CIFTI-2 Connectivity File Formats Documentation

CIFTI working group

Tim Coalson, Matt Glasser, John Harwell, David Van Essen, Mark Jenkinson, Robert Oostenveld

FINAL (1 March, 2014)

Introduction to CIFTI-2

Over the past decade, standardized file formats for many types of neuroimaging data have been widely adopted by the neuroimaging community and by the developers of major software platforms. Notably successful examples are the NIfTI-1 volume file format established by the NIfTI (Neuroimaging Informatics Technology Initiative) Committee and the GIFTI surface file format established by the GIFTI (Geometry Informatics Technology Initiative) committee.

A need for additional data formats recently emerged in order to handle data files related to brain connectivity and function (e.g., using fMRI, diffusion MRI, and tractography). These include the need to handle very large files and data representations that reflect aspects of brain geometry and circuitry. In 2011, two interrelated developments contributed to addressing these needs. (1) The NIfTI-2 file format was adopted by the NIfTI committee in order to circumvent the 16-bit limitation in the dimensions supported by NIfTI-1 files (32,767 dimension length). NIfTI-2 file header indices are 64-bit integers, allowing nearly unlimited dimension length (up to 9 quintillion) (see http://www.nitrc.org/forum/message.php?msg_id=3738 and http://nifti.nimh.nih.gov/nifti-2). (2) The CIFTI-1 file format was adopted by an ad hoc Connectivity Sub-committee of the Neuroimaging Informatics Technology Initiative that included current and prospective developers of connectivity-related software platforms. CIFTI-1 files are based on NIfTI-2, but they have additional information that allow them to encompass surface data (represented as vertices) and/or volume data (represented as voxels); these are collectively called 'brainordinates' in general, and 'grayordinates' if they only represent gray matter (Glasser et al 2013).

While CIFTI-1 has proven useful for analyses carried out by the Human Connectome Project (HCP) and the associated Connectome Workbench ("Workbench") visualization platform, several limitations motivate the introduction of the CIFTI-2 file format specified in the present document (see Appendix B for a list of changes). The increment in version number reflects the fact that CIFTI-2 is not backwards-compatible with CIFTI-1. However, conversion from CIFTI-1 to CIFTI-2 will be possible using a command-line utility provided with Workbench. Workbench will also maintain read compatibility with existing CIFTI-1 files at least until the next major revision of CIFTI.

CIFTI data files include additional information about the matrix indices, which is encoded in XML. In the case of CIFTI-2 files, this XML is placed into a header extension in a NIfTI-2 file. CIFTI-2 files are also given a two-part filename extension which distinguishes between different types of CIFTI files. This document defines the XML format of this information for CIFTI-2, the filename extensions, and other details of storage in a NIfTI-2 file. The examples that are discussed will focus on fMRI and diffusion imaging data, but CIFTI-2 is designed to also handle other data types such as MEG data.

The NITRC (Neuroimaging Informatics Tools and Resources Clearinghouse, http://www.nitrc.org) website is a portal for neuroimaging resources. CIFTI, GIFTI, and NIfTI all use NITRC for the discussion and dissemination of project related information. Documentation,

example files, and message forums have been set up for the CIFTI-2 file format on the NITRC website. Visit http://www.nitrc.org/projects/cifti for more information.

General Concepts of CIFTI

A CIFTI file has two essential parts: a matrix of values, and the CIFTI XML, which describes how the indices of the matrix are to be interpreted. Each dimension of the matrix uses one of several possible *mapping* types. Some of these are straightforward, such as "series", which represents regularly spaced samples. Other mapping types are more complex, such as "brain models", which can contain vertices from one or more surfaces and/or selected groups of voxels. In this way, a 2 dimensional matrix can contain joint timeseries data for multiple surfaces and voxels, by having a "brain models" mapping for the indices along one dimension, and a "series" mapping for the indices along the other dimension.

For clarity, we define the term "index" to mean an integer, starting from zero, indicating a position along one dimension of the matrix.

Many CIFTI file types contain at least one mapping which is spatial in nature, in that it refers to voxels and/or surface vertices. These use the "brain models" and "parcels" mapping types. While the CIFTI XML contains the required information to assign coordinates to voxel indices, it does not contain spatial information about surface vertices, or about the triangles that connect them. One reason is that surface data is frequently viewed on a non-anatomical surface (e.g., "inflated"), while processing generally takes place on anatomical surfaces (e.g., "midthickness", "white", etc). When displaying surface data from a CIFTI file, or using CIFTI in an analysis that requires surface coordinates or triangles, the user is expected to provide the desired/appropriate surface, for instance as a GIFTI surface file. The CIFTI file does, however, store the name of the structure (e.g., left cortex) and the vertex count from the data used to generate the CIFTI file, to prevent most cases in which a user might accidentally specify a surface not matched to the data.

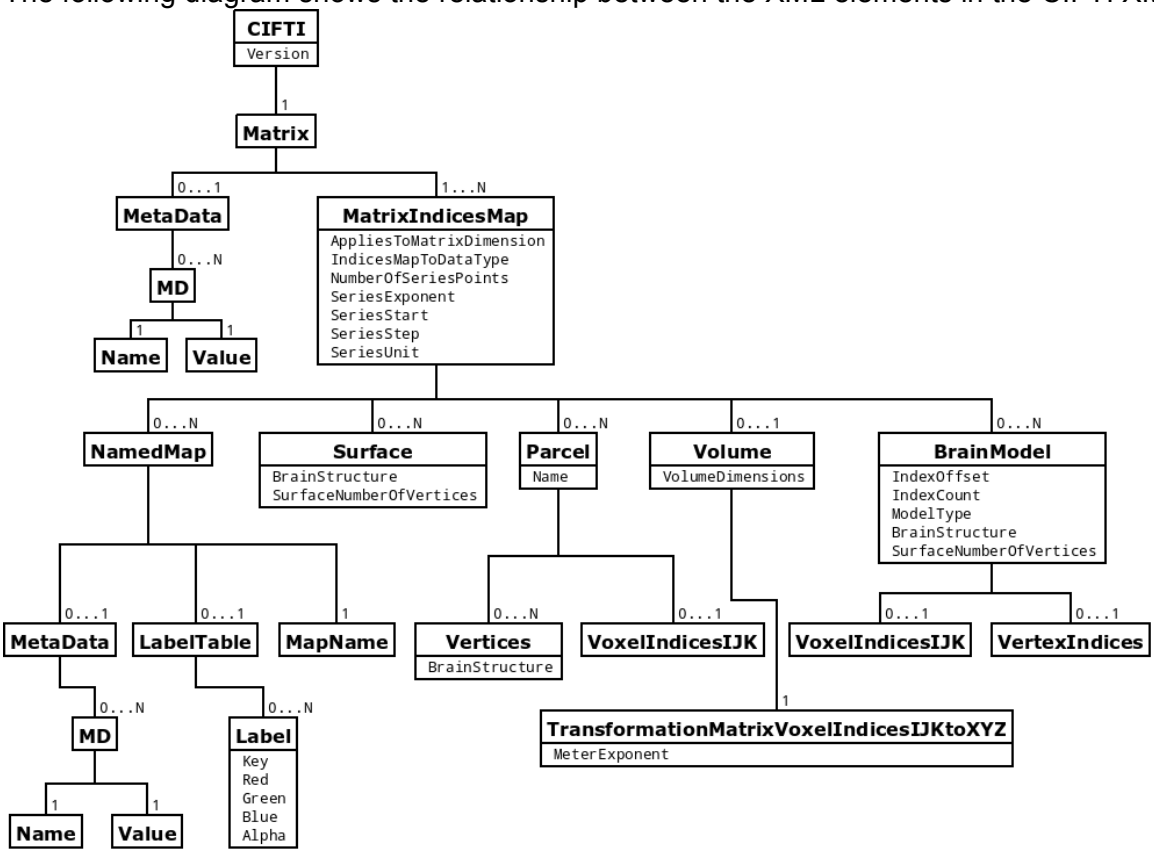
Matrix Storage

A major use case for CIFTI involves matrices that are too large to be loaded entirely into memory on typical computers. Accessing the matrix while it is not in memory (e.g., while on disk in a NIfTI-2 file) therefore needs to be efficient for specified access patterns. To this end, we define the term "row" to mean a one dimensional subset of the matrix consisting of all possible indices of the first CIFTI dimension, and one index for each other dimension. Note that this differs from the standard definition of row for mathematical two dimensional matrices. For CIFTI-2, as stored in a NIfTI-2 file, each row is contiguous in the file. This minimizes seeking when reading any single row and thus provides the desired efficiency. Additionally, the NIfTI-2 file should be used in uncompressed form, as opposed to the common practice of using gzip compression on NIfTI volume files, because seeking within a gzip compressed file is difficult and slow. Any future storage methods for CIFTI data will be designed to ensure efficient access to single rows in the matrix.

CIFTI-2 files use the NIfTI-2 file format for storing the data matrix and CIFTI XML. The CIFTI XML is stored in a header extension, as per the NIfTI standard. For additional details on how CIFTI-2 is stored in a NIfTI-2 file, see the section "Storage in NIfTI-2".

Cifti XML Hierarchy

The following diagram shows the relationship between the XML elements in the CIFTI XML.



Mapping Types

The MatrixIndicesMap XML element assigns a mapping (interpretation) to one or more matrix dimensions. It has an AppliesToMatrixDimension attribute, which is a comma separated list of the data matrix dimension(s) to which this particular mapping applies. If the list contains “0”, the mapping applies along dimension 0 (the first dimension), if it contains “1”, the mapping applies along dimension 1, and so on. For each matrix dimension, exactly one MatrixIndicesMap element must list it in the AppliesToMatrixDimension attribute. It is acceptable to use one MatrixIndicesMap element for multiple dimensions of the data matrix when they have identical mappings (e.g. a dense connectome that is all grayordinates by all grayordinates). The MatrixIndicesMap element and its attributes and/or children determine the length of the mapping, which must match the length of the matrix dimension(s) it applies to. Two dimensions with the same mapping type sometimes require separate MatrixIndicesMap elements, as there is additional information within each MatrixIndicesMap element that affects its interpretation.

The IndicesMapToDataType attribute of the MatrixIndicesMap element defines the mapping type as one of the following (with more detailed descriptions given after):

Value	Description
CIFTI_INDEX_TYPE_BRAIN_MODELS	The dimension represents one or more geometrical brain models (i.e., consisting of voxels or surface vertices).
CIFTI_INDEX_TYPE_PARCELS	The dimension represents groups of surface vertices and/or voxels.
CIFTI_INDEX_TYPE_SERIES	The dimension represents a series of regular samples.
CIFTI_INDEX_TYPE_SCALARS	The dimension represents named continuous value maps.

CIFTI_INDEX_TYPE_LABELS	The dimension represents named integer value maps.

The IndicesMapToDataType attribute also determines what additional attributes and child XML elements are needed in the MatrixIndicesMap element.

For CIFTI_INDEX_TYPE_BRAIN_MODELS, the mapping assigns one brainordinate to each index. It comprises one or more BrainModel elements, each of which specifies one brain structure (e.g. left thalamus, left cortex, etc.); together these make up the set of brainordinates. Additionally, all brainordinates within each brain model (one surface or one group of structurally related voxels) occupy a contiguous range of indices. For this type, the MatrixIndicesMap element must follow these rules:

- It must contain one or more BrainModel elements.
- Each BrainModel child element must contain either one VertexIndices element or one VoxelIndicesIJK element, depending on whether it represents surface or volumetric data.
- The value of the ModelType attribute in each BrainModel element must indicate which representation type that BrainModel element uses.
- Every BrainModel element must have a different value of BrainStructure than other BrainModel elements which use the same representation type in the same MatrixIndicesMap element.
- The IndexOffset and IndexCount attributes of the BrainModel elements in this MatrixIndicesMap must specify non-overlapping index ranges that do not leave any indices unassigned.
- The sum of the IndexCount attributes of the BrainModel elements must equal the length of the dimension(s) this MatrixIndicesMap element applies to.
- If at least one BrainModel element contains a VoxelIndicesIJK element, the MatrixIndicesMap element must contain a Volume element, in order to define the volume space. All voxel indices used by any BrainModel element in this MatrixIndicesMap must be within the dimensions specified in the VolumeDimensions attribute of the Volume element.

For CIFTI_INDEX_TYPE_PARCELS, the mapping assigns a set of parcels to the indices, with one parcel (a set of voxels and/or vertices from one or more surfaces) for each index. Each parcel is defined by a separate Parcel element. For this type, the MatrixIndicesMap element must follow these rules:

- It must contain the same number of Parcel elements as the length of the dimension(s) this MatrixIndicesMap element applies to.
- Each Parcel element may contain one or more Vertices elements, and may contain one VoxelIndicesIJK element.
- If any Parcel element in this MatrixIndicesMap contains a VoxelIndicesIJK element, then this MatrixIndicesMap element must contain a Volume element. All voxel indices used by any Parcel element in this MatrixIndicesMap must be within the dimensions specified in the VolumeDimensions attribute of the Volume element.
- Each Vertices element within a single Parcel element must have a different value of BrainStructure.
- For any value of BrainStructure used in a Vertices element, there must be exactly one Surface element in this MatrixIndicesMap element with a matching BrainStructure attribute.
- Parcel elements may not use a vertex or voxel that is used in another Parcel element in the same MatrixIndicesMap element.

For CIFTI_INDEX_TYPE_SERIES, the mapping assigns a value to each index, such that the values of each index are regularly spaced (for instance, time values for a timeseries). For this type, the MatrixIndicesMap element must follow these rules:

- The NumberOfSeriesPoints attribute must match the length of the dimension(s) this MatrixIndicesMap element applies to.
- The first index is associated with the quantity denoted by the SeriesStart attribute, multiplied by 10 raised to the power of SeriesExponent.
- Each other index is associated with a quantity greater than that of the previous index by the value of SeriesStep multiplied by 10 raised to the power of SeriesExponent.
- The quantities that this mapping associates with the indices use the units specified by the SeriesUnit attribute of the MatrixIndicesMap.

For CIFTI_INDEX_TYPE_SCALARS, the mapping assigns a name (string), and optional user metadata (string key/value pairs) to each index. This can be used to store a set of maps, such as functional statistical maps. For this type, the MatrixIndicesMap element must follow these rules:

- The MatrixIndicesMap element must contain the same number of NamedMap elements as the length of the dimension(s) this MatrixIndicesMap element applies to.
- Each NamedMap element must contain one MapName element, and may contain one MetaData element.

For CIFTI_INDEX_TYPE_LABELS, the mapping assigns a name and a label table (which associates an integer key to a name and color), and optionally user metadata to each index. Additionally, the use of this mapping type changes the meaning of the values in the matrix, in that they are to be interpreted as keys in the label table that corresponds to their index along the dimension that uses the CIFTI_INDEX_TYPE_LABELS type. A CIFTI file should never use this type on more than one dimension of a matrix. For this type, the MatrixIndicesMap element must follow these rules:

- The MatrixIndicesMap element must contain the same number of NamedMap elements as the length of the dimension this MatrixIndicesMap element applies to.
- Each NamedMap element must contain a MapName element and a LabelTable element, and may contain a MetaData element.

CIFTI XML Elements

The following is a list of all CIFTI XML elements, their attributes and children, and the meaning of each. They are ordered in a depth-first walk of the XML hierarchy diagram, skipping repeated elements, in such a way as to put each element near its parent element(s). For the list in alphabetical order, see Appendix C. Following each child element are quantity indicators, in parentheses, that describe the allowable quantity of that child element. This quantity is expressed in one of two ways. A number indicates a fixed number of children. A variable quantity is shown as a range of numbers separated by an ellipsis. When 'N' is present on the right side of the ellipsis, it indicates that there is no fixed upper bound in the XML specification (though it may be required to match something, such as the length of a dimension). For some examples of CIFTI XML, see Appendix D.

CIFTI Element

- *Description* – The root of the CIFTI XML.
- *Attributes*
    - **Version** – Indicates version of the CIFTI XML. For CIFTI-2, this must have the

value “2”. If the value is “1” or “1.0”, this document does not apply, and instead the file should follow the CIFTI-1 specification: http://www.nitrc.org/plugins/mwiki/index.php/cifti:MainPage. It should be noted that CIFTI-2 compliant software is not required to be able to read CIFTI-1.

- *Child Elements*
- **Matrix** (1). At this time, there is only one Matrix child. Future versions of CIFTI may allow more than one Matrix child.
- *Text Content: [NA]*
- *Parent Element: [NA]*

Matrix Element

- *Description* – Contains child elements that describe the meaning of the values in the matrix.
- *Attributes: [NA]*
- *Child Elements*
- **MetaData** (0…1)
- **MatrixIndicesMap** (1…N)
- *Text Content: [NA]*
- *Parent Element* – **CIFTI**

MetaData Element

- *Description* – Provides a simple method for user-supplied metadata that associates names with values.
- *Attributes: [NA]*
- *Child Elements*
- **MD** (0…N)
- *Text Content: [NA]*
- *Parent Elements* – **Matrix, NamedMap**

MD Element

- *Description* – A single metadata entry consisting of a name and a value.
- *Attributes: [NA]*
- *Child Elements*
- **Name** (1)
- **Value** (1)
- *Text Content: [NA]*
- *Parent Element* – **MetaData**

Name Element

- *Description* – Content is the name of a metadata entry.
- *Attributes: [NA]*
- *Child Elements: [NA]*
- *Text Content* – Name of metadata element.
- *Parent Element* – **MD**

Value Element

- *Description* – Content is the value of a metadata entry.
- *Attributes: [NA]*
- *Child Elements: [NA]*
- *Text Content* – Value of metadata element.
- *Parent Element* – **MD**

MatrixIndicesMap Element

- *Description* – Provides a mapping between matrix indices and their interpretation.
- *Attributes*
- **AppliesToMatrixDimension** – Lists the dimension(s) of the matrix to which this MatrixIndicesMap applies. The dimensions of the matrix start at zero (dimension 0 describes the indices along the first dimension, dimension 1 describes the indices along the second dimension, etc.). If this MatrixIndicesMap applies to more than one matrix dimension, the values are separated by a comma.
- **IndicesMapToDataType** – Type of data to which the MatrixIndicesMap applies.
- **NumberOfSeriesPoints** - Indicates how many samples there are in a series mapping type. For example, this could be the number of timepoints in a timeseries.
- **SeriesExponent** - Integer, SeriesStart and SeriesStep must be multiplied by 10 raised to the power of the value of this attribute to give the actual values assigned to indices (e.g., if SeriesStart is "5" and SeriesExponent is "-3", the value of the first series point is 0.005).
- **SeriesStart** - Indicates what quantity should be assigned to the first series point.
- **SeriesStep** - Indicates amount of change between each series point.
- **SeriesUnit** – Indicates the unit of the result of multiplying SeriesStart and SeriesStep by 10 to the power of SeriesExponent.
- *Child Elements*
- **BrainModel** (0…N)
- **NamedMap** (0…N)
- **Parcel** (0…N)
- **Surface** (0…N)
- **Volume** (0…1)
- *Text Content: [NA]*
- *Parent Element* – **Matrix**

IndicesMapToDataType

Value	Description
CIFTI_INDEX_TYPE_BRAIN_MODELS	The dimension represents one or more brain models.
CIFTI_INDEX_TYPE_PARCELS	The dimension represents a parcellation scheme.
CIFTI_INDEX_TYPE_SERIES	The dimension represents a series of regular samples.
CIFTI_INDEX_TYPE_SCALARS	The dimension represents named scalar maps.
CIFTI_INDEX_TYPE_LABELS	The dimension represents named label maps.

SeriesUnit

Value
SECOND

HERTZ
METER
RADIAN

NamedMap Element

- *Description* – Associates a name, optional metadata, and possibly a LabelTable with an index in a map.
- *Attributes: [NA]*
- *Child Elements*
- **MapName** (1)
- **LabelTable** (0…1)
- **MetaData** (0…1)
- *Text Content: [NA]*
- *Parent Element* – **MatrixIndicesMap**

MapName Element

- *Description* – Contains a map name.
- *Attributes: [NA]*
- *Child Elements: [NA]*
- *Text Content* – The name of the map.
- *Parent Element* - **NamedMap**

LabelTable Element

- *Description* - Used by NamedMap when IndicesMapToDataType is "CIFTI_INDEX_TYPE_LABELS" in order to associate names and display colors with label keys. Note that LABELS is the only mapping type that uses a LabelTable. Display coloring of continuous-valued data is not specified by CIFTI-2.
- *Attributes: [NA]*
- *Child Elements* – **Label** (0…N)
- *Text Content: [NA]*
- *Parent Element* – **NamedMap**

Label Element

- *Description* – Associates a label key value with a name and a display color.
- *Attributes*
- **Key** – Integer, data value which is assigned this name and color.
- **Red** – Red color component for label. Value is floating point with range 0.0 to 1.0.
- **Green** – Green color component for label. Value is floating point with range 0.0 to 1.0.
- **Blue** – Blue color component for label. Value is floating point with range 0.0 to 1.0.
- **Alpha** – Alpha color component for label. Value is floating point with range 0.0 to 1.0.
- *Child Elements: [NA]*
- *Text Content* – Name of the label.
- *Parent Element* – **LabelTable**

Volume Element

- *Description* – Provides information about the volume for any mappings that use voxels.
- *Attributes*
- **VolumeDimensions** – Three integer values separated by commas, the lengths of the three volume file dimensions that are related to spatial coordinates, in number of voxels. Voxel indices (which are zero-based) that are used in the mapping that this element applies to must be within these dimensions.
- *Child Elements*
- **TransformationMatrixVoxelIndicesIJKtoXYZ** (1)
- *Text Content: [NA]*
- *Parent Element* – **MatrixIndicesMap**

TransformationMatrixVoxelIndicesIJKtoXYZ Element

- *Description* – Contains a matrix that translates Voxel IJK Indices to spatial XYZ coordinates (+X=>right, +Y=>anterior, +Z=> superior). The resulting coordinate is the center of the voxel.
- *Attributes*
- **MeterExponent** - Integer, specifies that the coordinate result from the transformation matrix should be multiplied by 10 to this power to get the spatial coordinates in meters (e.g., if this is "-3", then the transformation matrix is in millimeters).
- *Child Elements: [NA]*
- *Text Content* - Sixteen floating-point values, in row-major order, that form a 4x4 homogeneous transformation matrix.
- *Parent Element* – **Volume**

The transformation matrix below is encoded into the XML as "m1 m2 m3 m4 m5 m6 m7 m8 m9 m10 m11 m12 m13 m14 m15 m16". The last row (elements m13, m14, m15, and m16) will always be [0, 0, 0, 1].

```
 m1  m2  m3  m4
 m5  m6  m7  m8
 m9 m10 m11 m12
m13 m14 m15 m16
```

BrainModel Element

- *Description* – Maps a range of indices to surface vertices or voxels when IndicesMapToDataType is "CIFTI_INDEX_TYPE_BRAIN_MODELS."
- *Attributes*
- **IndexOffset** - The matrix index of the first brainordinate of this BrainModel. Note that matrix indices are zero-based.
- **IndexCount** – Number of surface vertices or voxels in this brain model, must be positive.
- **ModelType** – Type of model representing the brain structure (surface or voxels). Valid values are listed in the table below.
- **BrainStructure** – Identifies the brain structure. Valid values for BrainStructure are listed in the table below. However, if the needed structure is not listed in the table, a message should be posted to the CIFTI Forum so that a standardized name can be created for the structure and added to the table.

- **SurfaceNumberOfVertices** - When ModelType is CIFTI_MODEL_TYPE_SURFACE this attribute contains the actual (or true) number of vertices in the surface that is associated with this BrainModel. When this BrainModel represents all vertices in the surface, this value is the same as IndexCount. When this BrainModel represents only a subset of the surface's vertices, IndexCount will be less than this value.
- *Child Elements*
- **VertexIndices** (0…1)
- **VoxelIndicesIJK** (0…1)
- *Text Content: [NA]*
- *Parent Element* – **MatrixIndicesMap**

ModelType Values

ModelType	Description
CIFTI_MODEL_TYPE_SURFACE	Modeled using surface vertices.
CIFTI_MODEL_TYPE_VOXELS	Modeled using voxels.

BrainStructure Values

BrainStructure
CIFTI_STRUCTURE_ACCUMBENS_LEFT
CIFTI_STRUCTURE_ACCUMBENS_RIGHT
CIFTI_STRUCTURE_ALL_WHITE_MATTER
CIFTI_STRUCTURE_ALL_GREY_MATTER
CIFTI_STRUCTURE_AMYGDALA_LEFT
CIFTI_STRUCTURE_AMYGDALA_RIGHT
CIFTI_STRUCTURE_BRAIN_STEM
CIFTI_STRUCTURE_CAUDATE_LEFT
CIFTI_STRUCTURE_CAUDATE_RIGHT
CIFTI_STRUCTURE_CEREBELLAR_WHITE_MATTER_LEFT
CIFTI_STRUCTURE_CEREBELLAR_WHITE_MATTER_RIGHT
CIFTI_STRUCTURE_CEREBELLUM
CIFTI_STRUCTURE_CEREBELLUM_LEFT
CIFTI_STRUCTURE_CEREBELLUM_RIGHT
CIFTI_STRUCTURE_CEREBRAL_WHITE_MATTER_LEFT
CIFTI_STRUCTURE_CEREBRAL_WHITE_MATTER_RIGHT
CIFTI_STRUCTURE_CORTEX
CIFTI_STRUCTURE_CORTEX_LEFT
CIFTI_STRUCTURE_CORTEX_RIGHT
CIFTI_STRUCTURE_DIENCEPHALON_VENTRAL_LEFT
CIFTI_STRUCTURE_DIENCEPHALON_VENTRAL_RIGHT
CIFTI_STRUCTURE_HIPPOCAMPUS_LEFT
CIFTI_STRUCTURE_HIPPOCAMPUS_RIGHT
CIFTI_STRUCTURE_OTHER
CIFTI_STRUCTURE_OTHER_GREY_MATTER
CIFTI_STRUCTURE_OTHER_WHITE_MATTER
CIFTI_STRUCTURE_PALLIDUM_LEFT
CIFTI_STRUCTURE_PALLIDUM_RIGHT
CIFTI_STRUCTURE_PUTAMEN_LEFT

CIFTI_STRUCTURE_PUTAMEN_RIGHT
CIFTI_STRUCTURE_THALAMUS_LEFT
CIFTI_STRUCTURE_THALAMUS_RIGHT

VertexIndices Element

- *Description* – Contains a list of vertex indices for a BrainModel with ModelType equal to CIFTI_MODEL_TYPE_SURFACE.
- *Attributes: [NA]*
- *Child Elements: [NA]*
- *Text Content* – The vertex indices (which are independent for each surface, and zero-based) that are used in this brain model, with each index separated by a whitespace character. The parent BrainModel's IndexCount attribute indicates the number of indices in this element's content.
- *Parent Element* – **BrainModel**

VoxelIndicesIJK Element

- *Description* – Identifies the voxels that model a brain structure, or participate in a parcel. Note that when this is a child of BrainModel, the IndexCount attribute of the BrainModel indicates the number of voxels contained in this element.
- *Attributes: [NA]*
- *Child Elements: [NA]*
- *Text Content* – IJK indices (which are zero-based) of each voxel in this brain model or parcel, with each index separated by a whitespace character. There are three indices per voxel. If the parent element is BrainModel, then the BrainModel element's IndexCount attribute indicates the number of triplets (IJK indices) in this element's content.
- *Parent Elements* – **BrainModel**, **Parcel**

Parcel Element

- *Description* – Associates a name, plus vertices and/or voxels, with an index.
- *Attributes*
- **Name** – The name of the parcel
- *Child Elements*
- **Vertices** (0…N)
- **VoxelIndicesIJK** (0…1)
- *Text Content: [NA]*
- *Parent Element* – **MatrixIndicesMap**

Vertices Element

- *Description* – Contains a BrainStructure type and a list of vertex indices within a Parcel.
- *Attributes*
- **BrainStructure** – A string from the BrainStructure list to identify what surface this vertex list is from (usually left cortex, right cortex, or cerebellum).
- *Child Elements: [NA]*
- *Text Content* – Vertex indices (which are independent for each surface, and zero-based) separated by whitespace characters.

- *Parent Element* – **Parcel**

Surface Element

- *Description* – Specifies the number of vertices for a surface, when IndicesMapToDataType is "CIFTI_INDEX_TYPE_PARCELS." This is separate from the Parcel element because there can be multiple parcels on one surface, and one parcel may involve multiple surfaces.
- *Attributes*
- **BrainStructure** – A string from the BrainStructure list to identify what surface structure this element refers to (usually left cortex, right cortex, or cerebellum).
- **SurfaceNumberOfVertices** – The number of vertices that this structure's surface contains.
- *Child Elements: [NA]*
- *Text Content: [NA]*
- *Parent Element* – **MatrixIndicesMap**

Storage in NIfTI-2

A CIFTI-2 file is a one-file (.nii) NIfTI-2 file with the CIFTI XML placed into a NIfTI header extension, and a NIfTI intent code in the range 3000-3099. While NIfTI-1 volume files are sometimes compressed with GZIP, CIFTI files stored in NIfTI-2 should not be compressed, in order to allow efficient on-disk access. A CIFTI-2 file contains exactly one matrix. Future versions of CIFTI may contain more than one matrix.

The CIFTI XML is stored in a NIfTI header extension. NIFTI_ECODE_CIFTI is the name for the CIFTI extension code, and its value is 32.

The matrix is always stored in row-major order, meaning that each row of a matrix is in a contiguous section of the file. Thus, one can efficiently access a single row in a CIFTI-2 file.

Because the first four dimensions in NIfTI are reserved for space and time, the CIFTI dimensions are stored in the NIfTI header in dim[5] and up, where dim[5] is the length of the first CIFTI dimension (number of values in a row), dim[6] is the length of the second CIFTI dimension, and dim[7] is the length of the third CIFTI dimension, if applicable. The fields dim[1] through dim[4] will be 1; dim[0] will be 6 or 7, depending on whether a third matrix dimension exists.

The datatype field in the NIfTI header can be any of float, double, and signed or unsigned int8, int16, int32, or int64, with bitpix set correspondingly, per the NIfTI standard. Composite datatypes, such as complex or RGB, and bitwise storage are not supported in CIFTI. Multiple scalar maps, or separate files, can be used instead of composite datatypes. The most commonly used datatype is likely to be float (float32), but dense connectomes are a candidate for int8 storage to reduce their size, and dense label files may use int16. The scl_slope and scl_inter fields in the NIfTI header will be important for storage in a smaller datatype, in order to rescale the data to the range needed, so they should always be applied, as defined in the NIfTI standard.

Standard CIFTI Mapping Combinations

In order to more easily identify particular types of CIFTI file (the file type being a specific combination of mapping types), they are given separate filename extensions and intent codes. The different filename extensions make it easier to find a desired file when a directory contains multiple types of CIFTI file. A list of standard mapping type combinations and their associated NIfTI intent codes and filename extensions are provided below. Appendix A contains a table of the NIfTI intent codes and their corresponding intent names. However, it is permissible to use a combination of mapping types that is not in this list. When doing so, the NIfTI intent code should be the special code NIFTI_INTENT_CONNECTIVITY_UNKNOWN, which has a value of 3000, it should have an intent_name of "ConnUnknown" in the NIfTI-2 header, and the filename should have an extension of the form ".something.nii" , with "something" replaced by a unique string for the mapping combination, of the user's choice. If a standardized intent code and extension are desired for a particular combination of mappings, the CIFTI working group should be contacted (hcp-cifti@humanconnectome.org) so that it can be incorporated into the next revision of CIFTI.

Dense Connectivity

Intent_code: 3001, NIFTI_INTENT_CONNECTIVITY_DENSE
Intent_name: ConnDense
File extension: .dconn.nii
AppliesToMatrixDimension 0: brain models
AppliesToMatrixDimension 1: brain models

This file type represents connectivity between points in the brain. A row is the connectivity from a single vertex or voxel in the mapping that applies along the second dimension, to all vertices and voxels along the first dimension. This specification of "from" and "to" is not intended to imply that the data is always directed, but to establish a convention so that directed data is stored consistently, and to ensure that interactive usage loads rows from the matrix, in order to maximize responsiveness. Note that this type can have a single mapping apply to both dimensions, but can also have separate, different mappings for rows and columns, for instance only containing connectivity from left cortex to cerebellum.

Dense Data Series

Intent_code: 3002, NIFTI_INTENT_CONNECTIVITY_DENSE_SERIES
Intent_name: ConnDenseSeries
File extension: .dtseries.nii
AppliesToMatrixDimension 0: series
AppliesToMatrixDimension 1: brain models

This file type represents data points in a series for every vertex and voxel in the mapping. A row is a complete data series, for a single vertex or voxel in the mapping that applies along the second dimension. A dataseries is often a timeseries, but it can also represent other data types such as a series of sampling depths along the surface normal from the white to pial surface. It retains the 't' in dtseries from CIFTI-1 naming conventions.

Parcellated Connectivity

Intent_code: 3003, NIFTI_INTENT_CONNECTIVITY_PARCELLATED
Intent_name: ConnParcels
File extension: .pconn.nii

AppliesToMatrixDimension 0: parcels
AppliesToMatrixDimension 1: parcels

This file type represents connectivity between areas or parcels of the brain. Similarly to Dense Connectivity, a row is the connectivity from a single parcel in the mapping that applies along the second dimension, to all parcels along the first dimension. Note that this type can have a single mapping apply to both dimensions, but can also be asymmetric, for example if a parcellated connection was from a subset of cortical areas to all of them (e.g. injection sites for tracer data).

Parcellated Data Series

Intent_code: 3004, NIFTI_INTENT_CONNECTIVITY_PARCELLATED_SERIES
Intent_name: ConnParcelSries (due to length constraints of the NIfTI header field, the first "e" in "series" is removed)
File extension: .ptseries.nii
AppliesToMatrixDimension 0: series
AppliesToMatrixDimension 1: parcels

This file type represents data points in a series for areas of the brain. Similarly to Dense Data Series, a row is a complete data series (e.g. ICA component timeseries), for a single parcel in the mapping that applies along the second dimension. This is called ptseries by analogy with dtseries.

Dense Scalar

Intent_code: 3006, NIFTI_INTENT_CONNECTIVITY_DENSE_SCALARS
Intent_name: ConnDenseScalar
File extension: .dscalar.nii, others - see "Specializations of Scalar Maps"
AppliesToMatrixDimension 0: scalars
AppliesToMatrixDimension 1: brain models

This file type stores named scalar maps across vertices and voxels, like a GIFTI "functional" file (aka a metric file) that can accommodate multiple surfaces, and voxels. It is often used to store task fMRI activity maps in the standard grayordinates space or myelin maps from both cortical hemispheres. A row is a single value from each named map, for one vertex or voxel.

Dense Label

Intent_code: 3007, NIFTI_INTENT_CONNECTIVITY_DENSE_LABELS
Intent_name: ConnDenseLabel
File extension: .dlabel.nii
AppliesToMatrixDimension 0: labels
AppliesToMatrixDimension 1: brain models

This file type stores named label maps across vertices and voxels, similarly to Dense Scalar, but with label keys in the data matrix instead of continuous-valued data. Each label map has its own separate label table defined within the CIFTI XML. A row is a single label value from each named label map, for one vertex or voxel.

Parcellated Scalar

Intent_code: 3008, NIFTI_INTENT_CONNECTIVITY_PARCELLATED_SCALAR
Intent_name: ConnParcelScalr (due to length constraints of the NIfTI header field, the last “a” is removed)
File extension: .pscalar.nii
AppliesToMatrixDimension 0: scalars
AppliesToMatrixDimension 1: parcels

This file type stores named scalar maps across parcels. A row is a single scalar from each map, for one parcel. For example, it could store task activity measures in parcels.

Parcellated Dense Connectivity

Intent_code: 3009, NIFTI_INTENT_CONNECTIVITY_PARCELLATED_DENSE
Intent_name: ConnParcelDense
File extension: .pdconn.nii
AppliesToMatrixDimension 0: brain models
AppliesToMatrixDimension 1: parcels

This file type stores connectivity from parcels to vertices and/or voxels. A row is the connectivity from one parcel to all vertices and voxels.

Dense Parcellated Connectivity

Intent_code: 3010, NIFTI_INTENT_CONNECTIVITY_DENSE_PARCELLATED
Intent_name: ConnDenseParcel
File extension: .dpconn.nii
AppliesToMatrixDimension 0: parcels
AppliesToMatrixDimension 1: brain models

This file type stores connectivity from vertices and voxels to parcels. A row is the connectivity from one vertex or voxel to all parcels.

Parcellated Connectivity Series

Intent_code: 3011,
NIFTI_INTENT_CONNECTIVITY_PARCELLATED_PARCELLATED_SERIES
Intent_name: ConnPPSr
File extension: .pconnseries.nii
AppliesToMatrixDimension 0: parcels
AppliesToMatrixDimension 1: parcels
AppliesToMatrixDimension 2: series

This file type stores connectivity between parcels at regular samples in a series, such as timepoints or frequency. A row is the connectivity from one parcel in the mapping that applies along the second dimension to all parcels in the first dimension at one sample in the series.

Parcellated Connectivity Scalar

Intent_code: 3012,
NIFTI_INTENT_CONNECTIVITY_PARCELLATED_PARCELLATED_SCALAR
Intent_name: ConnPPSc

File extension: .pconnscalar.nii
AppliesToMatrixDimension 0: parcels
AppliesToMatrixDimension 1: parcels
AppliesToMatrixDimension 2: scalars

This file type stores connectivity between parcels under named conditions (e.g. frequency bands, different subjects). A row is the connectivity from one parcel in the mapping that applies along the second dimension to all parcels in the first dimension at one named condition (e.g. frequency bands) in the third dimension.

Specializations of Scalar Maps

The scalar mapping type lends itself well to arbitrary data, including data that has some structure and specific interpretation. While the normal extension for a dense scalar file is ".dscalar.nii", it is also used for storing data used in a more specialized way, notably for tractography fiber orientations. This specialization has the same intent code and mapping types, but a different extension, and a well-defined map layout with a specific meaning associated with each map (for instance, the theta angle of a fiber sample). Other specializations of CIFTI files make use of two scalar dimensions in a similar way.

Dense Fiber Fans

Intent_code: 3002, NIFTI_INTENT_CONNECTIVITY_DENSE_SERIES
Intent_name: ConnDenseSeries
File Extension: .dfan.nii
AppliesToMatrixDimension 0: scalars
AppliesToMatrixDimension 1: brain models
Map Layout:

For brainordinate index B:
- Index (0, B): X coordinate
- Index (1, B): Y coordinate
- Index (2, B): Z coordinate
- For fiber F, starting at 0:
  - Index (7 * F + 3, B): Mean of fiber strength
  - Index (7 * F + 4, B): Standard deviation of fiber strength
  - Index (7 * F + 5, B): Theta angle of fiber bingham
  - Index (7 * F + 6, B): Phi angle of fiber bingham
  - Index (7 * F + 7, B): Dispersion parameter ka of fiber bingham
  - Index (7 * F + 8, B): Dispersion parameter kb of fiber bingham
  - Index (7 * F + 9, B): Psi angle of fiber bingham

This file is used to store fiber fan bingham models (either of fiber uncertainty—e.g. from bedpost—or fiber fanning), for the purpose of displaying probabilistic tractography data (Sotiropoulos et al., 2012).

Dense Fiber Samples

Intent_code: 3000, NIFTI_INTENT_CONNECTIVITY_UNKNOWN
Intent_name: ConnUnknown
File Extension: .dfibersamp.nii

AppliesToMatrixDimension 0: scalars
AppliesToMatrixDimension 1: scalars
AppliesToMatrixDimension 2: brain models
Map Layout:

For brainordinate index B:
    Index (0, 0, B): X coordinate
    Index (1, 0, B): Y coordinate
    Index (2, 0, B): Z coordinate
    Other indexes in (x, 0, B) are unused
    For sample S, starting at 0:
        For fiber F, starting at 0:
            Index (3 * F, S + 1, B): Fiber strength
            Index (3 * F + 1, S + 1, B): Theta angle of fiber
            Index (3 * F + 2, S + 1, B): Phi angle of fiber

This file is used to store samples of fiber orientations, in order to display the directions for a voxel as points on a sphere.

Dense Fan Samples

Intent_code: 3000, NIFTI_INTENT_CONNECTIVITY_UNKNOWN
Intent_name: ConnUnknown
File Extension: .dfansamp.nii
AppliesToMatrixDimension 0: scalars
AppliesToMatrixDimension 1: scalars
AppliesToMatrixDimension 2: brain models
Map Layout:

For brainordinate index B:
    Index (0, 0, B): X coordinate
    Index (1, 0, B): Y coordinate
    Index (2, 0, B): Z coordinate
    Other indexes in (x, 0, B) are unused
    For sample S, starting at 0:
        For fiber F, starting at 0:
            Index (6 * F, S + 1, B): Fan strength
            Index (6 * F + 1, S + 1, B): Theta angle of fan bingham
            Index (6 * F + 2, S + 1, B): Phi angle of fan
            Index (6 * F + 3, S + 1, B): Dispersion parameter ka of fan bingham
            Index (6 * F + 4, S + 1, B): Dispersion parameter kb of fan bingham
            Index (6 * F + 5, S + 1, B): Psi angle of fan bingham

This file is used for a similar purpose as dense fiber samples, but is used when fiber fanning is modeled directly.

REFERENCES

Glasser MF, Sotiropoulos SN, Wilson JA, Coalson TS, Fischl B, Andersson JL, Xu J, Jbabdi S, Webster M, Polimeni JR, Van Essen DC, Jenkinson M; WU-Minn HCP Consortium (2013). Neuroimage 80:105-124.

Sotiropoulos, S. N., Behrens, T. E., & Jbabdi, S. (2012). Ball and rackets: inferring fiber fanning from diffusion-weighted MRI. *Neuroimage*, *60*, 1412-1425.