

Program name: acpcdetect

Purpose: This program is a part of the *automatic registration toolbox* (ART) package. It is designed to automatically detect the locations of the anterior commissure (AC), the posterior commissure (PC), and the vertex of the superior pontine sulcus (VSPS) on 3D structural MRI volumes. The program is model-based and can be trained to operate on different types of image contrasts. A model (T1acpc.mdl) is distributed with the program for detecting these landmarks on T_1 -weighted scans. Models for other contrasts (e.g., T_2 -weighted) may also be created.

The program takes a NIFTI image of type *short* or *unsigned char* as input and writes the coordinates of the detected AC, PC, and VSPS landmarks to an output text file. The program also reports a mid-sagittal plane (MSP), adjusted to pass through the AC, PC, and VSPS, in the form: $ax + by + cz = d$.

Usage: acpcdetect [-h -v -D -m *model* -O *code* -rvsps *r* -rac *r* -rpc *r* -T2 -o *outputimage.nii* -oo *code* -onx *nx* -ony *ny* -onz *nz* -odx *dx* -ody *dy* -odz *dz*] -i *image.nii*

Required argument

-i or -image *image.nii*: Input image in NIFTI format. Must be of type *short* (*data type 4*) or *unsigned char* (*data type 2*).

Optional arguments

-h or -help: Prints help message

-v or -verbose: Enables verbose mode

-D: Prints additional information

-m or -model *model*: User-specified model. If this option is not specified at the command line, the program uses the model \$ARTHOME/T1acpc.mdl by default. Note that the user must set the environment variable \$ARTHOME to the directory that contains the model file T1acpc.mdl, which is distributed with the package.

-O or -orient *code*: Three-letter image orientation code (case-insensitive). This option overrides the image orientation information present in the NIFTI header. This option is useful in cases where the orientation information has not been properly recorded in the NIFTI header. There are 48 distinct possible orientation codes consisting of combinations of letters A, P, L, R, S, and I denoting, anterior, posterior, left, right, superior, and inferior directions, respectively.

Examples:

PIL for Posterior-Inferior-Left

RAS for Right-Anterior-Superior

See below for more details on how these orientation codes are defined.

-rvsps *r*: Radius of the search region for the VSPS in mm (default = 50 mm)

-rac *r*: Radius of the search region for the AC in mm (default = 15 mm)

-rpc *r*: Radius of the search region for the PC in mm (default = 15 mm)

-T2: Use this option if the input image is T_2 -weighted.

-o *outputimage.nii*: If this option is present, then the program outputs an “AC/PC aligned” version of the input image.

- oo code: Specifies an orientation code for the output image (default: same as the input image).
- onx nx, -ony ny, -onz nz: Specify the matrix dimensions for the output image (default: same as the input image).
- odx dx, -ody dy, -odz dz: Specify the voxel dimensions for the output image (default: same as the input image).

Program outputs:

image_ACPC_sagittal.ppm: Sagittal view of the detected AC (green), PC (red), and VSPS (blue) landmarks (in PPM format). Cross-sections of the corresponding search regions (circles) are also shown.

image_ACPC_axial.ppm: Axial view of the detected AC/PC locations (in PPM format). The intersection of the MSP (white line) is also outlined on this image.

image_ACPC.txt: Stores the detected landmark coordinates in voxel units in the *ijk* coordinate system (see below), and the mid-sagittal plane parameters with respect to the *xyz* coordinate system (see below).

Example: `acpcdetect -v -i ba.nii`

Image volume “ba.nii” is a 3D T_1 -weighted MRI scan with PIL orientation distributed with this package. The program outputs three files: “ba_ACPC_sagittal.ppm”, “ba_ACPC_axial.ppm”, and “ba_ACPC.txt.”

The user is also encouraged to run the algorithm on the NIFTI images `avg152T1_LR_nifti.nii` and `avg152T1_RL_nifti.nii` available online at: <http://nifti.nimh.nih.gov/nifti-1/data>.

Characteristics of the AC/PC-aligned output image: If the `-o outputimage.nii` option is present, then the program writes an “AC/PC aligned” version of the input image to outputimage.nii. In the output image, the AC would be located at the center of the FOV; the AC-PC line would coincide with one of the three image axes (for example, in a LPS image, the AC-PC line would be parallel to the y-axis, while for a PIL image, it would be parallel to the x-axis); and the MSP will coincide with one of the three image planes (for example, for a LIA image, the MSP would coincide on the yz-plane). The default orientation, matrix size, and voxel size of the output image will be the same as the input image. However, the default orientation can be overridden using `-oo code` option; the default matrix dimensions can be overridden using the `-onx nx`, `-ony ny`, and `-onz nz` options; and the default voxel dimensions may be overridden using the `-odx dx`, `-ody dy`, and `-odz dz` options.

Coordinate systems: Let n_x , n_y , and n_z represent the number of image columns, rows, and slices, respectively. The landmark locations are reported in the *ijk* coordinate system in voxel units. In this system, i , j , and k vary from 0 to $(n_x - 1)$, 0 to $(n_y - 1)$, and 0 to $(n_z - 1)$, respectively. **It is assumed that the voxels are stored in the image file in such a way that i varies the fastest, followed by j , and k .**

Note that the coordinate values under this system can be non-integers. If the program indicates, for example, $i = 1.5$, this refers to a point between the second and third image columns.

The equation for the MSP is reported in the *xyz* coordinate system in mm units. The relationship between the *ijk* and *xyz* systems are given as follows:

$$\begin{aligned}
 x &= i \times d_x - d_x \times (n_x - 1)/2 \\
 y &= j \times d_y - d_y \times (n_y - 1)/2
 \end{aligned}$$

$$z = k \times d_z - d_z \times (n_z - 1)/2$$

where d_x , d_y , and d_z are the voxel dimensions in mm in x , y , and z directions, respectively.

Image orientation codes: In a properly defined NIFTI file, the program automatically determines the input image orientation. However, in cases where this information does not exist or is inconsistent with the actual image orientation, the user can explicitly define the image orientation using the `-0` or `-orient` argument.

The three-letter image orientation codes indicate the directions towards which the ijk or equivalently the xyz axes point. For example, the orientation code PIL indicates that the x -direction of the input image volume is aligned with the subject's anterior-to-posterior direction (points towards the posterior direction), the y -direction of the image is aligned with the subject's superior-to-inferior direction (points towards the inferior direction), and finally the z -direction of the image, that is, the direction of increasing slice number, is aligned to the subject's right-to-left direction (points towards the subject's left). As another example, LIA indicates that the x , y , and z coordinates point towards, left, inferior, and anterior directions, respectively.

Reference: Ardekani, B.A. and Bachman, A.H., Model-based automatic detection of the anterior and posterior commissures on MRI scans. *Neuroimage*, 2009, 46(3):677–82.

Program and documentation authors: Babak A. Ardekani, Ph.D. and Alvin H. Bachman, Ph.D., Center for Advanced Brain Imaging, The Nathan S. Kline Institute for Psychiatric Research.
February 16, 2011.