



# Nilearn: Streamlined neuroimaging analysis now with surface data support

DOI [10.5281/zenodo.8397156](https://doi.org/10.5281/zenodo.8397156)

RRID [SCR\\_001362](https://rrid.info/RRID/SCR_001362)

Rémi Gau, Himanshu Aggarwal, Alexis Thual, Elizabeth DuPre, Hande Gözükan, Hao-Ting Wang, Jérôme Dockès, Michelle Wang, Mohammad Torabi, Nicolas Gensollen, Taylor Salo, Victoria Shevchenko, Yasmin Mzayek, Bertrand Thirion and Nilearn contributors

## What is Nilearn ?

- Python package for analysis of brain images
  - [Connectivity analysis \(resting-state\)](#)
  - [Decoding \(MVPA\)](#)
  - [GLM \(stats\)](#)
  - [Plotting volumetric and surface data](#)
- Well documented
- Supportive community
- Open-source and community driven

## Releases 0.10.4 - 0.11.1

- new SurfaceImage object to load and save meshes and data for both hemispheres.
- new maskers to extract data from SurfaceImage
- support for SurfaceImage across most modules: plotting, decoder and glm
- new examples demonstrating SurfaceImage in action

## New surface API

### # get a mesh and contrast maps in Nifti format

```
from nilearn.datasets import fetch_surf_fsaverage,
fetch_localizer_contrasts
```

```
fsaverage = fetch_surf_fsaverage()
cmaps = fetch_localizer_contrasts(n_subjects=10,
contrasts=["left button press", "right button press"])
```

### # project volume onto a mesh via SurfaceImage object

```
from nilearn.surface import SurfaceImage
```

```
mesh = {"left": fsaverage["pial_left"],
        "right": fsaverage["pial_right"]}
```

```
surf_img = SurfaceImage.from_volume(mesh=mesh,
volume_img=cmaps["cmaps"][0])
```

```
print(surf_img)
<SurfaceImage (20484, 1)>
```

### # do it for each contrast map

```
surf_imgs = []
labels = []

for cmap in cmaps["cmaps"]:
    surf_img = SurfaceImage.from_volume(mesh=mesh,
volume_img=cmap)
    surf_imgs.append(surf_img)
    if "Left" in cmap:
        labels.append("left")
    else:
        labels.append("right")
```

### # quickly Decode from surface data

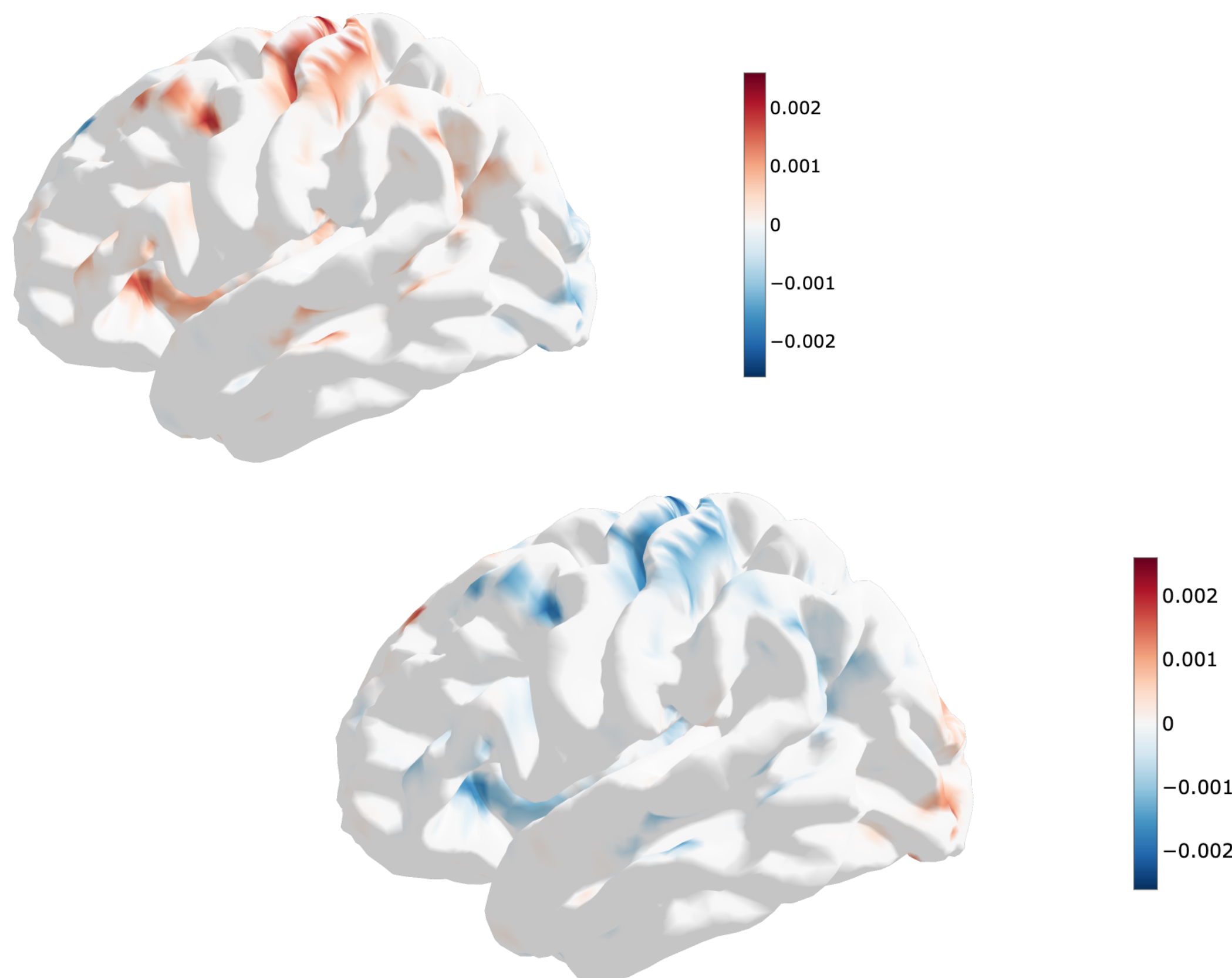
```
from nilearn.decoding import Decoder
```

```
decoder = Decoder(n_jobs=5)
decoder.fit(surf_imgs, y=labels)
```

### # easily plot classifier's coefficients

```
from nilearn.plotting import view_surf
```

```
view_surf(decoder.coef_img_["left"])
view_surf(decoder.coef_img_["right"])
```



## Future directions

- Extend surface support for image module
- Improve plotting backends
- Better compliance with scikit-learn requirements

## Join the community!!!

- Check the documentation [nilearn.github.io](https://nilearn.github.io)
- Ask questions on [neurostars.org/tag/nilearn](https://neurostars.org/tag/nilearn)
- Contribute GitHub [github.com/nilearn/nilearn](https://github.com/nilearn/nilearn)
- Weekly drop-in hour, Wednesday 4pm UTC [meet.jit.si/nilearn-drop-in-hours](https://meet.jit.si/nilearn-drop-in-hours)



Human Brain Project

