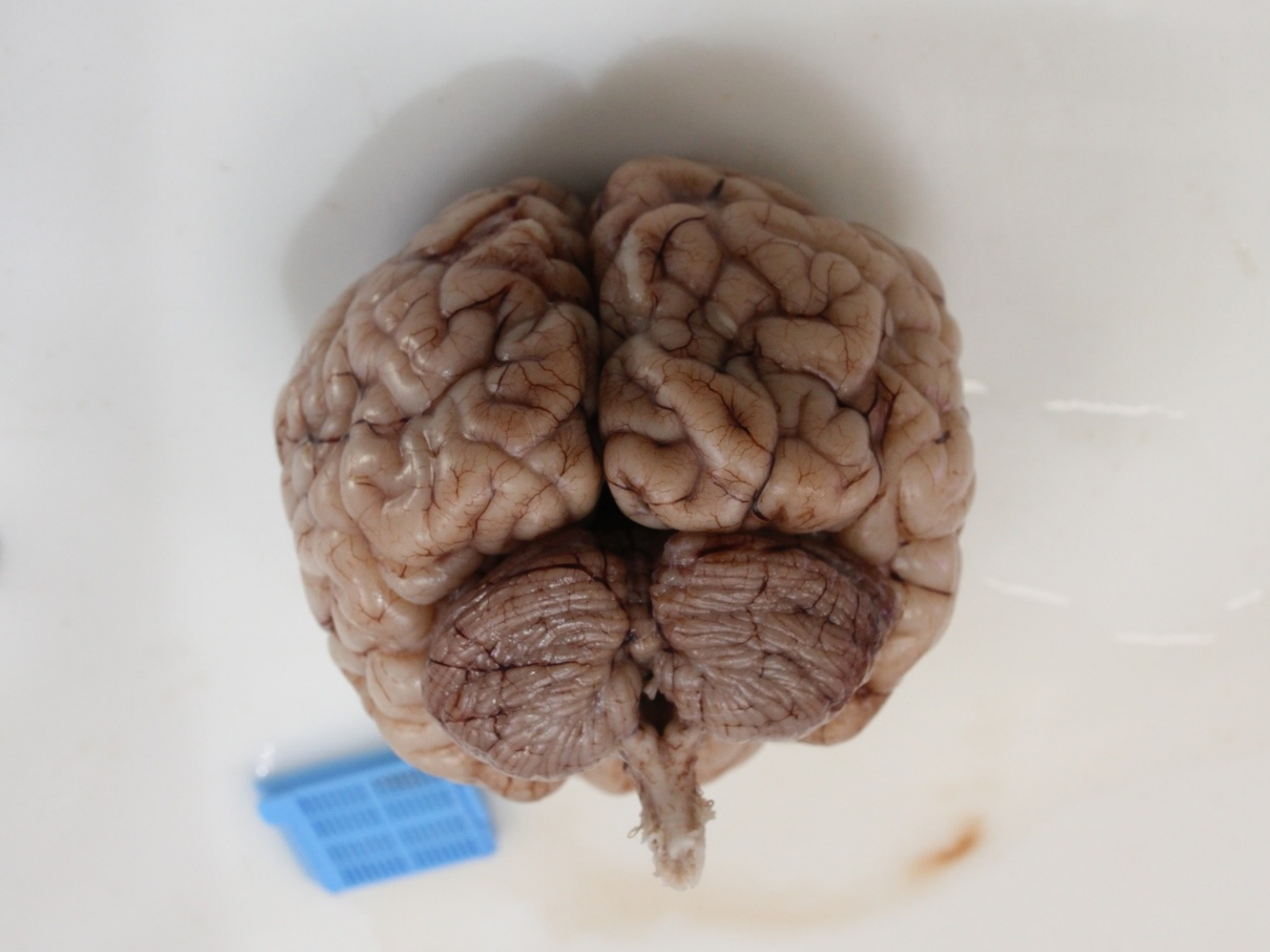
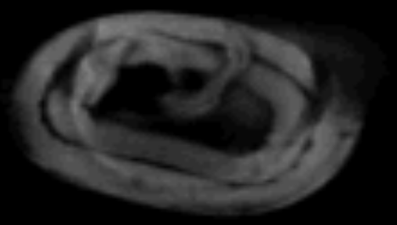
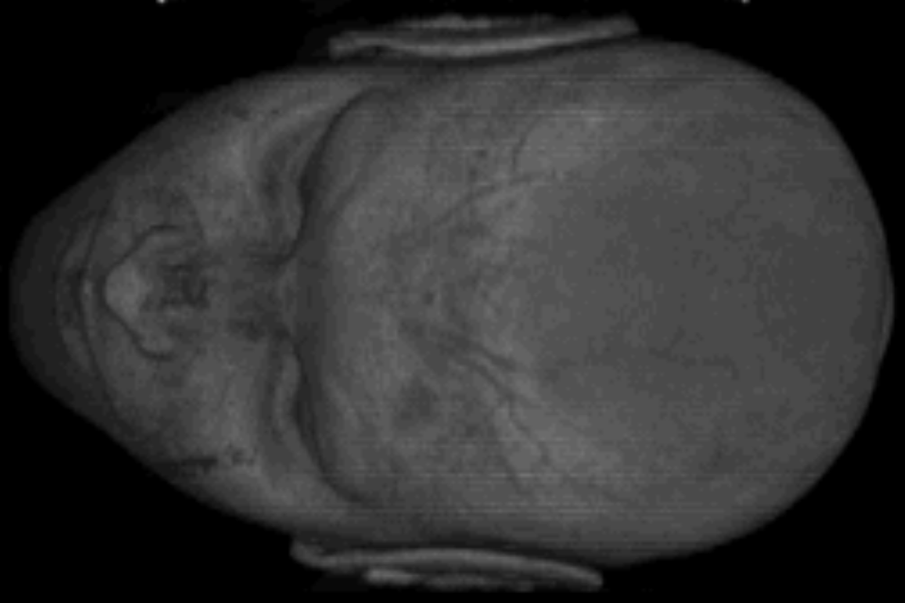
Mindboggle

SeaPIG 1/12/2015
Arno Klein
arno@binarybottle.com

Sage BIONETWORKS







Mindboggle

Welcome to Mindboggle!

We are developing Mindboggle to improve the accuracy, precision, and reliability of automated labeling and shape analysis of human brain image data, and to promote open science by making all data, software, and documentation freely and openly available.

We are gearing up for our first official release!

Software

Please see the README for Mindboggle's open source, freely available python software for automated labeling, feature extraction, and shape analysis of human brain image data.

Data

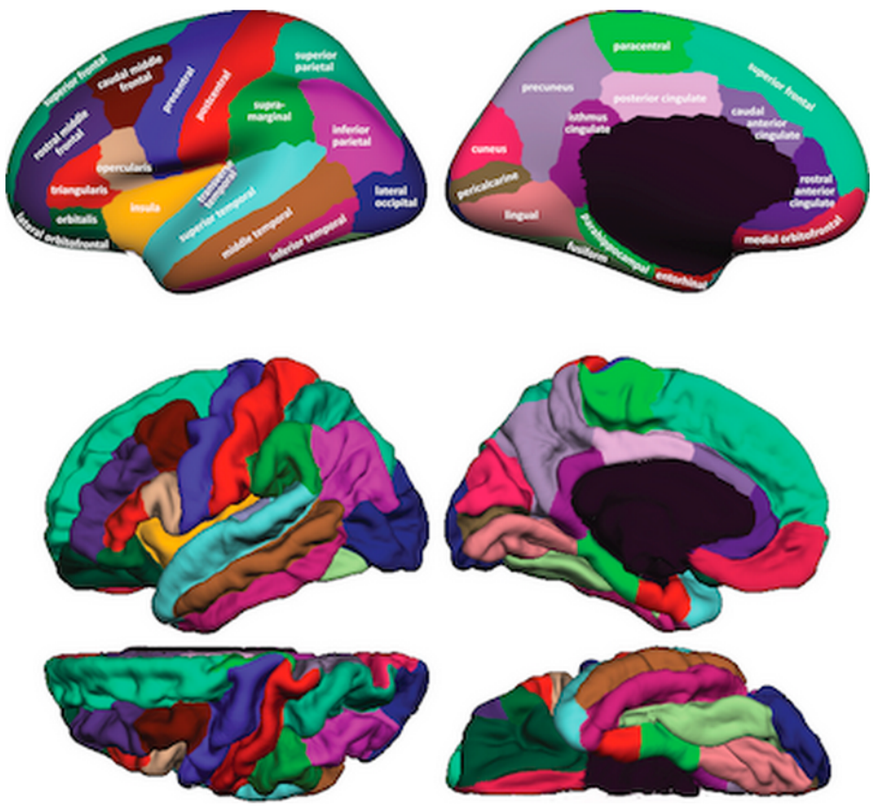
Feel free to download Mindboggle-101 atlases, templates, and individual labeled brain image data, the largest collection of publicly available, manually labeled human brains in the world!

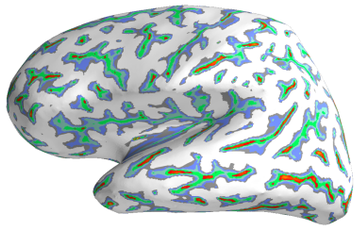
Mindboggle

Data

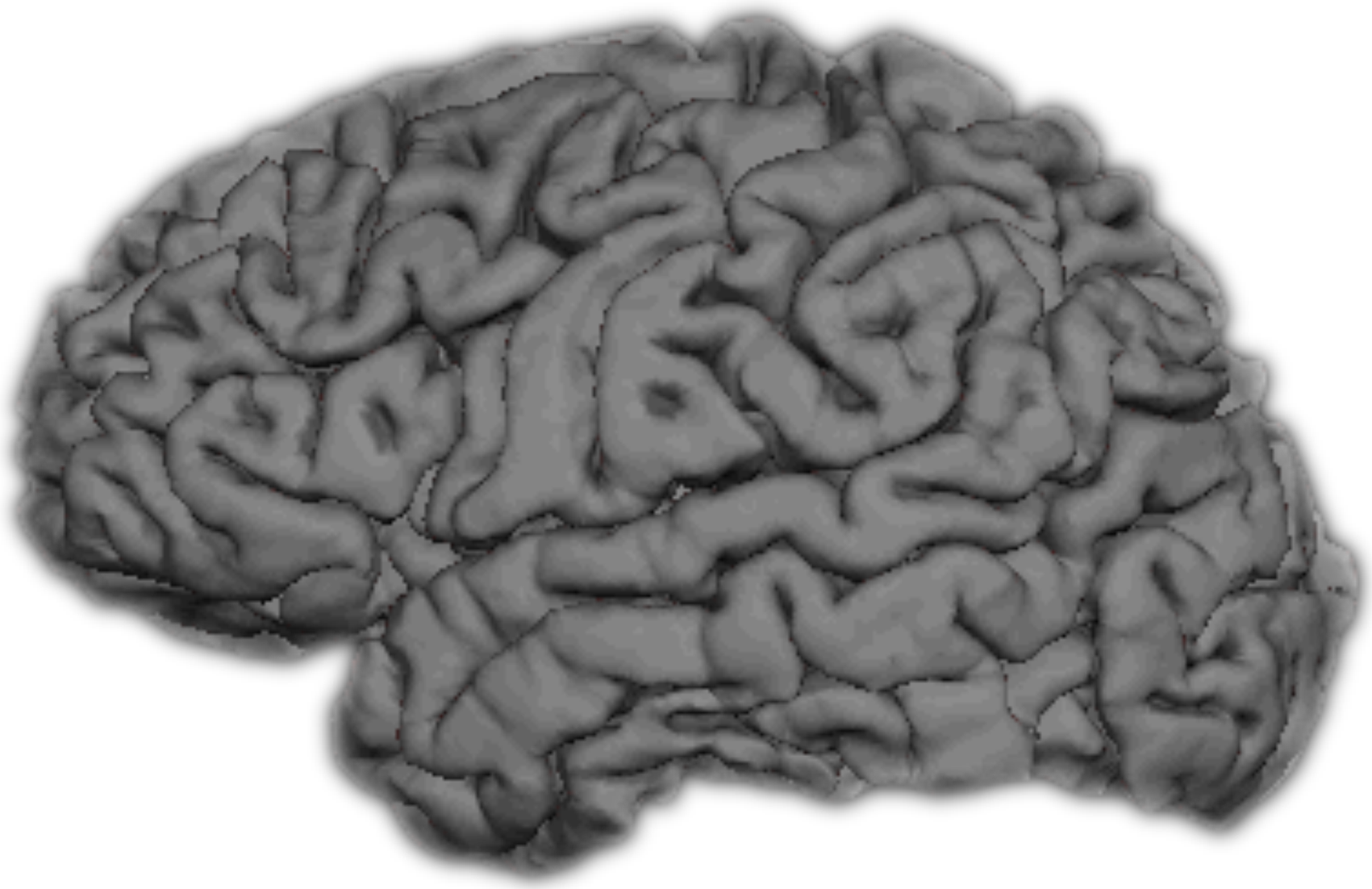
Welcome to the world's largest collection of free, manually labeled human brain image data!
Please cite the following article and this website when making use of Mindboggle-101 data:
101 labeled brain images and a consistent human cortical labeling protocol
Arno Klein, Jason Tourville. Frontiers in Brain Imaging Methods. 6:171. DOI: 10.3389/fnins.2012.00171

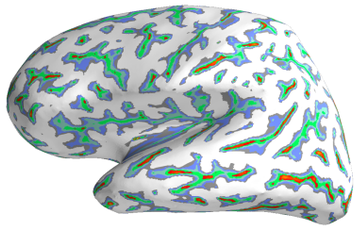
See the README, subjects table, scan info, labels, the CHANGELOG, and MD5SUMS,
which describe the labeled nifti volumes (nii), vtk surfaces (vtk), and FreeSurfer files (mgh, etc.).
Except where noted, all data are licensed under a Creative Commons License: (cc) BY-NC-SA



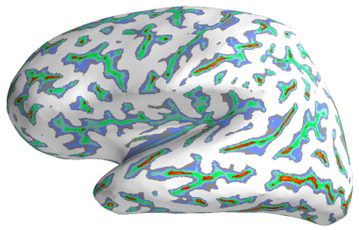


Cortical surface

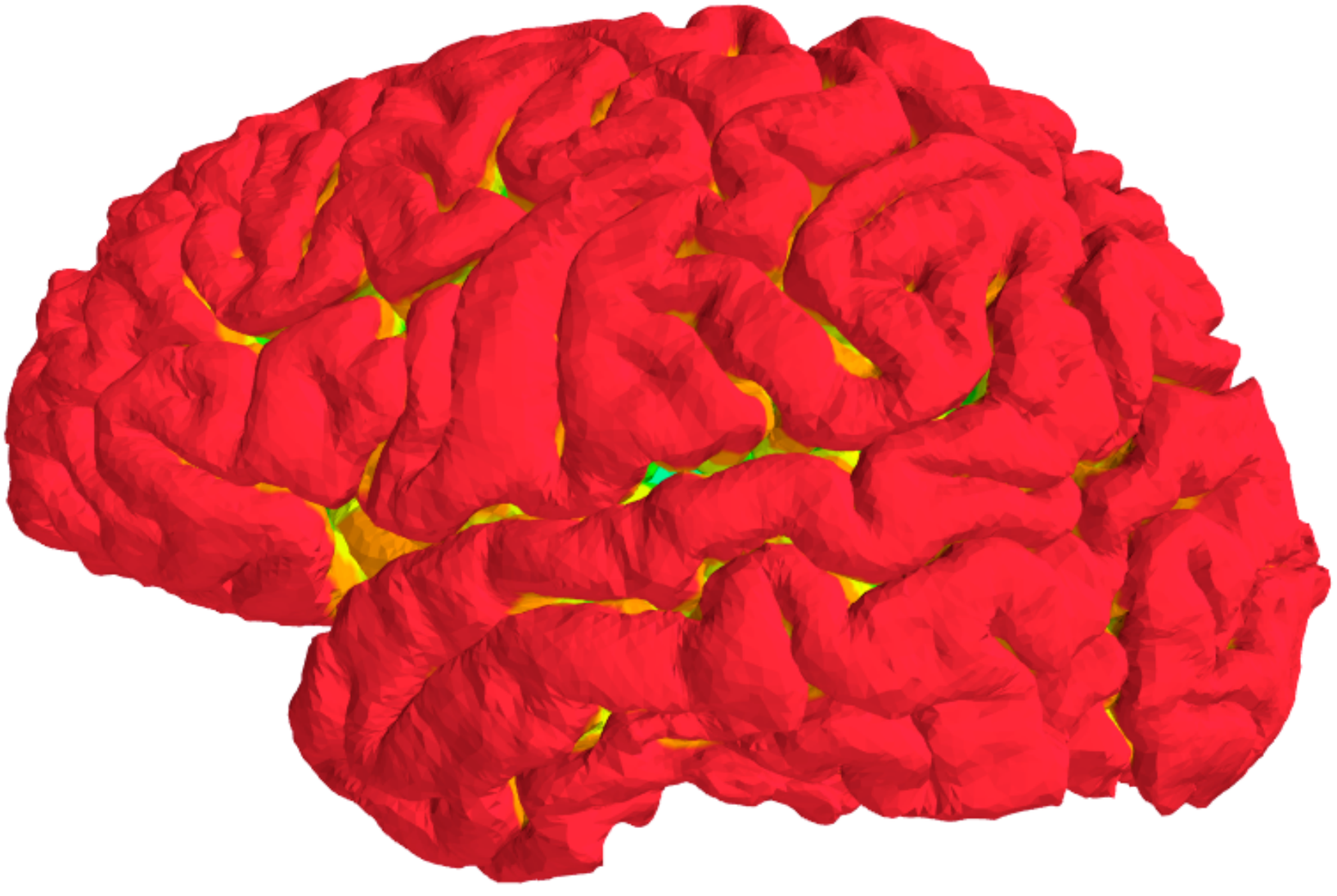


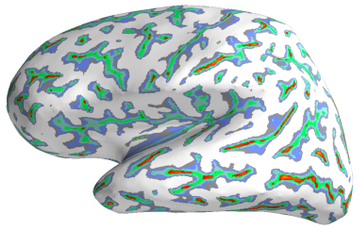


Mean curvature

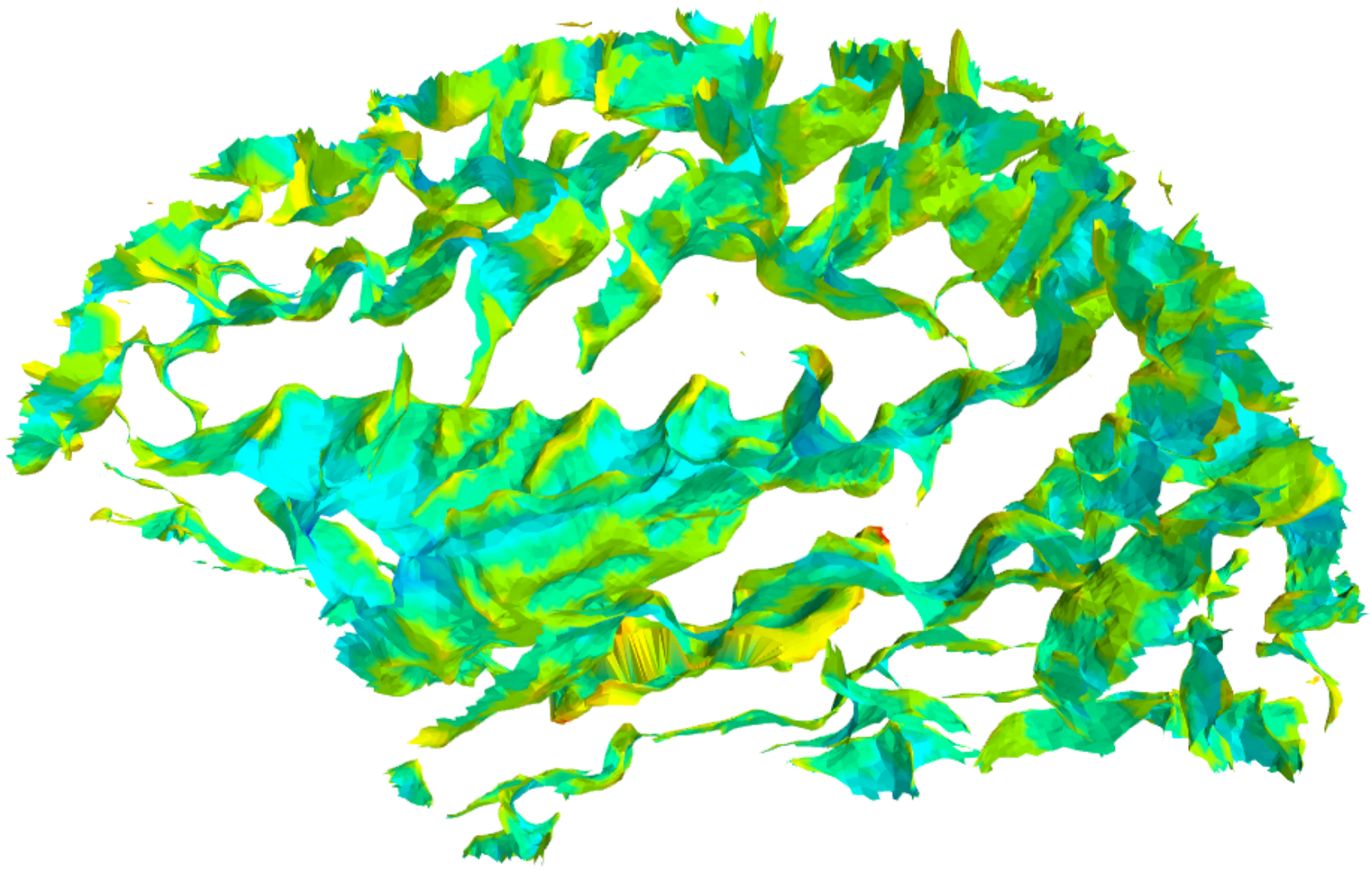


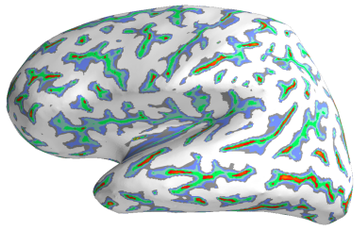
Travel depth



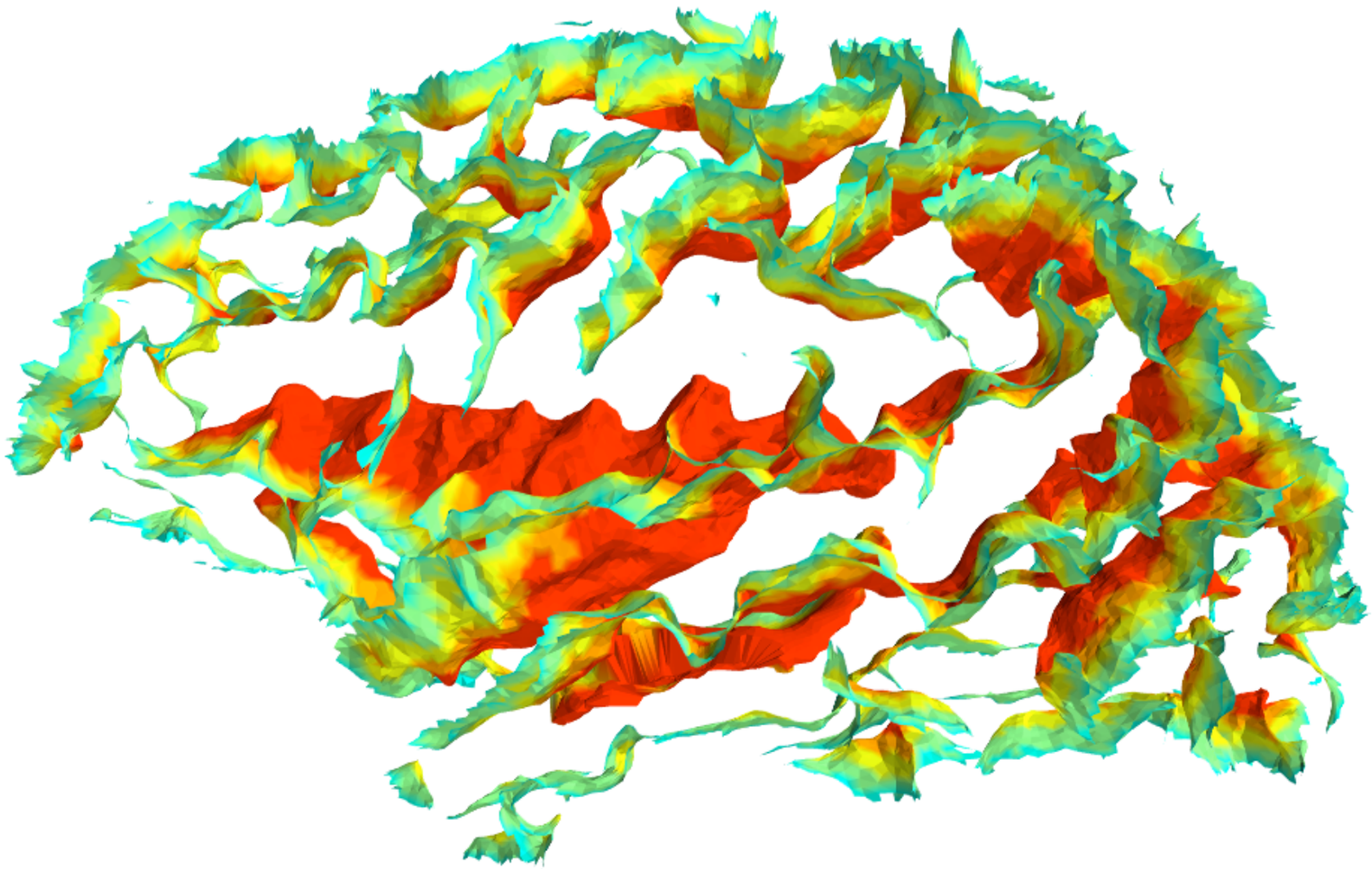


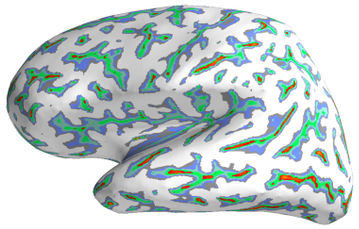
Mean curvature (folds)



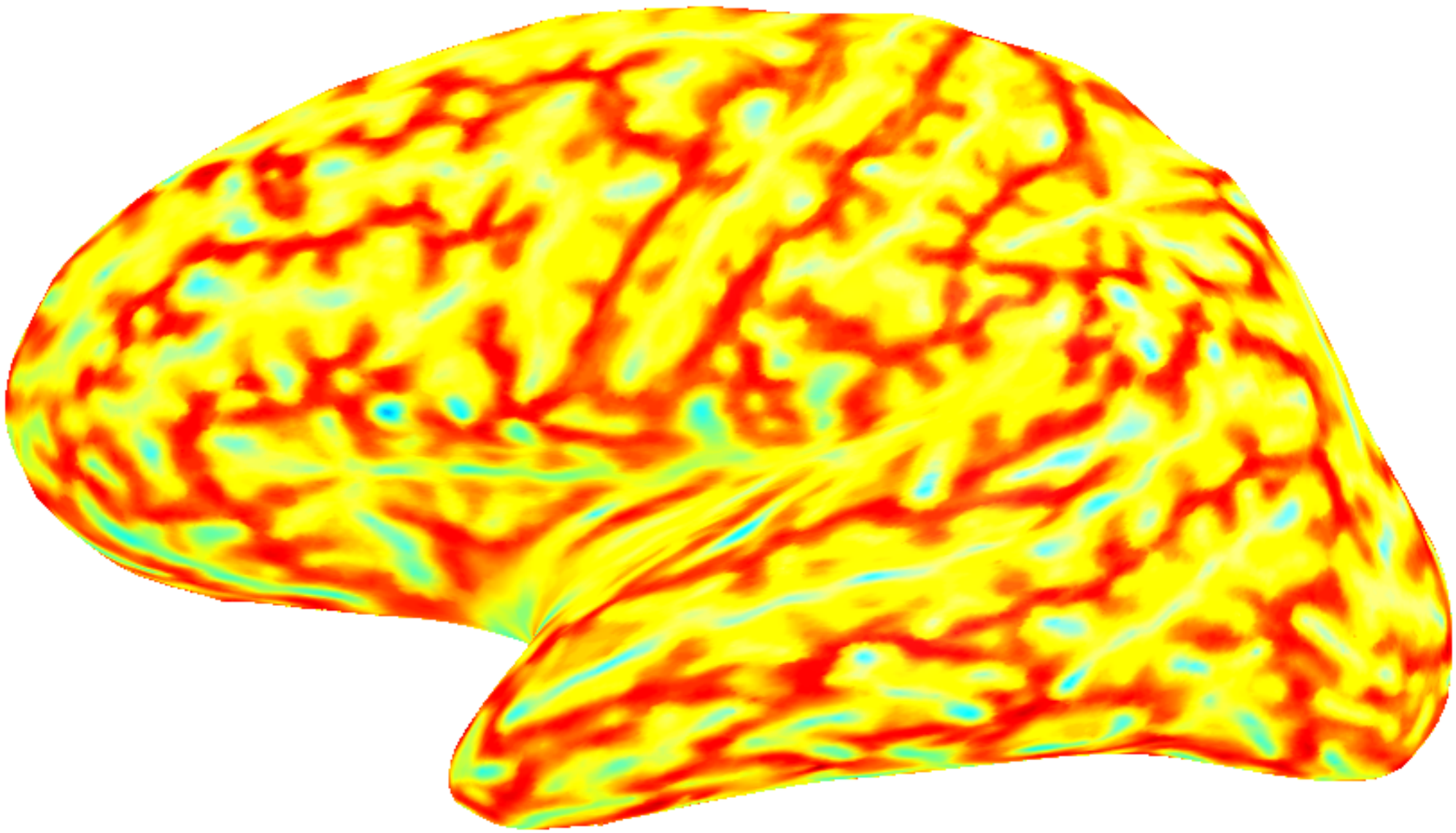


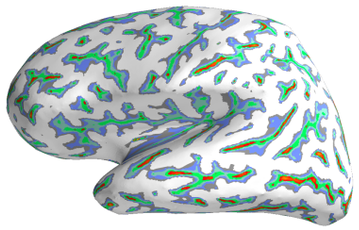
Travel depth (folds)



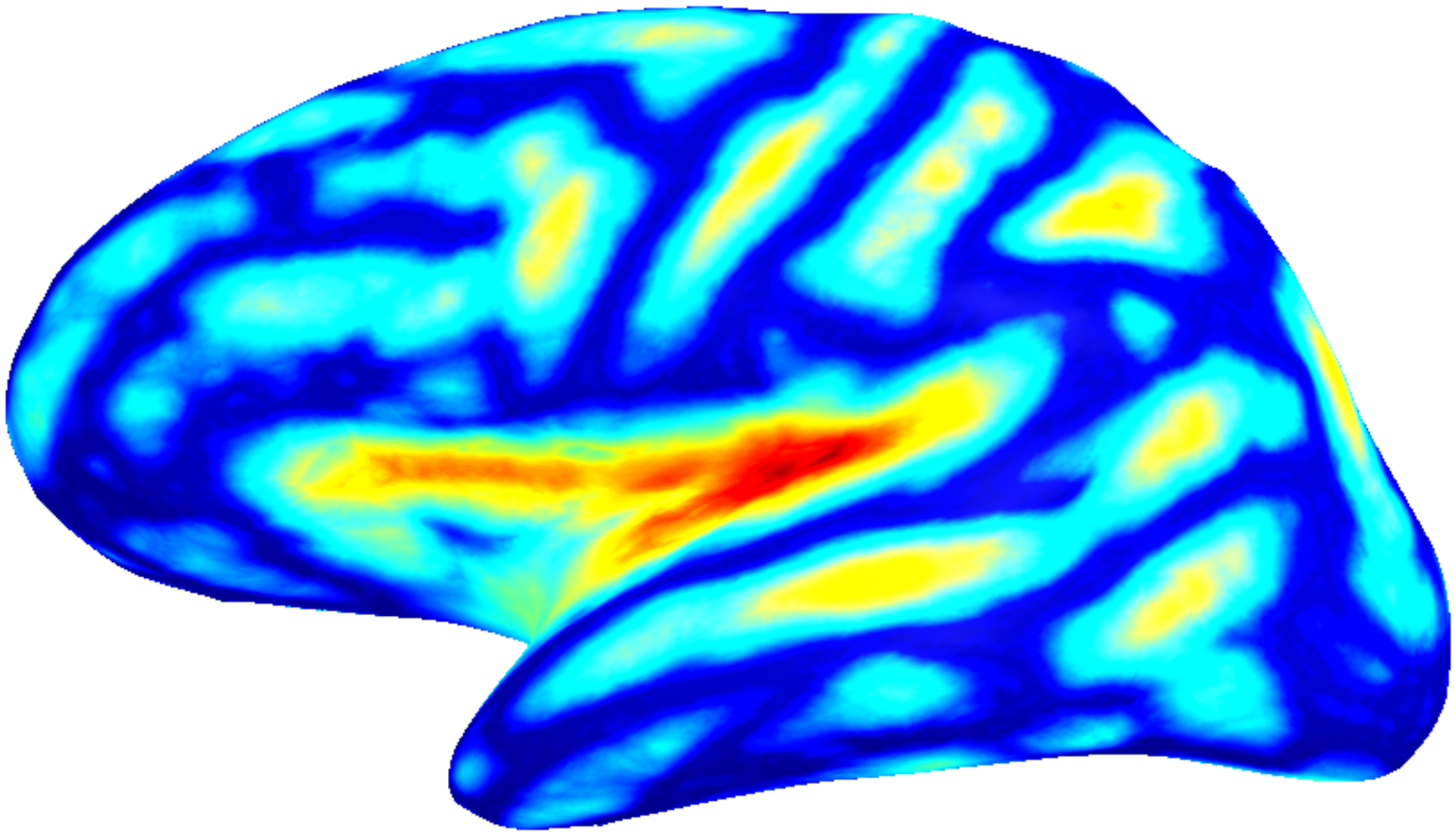


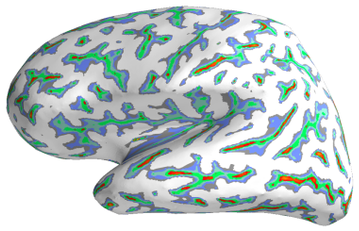
Mean curvature (inflated surface)



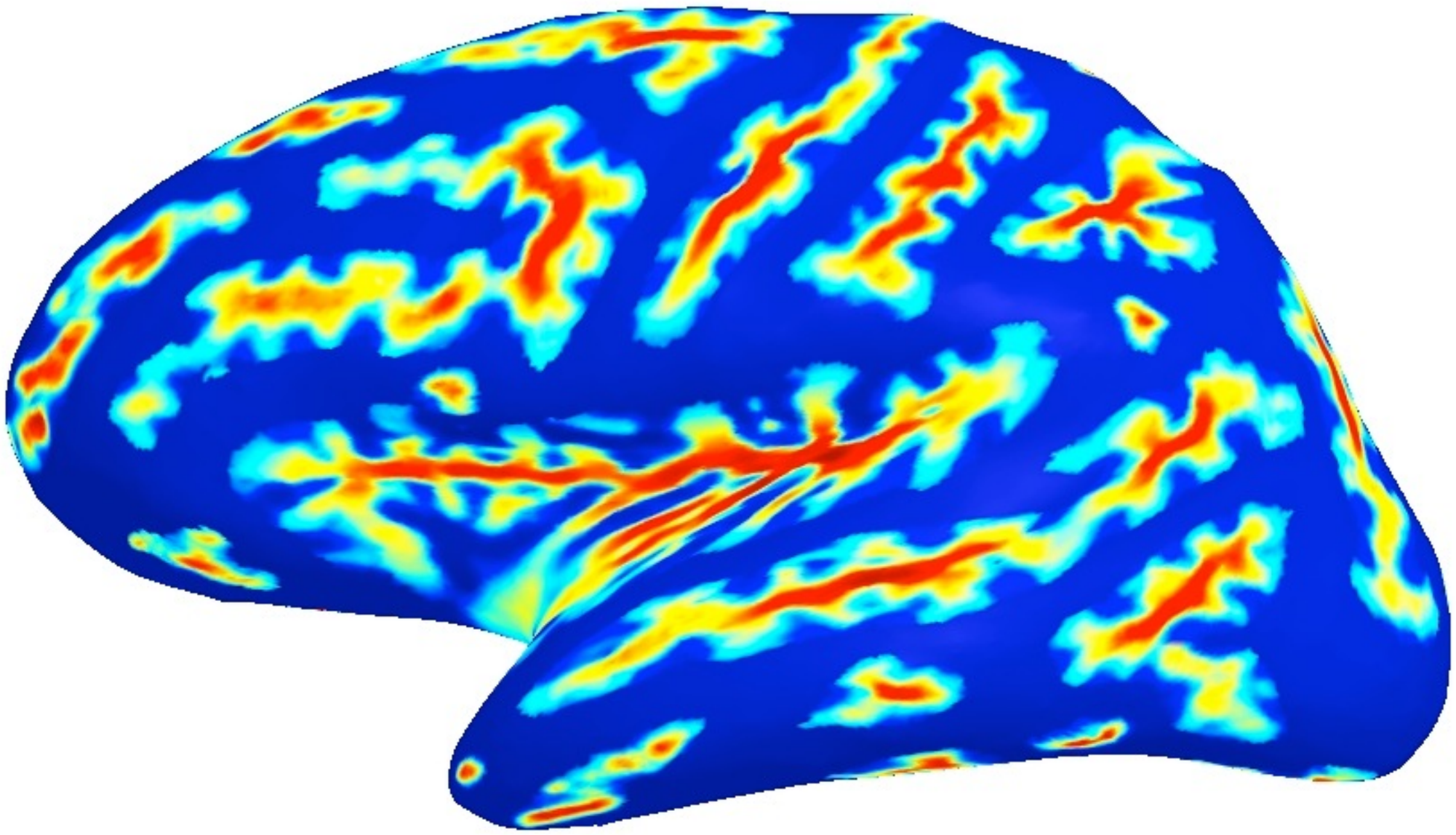


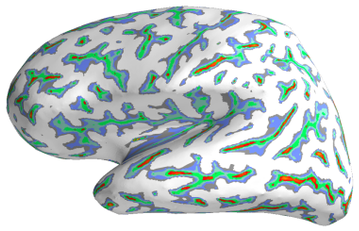
Travel depth (inflated surface)



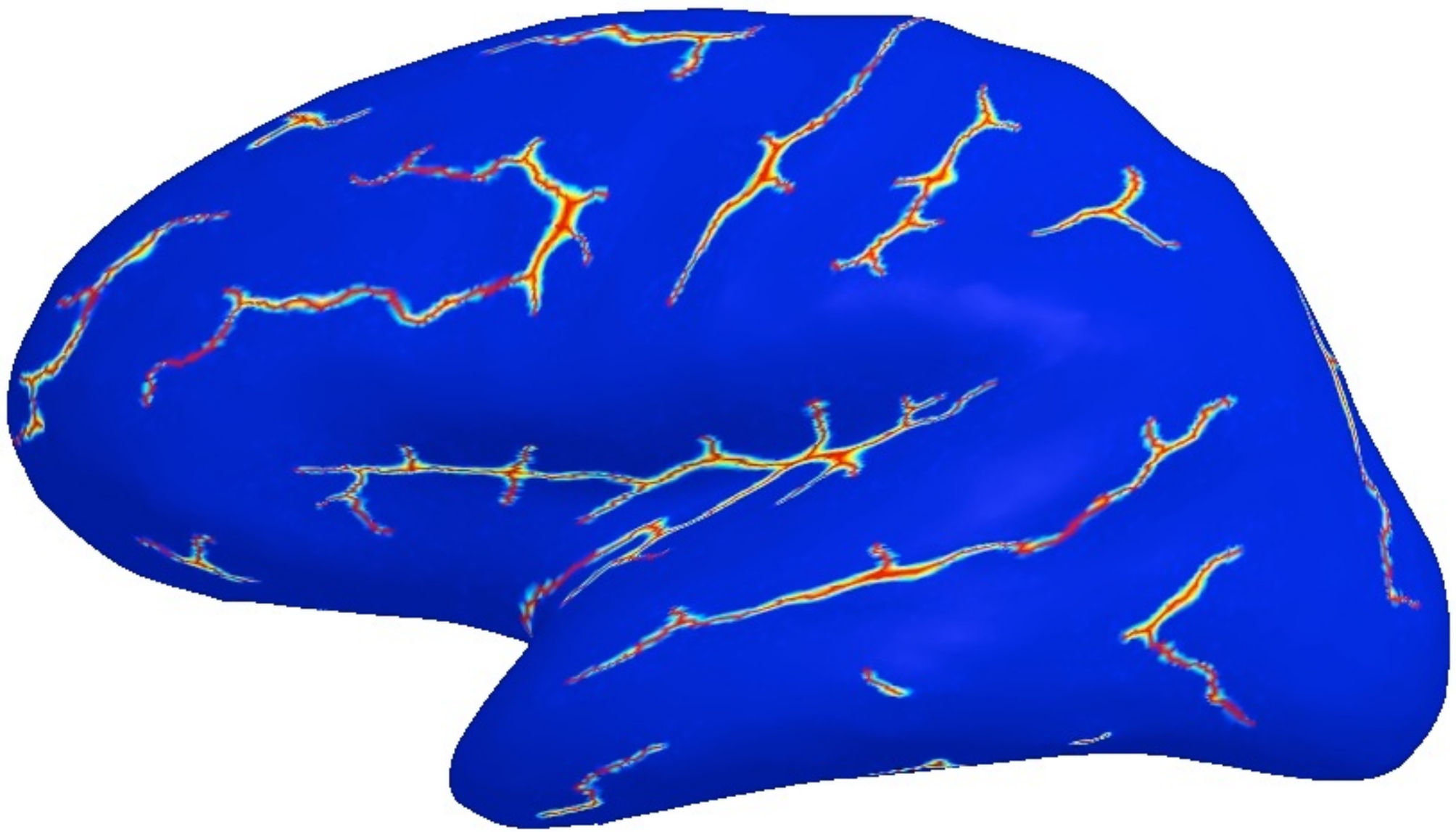


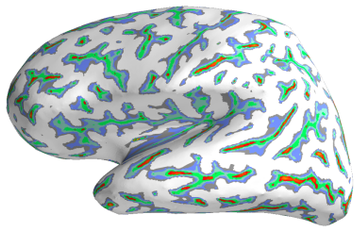
Fundus curves



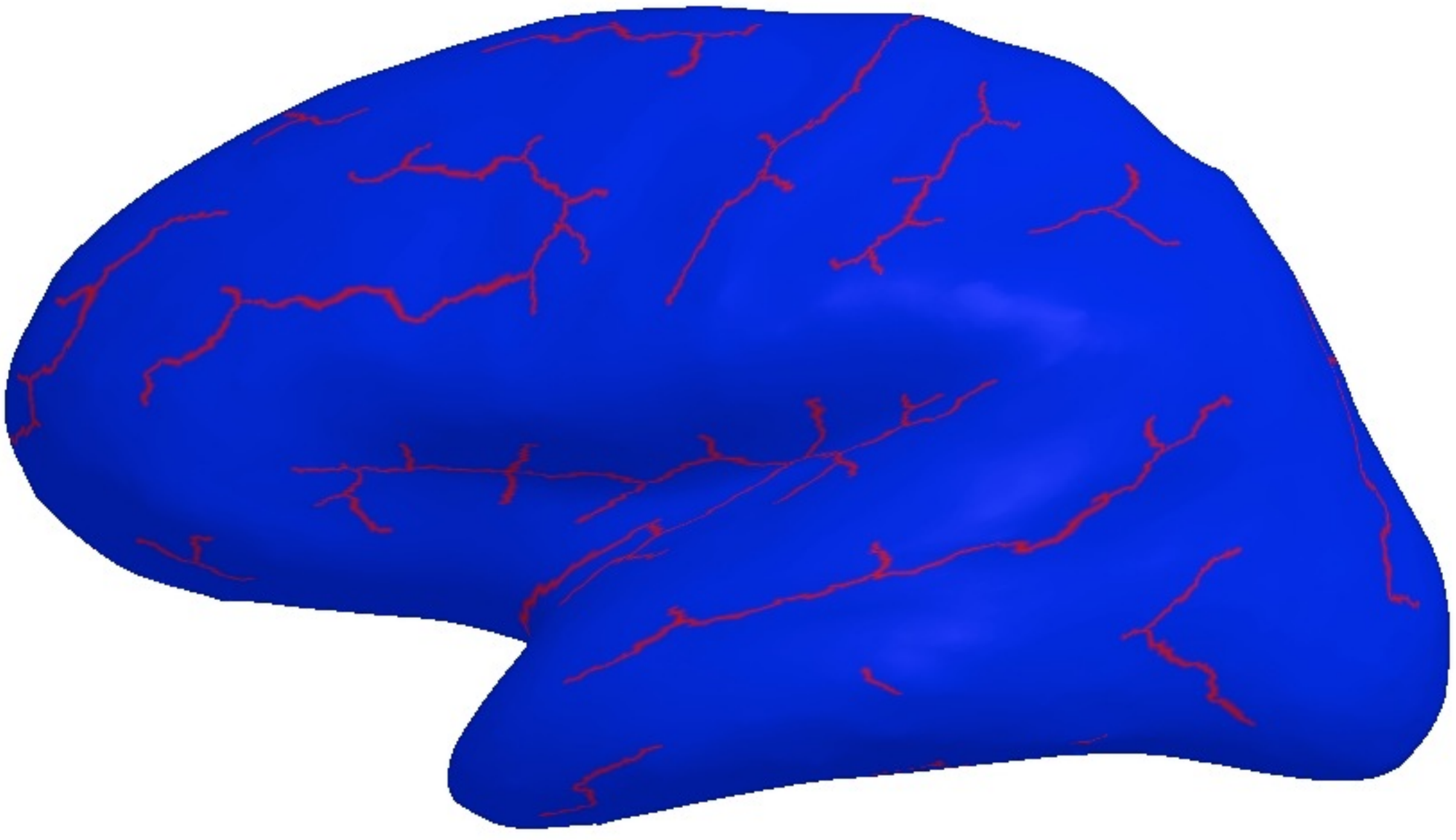


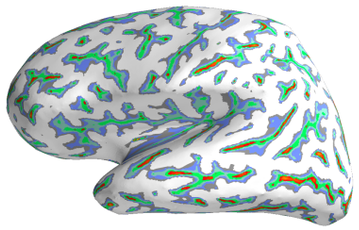
Fundus curves



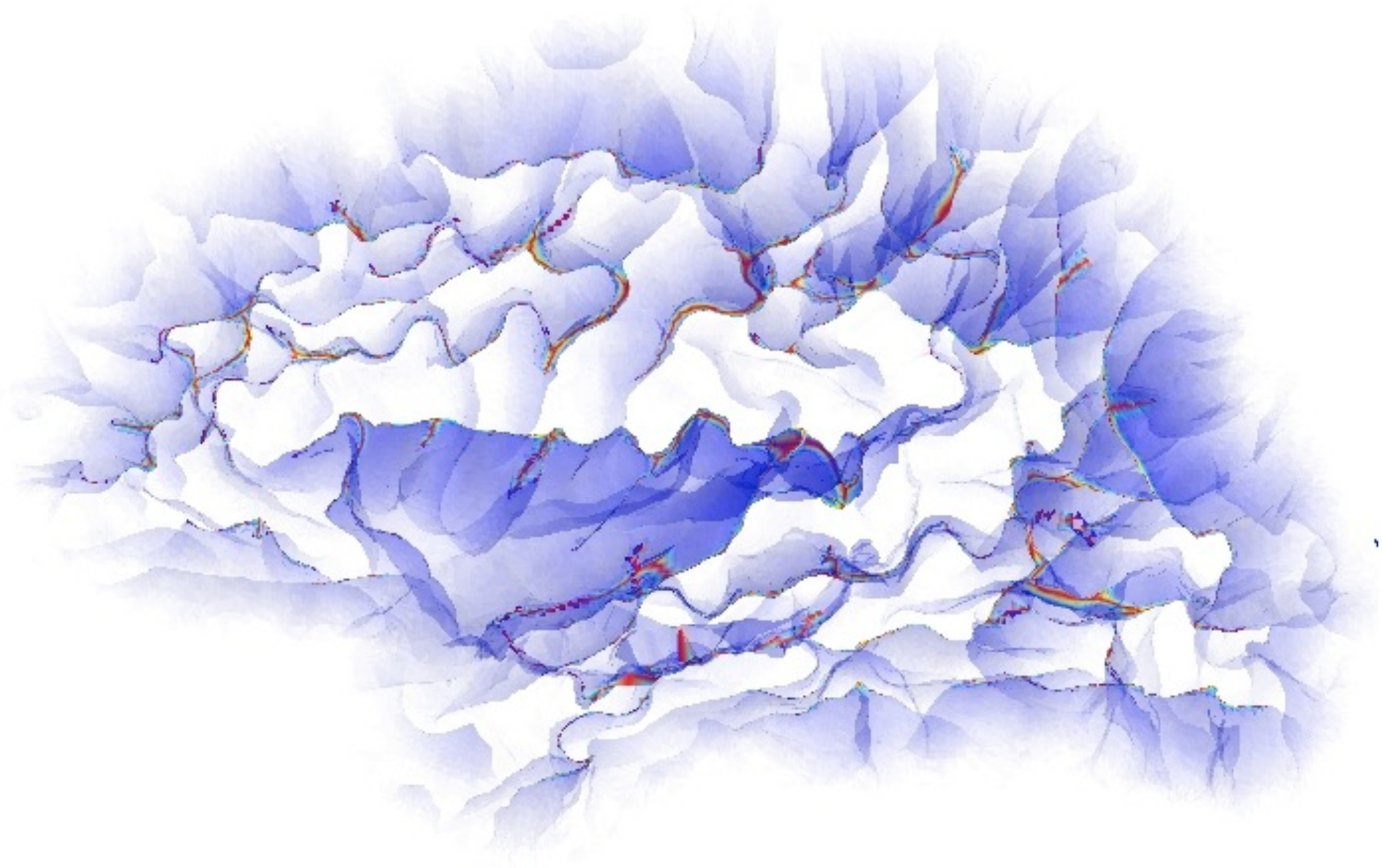


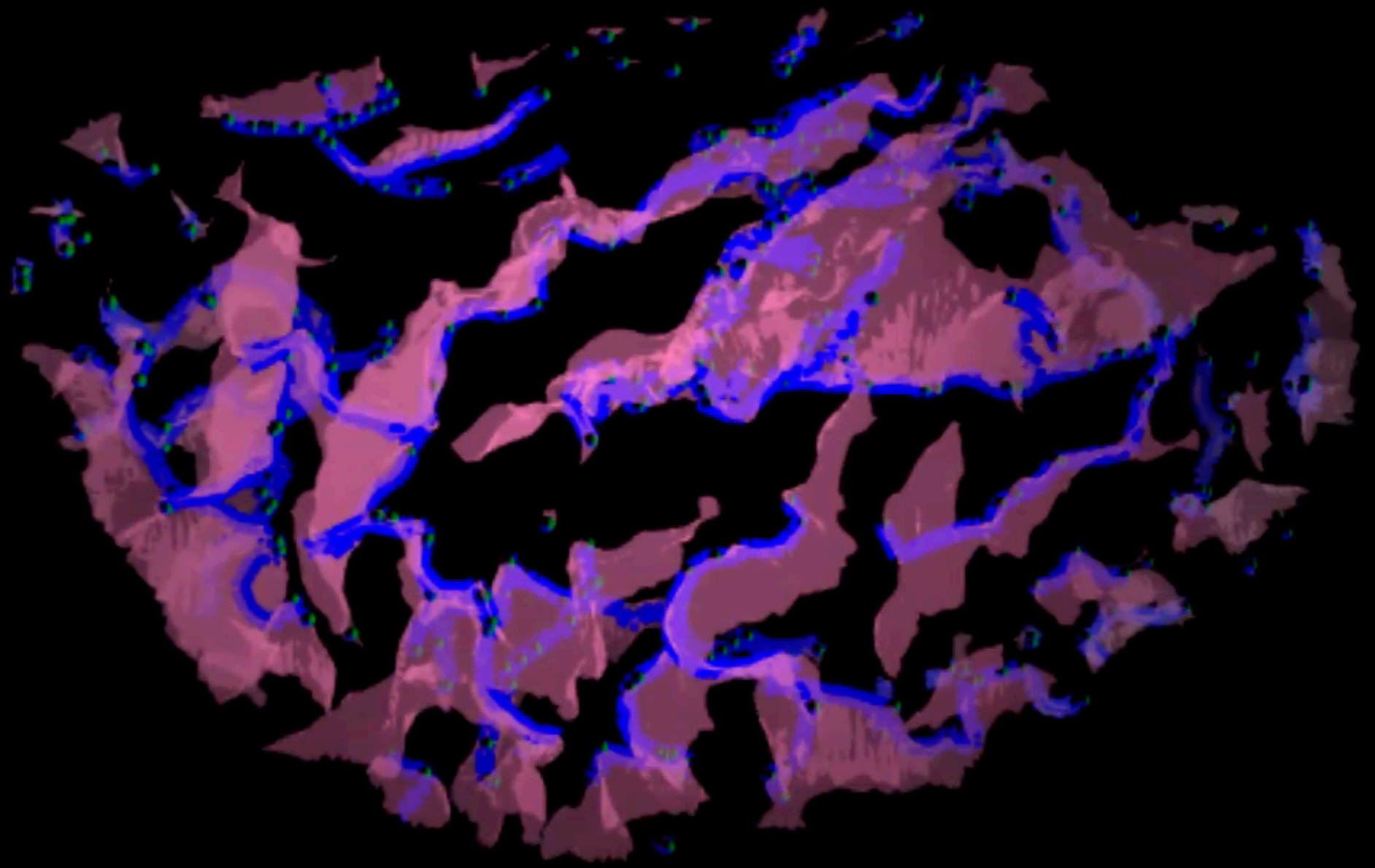
Fundus curves





Folds + fundus curves





Mindboggle

Software

The Mindboggle software package automates shape analysis of anatomical labels and features extracted from human brain MR image data. Mindboggle can be run as a single command, and can be easily installed as a cross-platform virtual machine for convenience and reproducibility of results. Behind the scenes, open source Python and C++ code run within a Nipype pipeline framework.

Help with Mindboggle

- Help in a terminal window (see below for inputs and outputs):

```
mindboggle -h
```

- Documentation
- Installation
- GitHub
- License
- Contributors

Set up Mindboggle

If running Mindboggle in a virtual machine (recommended), type the following two commands in a terminal window, in the same directory as the Vagrantfile you generated (see INSTALL). This will launch and log into the Mindboggle virtual machine (requires an active Internet connection):

```
vagrant up
vagrant ssh
```

Example Mindboggle commands

Example 1: The following bare-bones command runs Mindboggle (and its dependencies) on data processed by FreeSurfer but not ANTs (replace `SUBJECT` with the name of a FreeSurfer subject directory, such as `bert`):

```
mindboggle SUBJECT
```

Example 2: To generate only volume data (no surface labels or measures), this command uses ANTs output files (replace `SEGMENTS` with an ANTs segmented file, such as `ants_output/subject1/antsBrainSegmentation.nii.gz`):

```
mindboggle SUBJECT --ants SEGMENTS --no surfaces
```

mindboggle/mindboggle - mindboggle - [~/Projects/Mindboggle/mindboggle]

mindboggle › mindboggle › mindboggle

Doctest convert2nii

Project

mindboggle (~/Projects/Minc
- build
- database
- doc
- mindboggle
  - evaluate
  - features
  - labels
  - shapes
  - thirdparty
  - utils
  - __init__.py
  - COMMIT_INFO.txt
  - DATA.py
  - info.py
  - LABELS.py
  - LUT.py
  - mindboggle
  - pkg_info.py
- mindboggle_tools
- terms
- .gitignore
- INSTALL
- LABELS
- LICENSE
- MANIFEST.in
- README
- setup.py
- setup_egg.py
- setup_mindboggle
- setup_mindboggle.sh
- STORY
- THANKS
- Vagrantfile
- External Libraries

mindboggle/mindboggle | mindboggle/.../fundi.py | mindboggle/.../plots.py

```
263  #============================================================================
264  #
265  #    Import Nipype and Mindboggle libraries
266  #
267  #============================================================================
268  #----------------------------------------------------------------------------
269  # Nipype libraries
270  #----------------------------------------------------------------------------
271  from nipype import config, logging
272  from nipype.pipeline.engine import Workflow, Node, JoinNode
273  from nipype.interfaces.io import DataGrabber, DataSink
274  from nipype.interfaces.utility import Function as Fn
275  from nipype.interfaces.utility import IdentityInterface
276  from nipype.interfaces.ants import ApplyTransforms
277  #----------------------------------------------------------------------------
278  # Mindboggle libraries
279  #----------------------------------------------------------------------------
280  from mindboggle.LABELS import DKTprotocol
281  from mindboggle.DATA import hashes_url
282  #from mindboggle.evaluate.evaluate_labels import measure_surface_overlap, \
283  #    measure_volume_overlap
284  from mindboggle.features.folds import extract_folds
285  from mindboggle.features.fundi import extract_fundi, segment_fundi
286  from mindboggle.features.sulci import extract_sulci
287  from mindboggle.labels.relabel import relabel_surface, relabel_volume, \
288      keep_volume_labels, remove_volume_labels, overwrite_volume_labels
289  from mindboggle.shapes.laplace_beltrami import spectrum_per_label
290  from mindboggle.shapes.likelihood import compute_likelihood
291  from mindboggle.shapes.shape_tools import area, travel_depth, \
292      geodesic_depth, curvature
293  from mindboggle.shapes.zernike.zernike import zernike_moments_per_label
294  from mindboggle.utils.ants import fetch_ants_data, \
295      ImageMath, PropagateLabelsThroughMask, fill_volume_with_surface_labels, \
296      thickinthehead
297  from mindboggle.utils.compute import volume_per_label
298  from mindboggle.utils.io_nii import convert2nii
299  from mindboggle.utils.io_table import write_shape_stats, \
300      write_vertex_measures
301  from mindboggle.utils.io_uri import retrieve_data
302  from mindboggle.utils.io_vtk import read_vtk, apply_affine_transforms, \
303      freesurfer_surface_to_vtk, freesurfer_curvature_to_vtk, \
304      freesurfer_annot_to_vtk
305  from mindboggle.utils.mesh import rescale_by_neighborhood
306  from mindboggle.utils.paths import smooth_skeleton
307  from mindboggle.utils.segment import split_brain, combine_2labels_in_2volumes
308  from mindboggle.utils.utils import list_strings
309
310  #============================================================================
311  #
312  #    Hidden arguments: paths, label and template data
```

1: Project | 7: Structure | 2: Favorites | Database

mindboggle/.../folds.py - mindboggle - [~/Projects/Mindboggle/mindboggle]

mindboggle › mindboggle › features › folds.py

Doctest convert2nii

Project

mindboggle (~/Projects/Minc
- build
- database
- doc
- mindboggle
  - evaluate
  - features
    - __init__.py
    - folds.py
    - fundi.py
    - sulci.py
  - labels
  - shapes
  - thirdparty
  - utils
  - __init__.py
  - COMMIT_INFO.txt
  - DATA.py
  - info.py
  - LABELS.py
  - LUT.py
  - mindboggle
  - pkg_info.py
- mindboggle_tools
- terms
- .gitignore
- INSTALL
- LABELS
- LICENSE
- MANIFEST.in
- README
- setup.py
- setup_egg.py
- setup_mindboggle
- setup_mindboggle.sh
- STORY
- THANKS
- Vagrantfile

External Libraries

mindboggle/mindboggle | mindboggle/.../folds.py | mindboggle/.../plots.py

```
#!/usr/bin/env python
"""Functions to extract folds...."""

#=============================================================================
# Extract folds
#=============================================================================
def extract_folds(depth_file, min_fold_size=50, tiny_depth=0.001, save_file=False):
    """
    Use depth to extract folds from a triangular surface mesh.

    Steps ::
        1. Compute histogram of depth measures.
        2. Define a depth threshold and find the deepest vertices.
        3. Segment deep vertices as an initial set of folds.
        4. Remove small folds.
        5. Find and fill holes in the folds.
        6. Renumber folds.

    Step 2 ::
        To extract an initial set of deep vertices from the surface mesh,
        we anticipate that there will be a rapidly decreasing distribution
        of low depth values (on the outer surface) with a long tail
        of higher depth values (in the folds), so we smooth the histogram's
        bin values, convolve to compute slopes, and find the depth value
        for the first bin with slope = 0. This is our threshold.

    Step 5 ::
        The folds could have holes in areas shallower than the depth threshold.
        Calling fill_holes() could accidentally include very shallow areas
        (in an annulus-shaped fold, for example), so we call fill_holes() with
        the argument exclude_range set close to zero to retain these areas.

    Parameters
    ----------
    depth_file : string
        surface mesh file in VTK format with faces and depth scalar values
    min_fold_size : integer
        minimum fold size (number of vertices)
    tiny_depth : float
        largest non-zero depth value that will stop a hole from being filled
    save_file : Boolean
        save output VTK file?

    Returns
    -------
    folds : list of integers
        fold numbers for all vertices (-1 for non-fold vertices)
    n_folds : int
        number of folds
    depth_threshold : float
```

2: Favorites

Database

mindboggle/.../plots.py - mindboggle - [~/Projects/Mindboggle/mindboggle]

mindboggle › mindboggle › utils › plots.py

Doctest convert2nii

Project

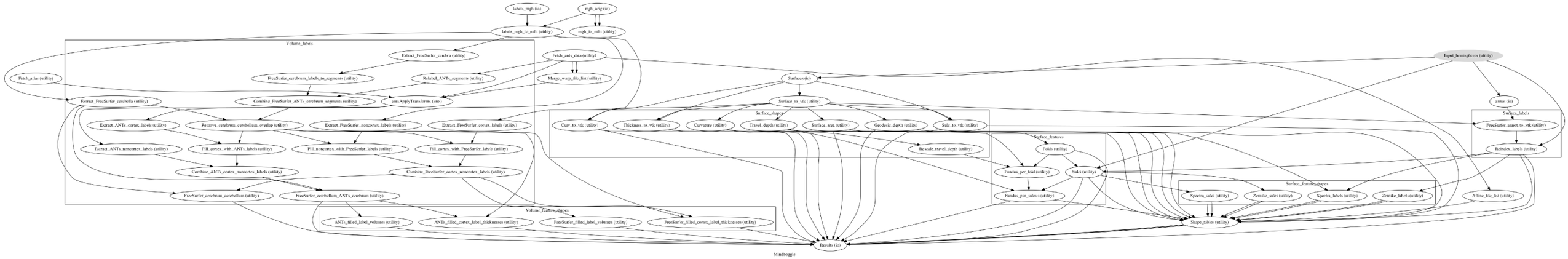
mindboggle (~/Projects/M
- build
- database
- doc
- mindboggle
  - evaluate
  - features
  - labels
  - shapes
  - thirdparty
  - utils
    - __init__.py
    - ants.py
    - compute.py
    - graph.py
    - io_nii.py
    - io_table.py
    - io_uri.py
    - io_vtk.py
    - kernels.py
    - mesh.py
    - morph.py
    - paths.py
    - plots.py
    - segment.py
    - utils.py
  - __init__.py
  - COMMIT_INFO.txt
  - DATA.py
  - info.py
  - LABELS.py
  - LUT.py
  - mindboggle
  - pkg_info.py
- mindboggle_tools
- terms
- .gitignore
- INSTALL
- LABELS

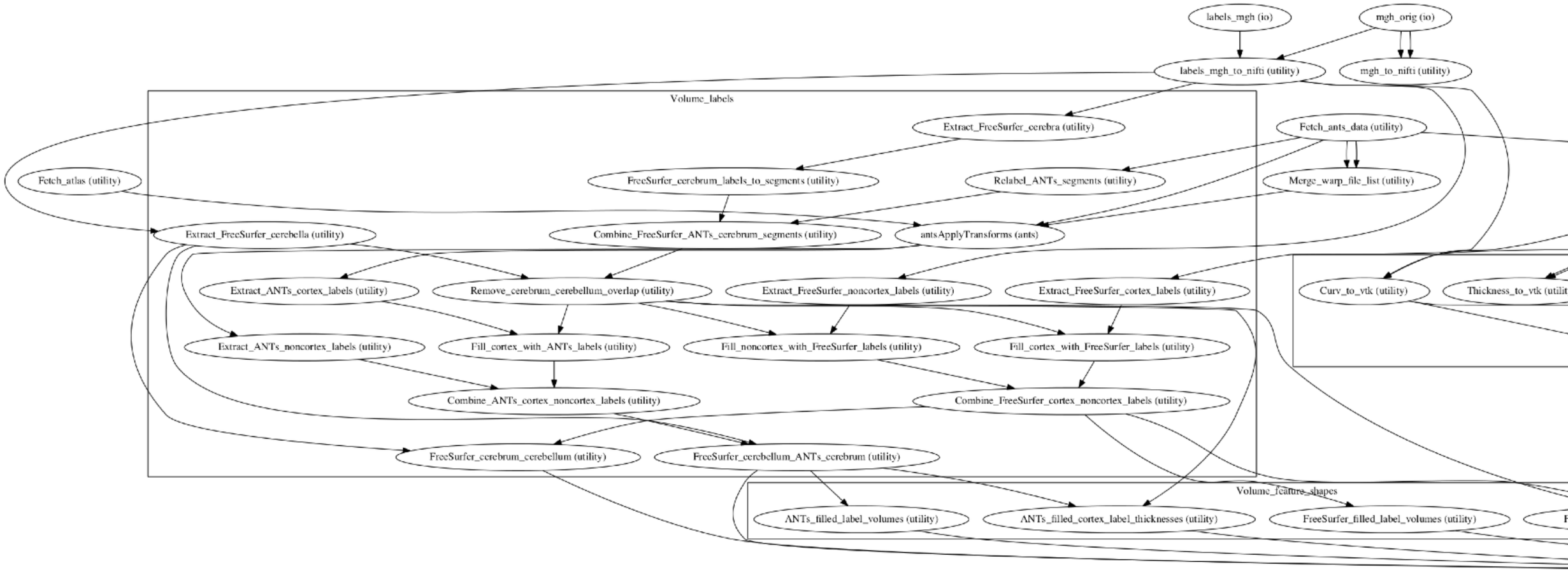
mindboggle/mindboggle | mindboggle/.../folds.py | mindboggle/.../plots.py

```
80
81  def plot_mask_surface(vtk_file, mask_file='', nonmask_value=-1,
82                        masked_output='', remove_nonmask=False,
83                        program='vtkviewer',
84                        use_colormap=False, colormap_file=''):
85      """
86      Use vtkviewer or mayavi2 to visualize VTK surface mesh data.
87
88      If a mask_file is provided, a temporary masked file is saved,
89      and it is this file that is viewed.
90
91      If using vtkviewer, can optionally provide colormap file
92      or set $COLORMAP environment variable.
93
94      Parameters
95      ----------
96      vtk_file : string
97          name of VTK surface mesh file
98      mask_file : string
99          name of VTK surface mesh file to mask vtk_file vertices
100     nonmask_value : integer
101         nonmask (usually background) value
102     masked_output : string
103         temporary masked output file name
104     remove_nonmask : Boolean
105         remove vertices that are not in mask? (otherwise assign nonmask_value)
106     program : string {'vtkviewer', 'mayavi2'}
107         program to visualize VTK file
108     use_colormap : Boolean
109         use Paraview-style XML colormap file set by $COLORMAP env variable?
110     colormap_file : string
111         use colormap in given file if use_colormap==True? if empty and
112         use_colormap==True, use file set by $COLORMAP environment variable
113
114     Examples
115     --------
116     >>> import os
117     >>> from mindboggle.utils.plots import plot_mask_surface
118     >>> path = os.environ['MINDBOGGLE_DATA']
119     >>> vtk_file = os.path.join(path, 'arno', 'labels', 'lh.labels.DKT31.manual.vtk')
120     >>> mask_file = os.path.join(path, 'test_one_label.vtk')
121     >>> nonmask_value = 0 #-1
122     >>> masked_output = ''
123     >>> remove_nonmask = True
124     >>> program = 'vtkviewer'
125     >>> use_colormap = True
126     >>> colormap_file = '' #'/software/mindboggle_tools/colormap.xml'
127     >>> plot_mask_surface(vtk_file, mask_file, nonmask_value, masked_output, remove_nc
128
```

1: Project | Z: Structure | 2: Favorites | Database

NiPype pipeline framework





Hackathon challenge #3: MNI152 mashup

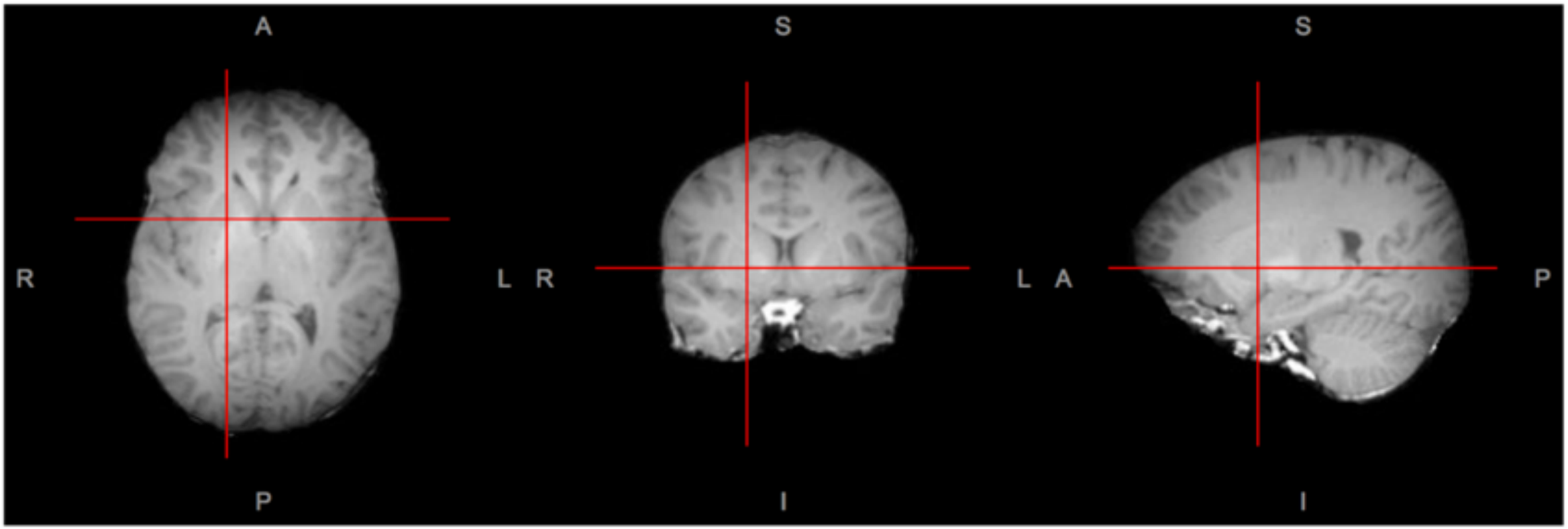
Anatomical labels, shape measures, and gene expression in MNI152 space

Mindboggle-102 team

arno klein / jason tourville / jay bohland / rich stoner

Allen human brain image and gene expression data

H0351.2001



1cm MNI Coordinate: **21, 8, 1**

Structure: **Striatum**

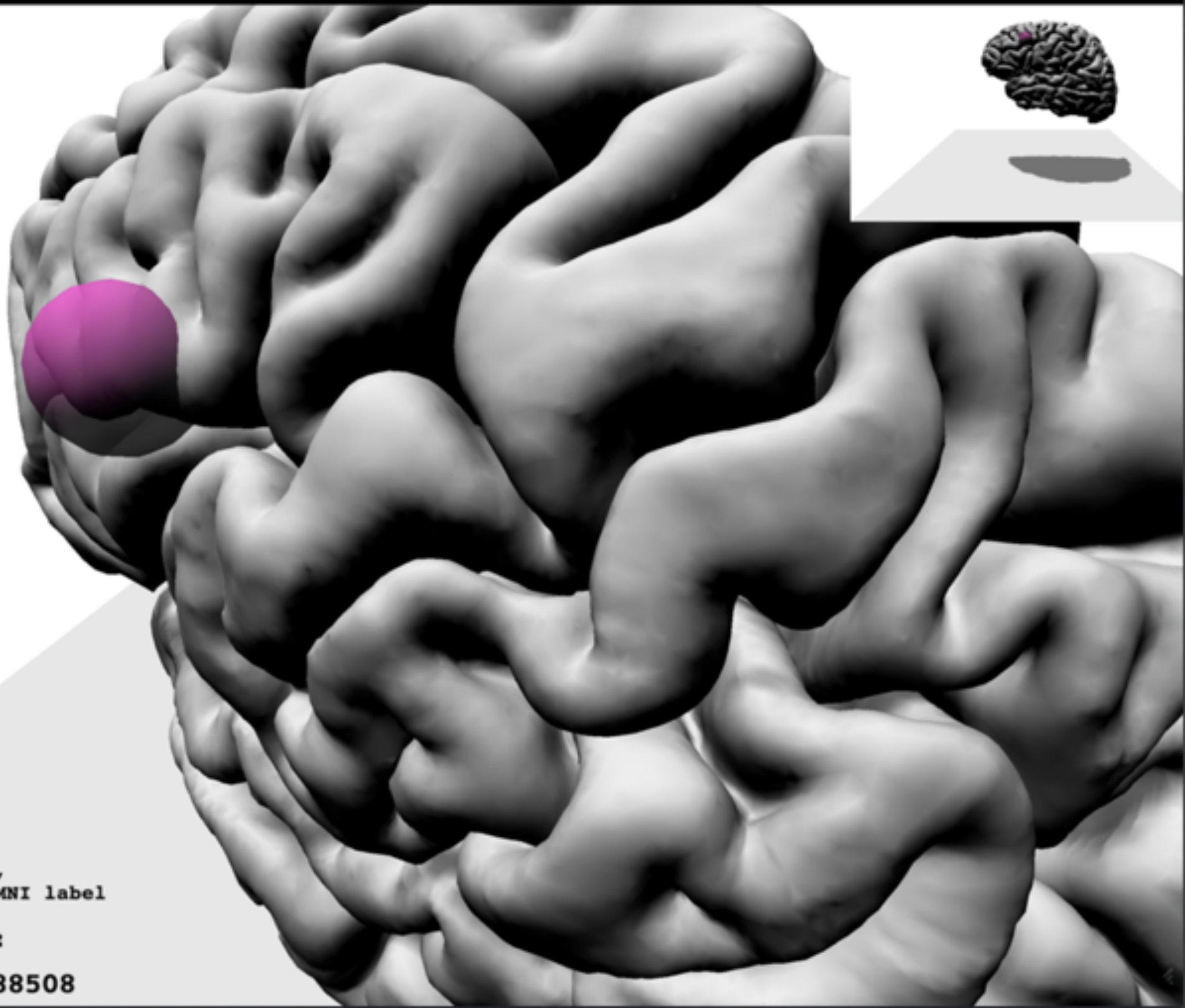
Putamen, Right (Pu)

MNI152 Visualization Demo

Show select label

Controls

Label 3



Highest expression,
aggregated within MNI label

A_24_P109661 :

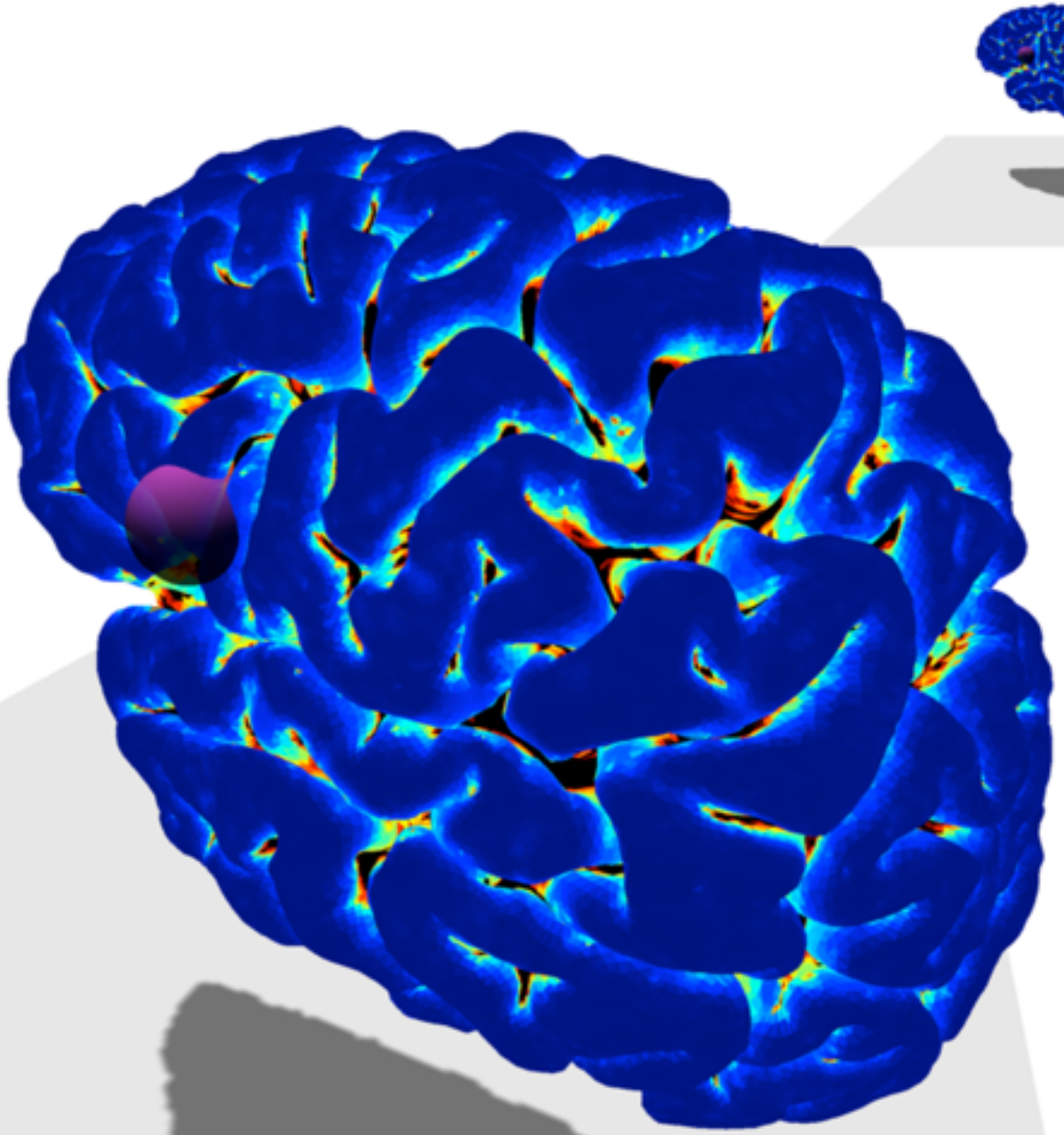
5.3213262053588508

MNI152 Visualization Demo

Show select label

Controls

Label 18



Highest expression,
aggregated within MNI label

A_23_P169812 :

6.4810893969243741

Alzheimers Disease Big Data DREAM Challenge 1

Sharing

Synapse ID: syn2290704
DOI: (doi:10.7303/syn2290704)

Wiki Files

Pages

Alzheimers Disease Big Data DREAM Challenge 1

1. Overview
  1.1 Steps to Participate
  1.2 DREAM9 Challenges Rules
2. Incentives
3. Data Access
4. Data Description and Format
  4.1 Training
  4.2 Ancillary
  4.3 Test - ROS/MAP
  4.4 Test - AddNeuroMed
5. Questions and Scoring
  5.1 Timelines
6. Leaderboards
  6.1 Submitting Results
    6.1.1 Making submission via the web

Please see News and Updates for messages about data and other AD Challenge information. Questions can be posed in the Community Forum. See Step 3 for how to sign up

Alzheimer's Disease Big Data DREAM Challenge #1
Launch: June 2, 2014
Close: October 3, midnight Pacific Time

Join

Alzheimers Disease Big Data DREAM Challenge 1

Sharing

Synapse ID: syn2290704
DOI: (doi:10.7303/syn2290704)

Wiki Files

Pages

Alzheimers Disease Big Data DREAM Challenge 1
1. Overview
1.1 Steps to Participate
1.2 DREAM9 Challenges Rules
2. Incentives
3. Data Access
4. Data Description and Format
4.1 Training
4.2 Ancillary
4.3 Test - ROS/MAP
4.4 Test - AddNeuroMed
5. Questions and Scoring
5.1 Timelines
6. Leaderboards
6.1 Submitting Results
6.1.1 Making submission via the web

Alzheimers Disease Big D... » 6.4 Q3 Leaderboard

6.4 Q3 Leaderboard

ID	Date	Name	entity	team	Percent Correct	Pearson MMSE	CCC MMSE	Mean Rank	Final Rank
2636580	08/21/2014 08:31:37AM	Guanlab05	syn2636579	Guanlab_UMich	38.30	0.6735	0.5792	1.50	1.50
2665590	09/03/2014 11:48:08AM	Guanlab6	syn2665589	Guanlab_UMich	48.94	0.6716	0.5803	1.50	1.50
2666633	09/03/2014 05:58:07PM	pred_11.csv	syn2666632	jn13	52.13	0.5767	0.5360	4.00	3.50
2677163	09/10/2014 08:37:39PM	SS100250	syn2677162	Bias	59.57	0.6018	0.5338	4.00	3.50
2633527	08/20/2014 02:00:40PM	Guanlab04	syn2633526	Guanlab_UMich	46.81	0.6275	0.5232	4.50	5.00
2677157	09/10/2014 08:29:07PM	SS250	syn2677156	Bias	59.57	0.5688	0.5245	6.00	6.00
2668034	09/05/2014 08:10:34AM	pred_12.csv	syn2666889	jn13	52.13	0.5708	0.5114	6.50	7.00
2662296	09/01/2014 02:47:55PM	pred_9.csv	syn2662278	jn13	52.13	0.5679	0.4676	9.50	8.00
2648187	08/23/2014 06:52:21PM	pred_8.csv	syn2648186	jn13	52.13	0.5605	0.4699	10.00	9.00
2662777	09/01/2014	pred_10.csv	syn2662761	jn13	52.13	0.5677	0.4653	10.50	10.50

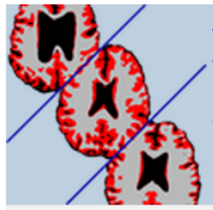
All Challenges

Here is an overview of all challenges that have been organized within the area of medical image analysis that we are aware of. If you know any study that would fit in this overview, please let us know by filling out this form.

Showing 76 projects of 76

Filter by: ☐ Open for submissions (41) ☐ Data download (47) ☐ Hosted on Grand-challenge (6)

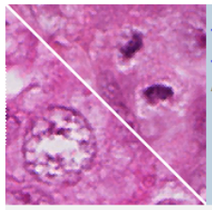
2014



Open for submissions
Data download
Workshop: Sep 18, 2014
Associated with: MICCAI 2014
Hosted on: grand-challenge.org

CADDementia

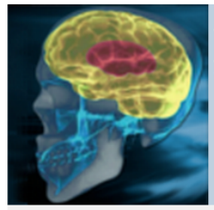
We seek algorithms that perform multi-class classification of patients with Alzheimer's disease (AD), patients with mild cognitive impairment (MCI) and healthy controls (CN) using multi-center structural MRI data.



Open for submissions
Data download
Associated with: ICPR 2014
Hosted on: grand-challenge.org

MITOS-ATYPIA-14

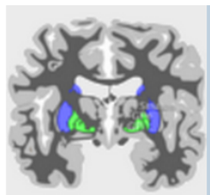
MITOS & ATYPIA 14 Contest, hosted by conference ICPR 2014 Detection of mitosis and evaluation of nuclear atypia on breast cancer H&E stained images



Open for submissions
Data download
Workshop: Sep 14, 2014

Brain Tumor Image Segmentation (BraTS 2014)

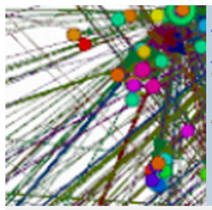
Contains three sub-challenges. 1: Segmentation of brain tumor (GBM) and its components, 2: Longitudinal Evaluation of time series image data and 3: Classification into Low- and and High Grade (GBM) gliomas.



Open for submissions
Data download
Associated with: STN 2014

Subthalamic Nucleus Segmentation Challenge 2014

This challenge seeks to bring together leading academic researchers to tackle the difficult problem of Subthalamic Nucleus (STN) segmentation in MRI volumes. Sponsored by Renishaw plc and organised by Aberytswyth University



Open for submissions
Data download
Workshop: Sep 14, 2014
Associated with: MICCAI 2014

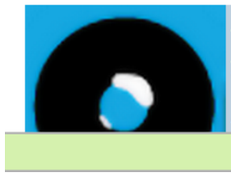
Brain Tumor Digital Pathology Challenge

Evaluation of classification of LGG and GBM, and segmentation of necrotic and normal brain regions from high-res digital pathology slide clinical cases.

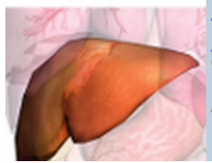
Data download
Workshop: Sep 12, 2014
Associated with: ECCV 2014

Leaf Segmentation Challenge

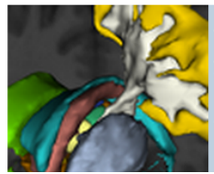
To advance the state of the art in leaf segmentation and to demonstrate the difficulty of segmenting all leaves in an image of plants, we organize the Leaf Segmentation Challenge (LSC).



Open for submissions
Data download
Workshop: Sep 18, 2014
Associated with: MICCAI 2014



Open for submissions
Data download
Workshop: Sep 14, 2014
Associated with: MICCAI 2014



Open for submissions
Data download
Workshop: Sep 14, 2014
Associated with: MICCAI 2014