hClustering custom file formats

This file contains information about the most important custom file formats implemented in the *hClustering* algorithm framework. All this files are automatically generated and recognized by the different *hClustering* commands and algorithms.

- **.cmpct** compact tractogram format (used in nifti mode).

In the *hClustering* algorithms 3D tractograms are compacted into 1D vectors containing only the voxels defined in the white matter mask used for tracking. This way computational time is saved when doing dissimilarity operations on those vectors, as all the voxels not belonging to the white matter (and which value would always be 0 in a tractogram) are removed.

These 1D vectors (called compact tractograms hereby on) can become very large in case of using high resolution data (around 800.000 data points for a 1 mm3 dataset). Unfortunately the nifti format can only handle dimensions up to 65.535 data points. To overcome this shortcoming, a very simple format was implemented to handle 1D compact tractograms.

The **.cmpct** format is a binary file with 3 fields: type, size, and data.

- The **type** field is a uint32_t unsigned integer (32 bits) which can have one of two values: 8 if the data is represented in uint8 (8-bit unsigned integer) or 32 if the data is represented in float32 (32-bit floating point).

- The **size** field is a uint32_t unsigned integer (32 bits) which defines the number of data points that will be contained in the data field.

- The **data** field contains the 1D vector data points, as many data points as defined by the size field, with the datatype defined by the type field.

Type (32 bits) Unsigned integer	Size (32 bits) Unsigned integer	Data (either Uint8 or Float32, defined by Type field) Number of data points defined by Size field

The .cmpct binary file format

- **roi.txt** file format.

The roi file is an ASCII file with the relevant information regarding seed voxels coordinates, and the tractograms obtained from these seeds. It contains several fields delimited by flags of the type:

```
#flag
data of field defined by ‘flag’
#endflag
```

below are shown the possible flag labels and the format and meaning of the data contained in the respective fields:

#imagesize:
contains a set of three values indicating the total size in voxels of the x, y and z dimensions of the diffusion 3D image used for tracking followed by a string indicating the reference frame used for the seed voxel coordinates, these can be “vista”, “nifti” or “surf” (for *freesurfer* RAS coordinates).

#streams:
Contains a single integer indicating the number of streamlines or particles generated during the creation of a single probabilistic tractogram.

#trackindex:
Consists of a list of the tractogram IDs corresponding to the seed coordinates contained in the following section. Each ID is written in a separate line, and an ID in a certain position in the list will match the coordinate found in the same position of the corresponding coordinate list. This field is important when tractogram files are named by tractogram ID, in cases like *vdconnect* vista tracking, where tract files are named by coordinates, this field might be missing.

#roi:
A list of the seed coordinates forming the ROI from which the tractograms where generated (one tract per seed voxel). Each coordinate consists of three integers separated with spaces and each coordinate is written in a new line. The position of a coordinate in the list defines that seed’s ID (also called leaves in the context of tree building).

- **tree.txt** file format.

The tree file is an ASCII file with the relevant information regarding the seed voxels the tree was built from, the node distance value and hierarchical information, and additional fields such as partitions and discarded seeds. It contains several fields delimited by flags of the type:

#flag
data of field defined by ‘flag’
#endflag

below are shown the possible flag labels and the format and meaning of the data contained in the respective fields:

#imagesize:
contains a set of three values indicating the total size in voxels of the x, y and z dimensions of the diffusion 3D image used for tracking followed by a string indicating the reference frame used for the seed voxel coordinates, these can be “vista”, “nifti” or “surf” (for *freesurfer* RAS coordinates).

#streams:
Contains a single integer indicating the number of streamlines or particles generated during the creation of a single probabilistic tractogram.

#logfactor:
If the tractograms where normalized using logarithmic transformation, these field defines the value that must be used as logarithmic factor in order to transform a tractogram from normalized units to natural units and vice versa.

#coordinates:
A list of the seed coordinates forming the ROI from which the tractograms where generated (one tract per seed voxel). Each coordinate consists of three integers separated with spaces and each coordinate is written in a new line. The position of a coordinate in the list defines that seed’s ID (called leaves in the context of tree building).

#trackindex:
Consists of a list of the tractogram IDs corresponding to the seed coordinates contained in the previous section. Each ID is written in a separate line, and an ID in a certain position in the list will match the coordinate found in the same position of the corresponding coordinate list. This field is important when tractogram files are named by tractogram ID, in cases like *vdconnect* vista tracking, where tract files are named by coordinates, this field might be missing.

#clusters:
Each line represents a node in the tree, and the position in the list defines the ID of the node. The first value of the line is a floating point defining the distance value of that node. Following are a set of Boolean-integer pairs separated with spaces each one defining a leaf or node that was merged at this point to form the node. If the Boolean value is 0, the accompanying integer value is the ID of a leaf, if the Boolean is a 1, the ID corresponds to a node.

#discarded:
This field contains the coordinates of seed voxels that were discarded during the tree building process (either for being regarded as outliers or for being voxels unconnected to the rest of the roi).

#cpcc:
When present, indicates the tree cophenetic correlation coefficient has been calculated, and is equal to the single floating point value contained in this field.

#partvalues and **#partitions:**
This fields come always followed by one another and are present if partitions where selected or computed and saved into the tree file, the must contain the same number of lines, each corresponding to the information of one partition. #partvalue saves the partition selection value that defined that selection (might be distance value if it was a horizontal partition, or SS index value for the best quality partition selection. #partitions lines are a set of integers corresponding to the node IDs that constitute each partition.

#partcolors:
This field only appears if #partvalues and #partitions are also present in the file, and it means that predefined colors where saved for each cluster of each saved partition in order to be loaded in the *OpenWalnut* hierarchical tree exploration module. It contains a line for each partition and on each line a triple of integers for each cluster in the partition, defining the R, G, and B values corresponding to each cluster. The values within each triple are separated by whitespaces, while the triples themselves are separated by semicolons.

- **baselist.txt** file format.

The baselist file is an ASCII file with the relevant information regarding meta-laves (or base-nodes) created during the tree building process. It contains several fields delimited by flags of the type:

#flag
data of field defined by ‘flag’
#endflag

Below are shown the possible flag labels and the format and meaning of the data contained in the respective fields:

#bases
Contains a list of the node-IDs that form the meta-leaves set for that tree, each ID is written in a separate line

#pruned
Contains a list of leaf-IDs of seeds that might have been pruned from the tree during tree processing operations.

- Distance matrix **roi_index.txt** format.

An ASCII file with a single field (**#distindex**) matching each seed voxel coordinate to a distance matrix row/column block-ID and index position within the block, so that given two coordinates the exact block and position for the distance value between them may be pinpointed.

Each line has the following structure:

X Y Z   b   BLOCK   i   INDEX

where X Y Z are the integer voxel coordinates of a particular seed, BLOCK is an integer defining the block ID and INDEX an integer defining the position within that block. As an example a few lines of a real roi_index.txt file would look like this:

```
052 054 052 b 003 i 1236
044 055 052 b 003 i 1237
048 055 052 b 003 i 1238
049 055 052 b 003 i 1239
050 055 052 b 003 i 1240
```