 image files.

```
--dimzsize <size_t>
--dimzsize=<size_t>
    Specifies the size of the z dimension (i.e. number of slices per
    timepoint). Default is to use the number of input files. Equivalent
    to (and overrides) --hintsizez.
--dimtsiz <size_t>
--dimtsiz=<size_t>
    Specifies the size of the t dimension (i.e. number of timepoints).
    Default is number of input files divided by number of slices per
    timepoint (as specified by --dimzsize). Equivalent to (and overrides)
    --hintsize.
--inputsfromfile <str>
--inputsfromfile=<str>
    Read list of input files from this file.
--version
    Print version string and exit.
--hintsize x <size_t>
--hintsize x=<size_t>
--hintsize y <size_t>
--hintsize y=<size_t>
--hintsize z <size_t>
--hintsize z=<size_t>
--hintsize t <size_t>
--hintsize t=<size_t>
--hintorigin x <double>
--hintorigin x=<double>
--hintorigin y <double>
--hintorigin y=<double>
--hintorigin z <double>
--hintorigin z=<double>
--hintorigin t <double>
--hintorigin t=<double>
--hintspacing x <double>
--hintspacing x=<double>
--hintspacing y <double>
--hintspacing y=<double>
--hintspacing z <double>
```

```

--hintspacingz=<double>
--hintspacingt <double>
--hintspacingg=<double>
--hintgapx <double>
--hintgapx=<double>
--hintgapy <double>
--hintgapy=<double>
--hintgapz <double>
--hintgapz=<double>
--hintgapt <double>
--hintgapt=<double>
    These options will provide a hint to the program of the 'size',
    'origin', 'spacing', or 'gap' of the specified dimension. Some image
    types will not use all these values. In particular, sizex and sizey
    are assumed correct in most image headers, but they, as well as sizex
    and sizet options may be useful with image type 'pfile'. Origin and
    spacing hints will be used by most image types.
--forceorientation <str>
--forceorientation=<str>
    This option will force the labeled orientation of the image to match
    the given three letter orientation code. Each letter must come from
    the following groups in any order: R(ight) or L(eftrightarrow); A(nterior) or
    P(osterior); S(uperior) or I(nferior). Only one letter from each group
    is allowed.
--xcede
    Write XCEDE-style XML files.
--xcede2
    Write XCEDE 2-style XML files.

```

A.42. Usage for xcede_extract_schedules.pl

```

Usage:
xcede_extract_schedules.pl [OPTIONS] outputformat xmldir outputlocation queryfiles...
xcede_extract_schedules.pl [OPTIONS] outputformat xmlfile outputlocation
queryfiles...
xcede_extract_schedules.pl [OPTIONS] outputformat xmlfile1,xmlfile2,...
outputlocation queryfiles...
Values for 'outputformat':
fsl -
    outputlocation should be a directory. Output is one or more .stf files
    with base name derived from the conditions specified in the queryfile(s),
    each having three columns: onset, duration, and weight.
par -
    outputlocation should be a file, which will have four columns: onset,
    condition index, duration, and condition name. The conditions will be
    numbered from zero (0), in the order they are specified in the query file.

Options:
--overwrite
    Overwrite existing output files (otherwise error and exit).
--bypass
    Missing files or other errors will result in warning messages but
    processing on other files will continue, and the exit status will
    be 0 (success).
--verbose
    Provide more info for debugging.
--tr TR
    Specify the TR. Equivalent to specifying "forcetr" in the query file.
    This is used to convert "ptsbefore" and "ptsafter" options into seconds.
--fileprefix PREFIX
    Except for outputformat 'par', output file names will have this prefix.
    Default is no prefix.

```



```
--weightquery STRING
    If specified (and if supported by the output type) the weight for an
    event is 1 if this query matches on the event, and 0 otherwise.
    If neither --weightquery or --weightvaluequery is specified, the
    default weight is 1 for all events.
--weightvaluequery STRING
    If specified (and if supported by the output type) the weight for an
    event is given by the value matching this query on each event, and the
    weight is zero for any event where this query does not match a value.
    If neither --weightquery or --weightvaluequery is specified, the
    default weight is 1 for all events.
```

A.43. Usage for ximg2bxh

Usage:

```
ximg2bxh [opts] [ximgfiles...] output.bxh
```

This program creates an XML wrapper for GE Ximg image files.

```
--dimzsize <size_t>
--dimzsize=<size_t>
    Specifies the size of the z dimension (i.e. number of slices per
    timepoint). Default is to use the number of input files. Equivalent
    to (and overrides) --hintsizez.
--dimtsize <size_t>
--dimtsize=<size_t>
    Specifies the size of the t dimension (i.e. number of timepoints).
    Default is number of input files divided by number of slices per
    timepoint (as specified by --dimzsize). Equivalent to (and overrides)
    --hintsize_t.
--inputsfromfile <str>
--inputsfromfile=<str>
    Read list of input files from this file.
--version
    Print version string and exit.
--hintsize_x <size_t>
--hintsize_x=<size_t>
--hintsize_y <size_t>
--hintsize_y=<size_t>
--hintsize_z <size_t>
--hintsize_z=<size_t>
--hintsize_t <size_t>
--hintsize_t=<size_t>
--hintorigin_x <double>
--hintorigin_x=<double>
--hintorigin_y <double>
--hintorigin_y=<double>
--hintorigin_z <double>
--hintorigin_z=<double>
--hintorigin_t <double>
--hintorigin_t=<double>
--hintspacing_x <double>
--hintspacing_x=<double>
--hintspacing_y <double>
--hintspacing_y=<double>
--hintspacing_z <double>
--hintspacing_z=<double>
--hintspacing_t <double>
--hintspacing_t=<double>
--hintgap_x <double>
--hintgap_x=<double>
--hintgap_y <double>
--hintgap_y=<double>
```

```

--hintgapz <double>
--hintgapz=<double>
--hintgapt <double>
--hintgapt=<double>
    These options will provide a hint to the program of the 'size',
    'origin', 'spacing', or 'gap' of the specified dimension. Some image
    types will not use all these values. In particular, sizex and sizey
    are assumed correct in most image headers, but they, as well as sizez
    and sizet options may be useful with image type 'pfile'. Origin and
    spacing hints will be used by most image types.
--forceorientation <str>
--forceorientation=<str>
    This option will force the labeled orientation of the image to match
    the given three letter orientation code. Each letter must come from
    the following groups in any order: R(ight) or L(ef); A(nterior) or
    P(osterior); S(uperior) or I(nferior). Only one letter from each group
    is allowed.
--xcede
    Write XCEDE-style XML files.
--xcede2
    Write XCEDE 2-style XML files.

```

A.44. Usage for fmriqa_volmeasures

Usage:

```
fmriqa_volmeasures [opts] xmlfile
```

This program calculates various measures per volume of the input 4-D time series, such as mean/minimum/maximum intensity, standard deviation, and center-of-mass in all three dimensions. The input file must be BXH or XCEDE file, and the output will be written to standard output.

```

--version
    Print version string and exit.
--maskfile <str>
--maskfile=<str>
    Use this mask (should be a BXH or XCEDE XML file).
--timeselect <str>
--timeselect=<str>
    Comma-separated list of timepoints to use (first timepoint is 0). Any
    timepoint can be a contiguous range, specified as two numbers separated
    by a colon, i.e. 'START:END'. An empty END implies the last timepoint.
    The default step of 1 (one) in ranges can be changed using
    'START:STEP:END', which is equivalent to
    'START,START+STEP,START+(2*STEP),...,END'.
--xselect <str>
--xselect=<str>
    Just like timeselect, but for the 'x' dimension.
--yselect <str>
--yselect=<str>
    Just like timeselect, but for the 'y' dimension.
--zselect <str>
--zselect=<str>
    Just like timeselect, but for the 'z' dimension.

```