



antsxmm

The Automated **Bridge** Between
BIDS and **ANTsPyMM**

Modern neuroimaging research is a multi-modal challenge.

Researchers work with a diverse array of imaging data, each providing a unique view of the brain. Managing these modalities—from structural T1w and FLAIR to diffusion (DWI) and functional (fMRI)—is the first hurdle in any analysis pipeline.



T1w (Anatomical)



DWI (Diffusion)



rsfMRI (Functional)



FLAIR (Pathology)



Neuromelanin



ASL Perfusion



PET

Two powerful standards, one critical gap.



The BIDS Standard

Organizes complex data from tools like `antsxbids` into a predictable, shareable structure. It's the standard for data organization.

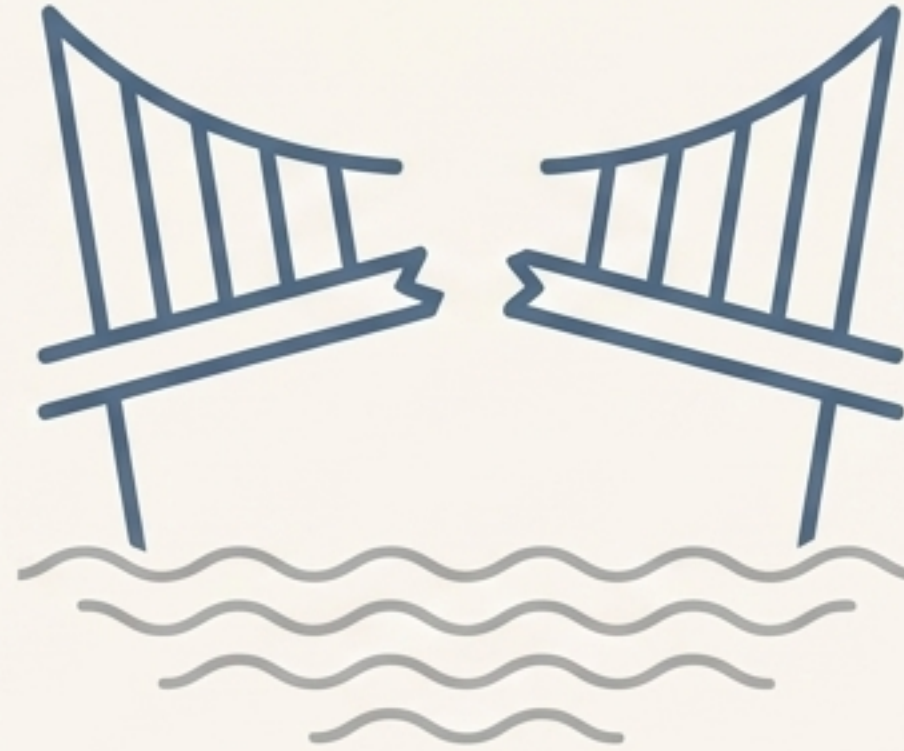
```
/sub-211239
├── /ses-20230405
│   ├── /anat
│   │   └── ..._T1w.nii.gz
│   └── /dwi
│       └── ..._dir-LR_dwi.nii.gz
```



The ANTsPyMM Engine

A robust library for advanced neuroimage processing, but it expects a very specific input format.

```
/PPMI
├── /sub-211239
│   └── /ses-20230405
│       └── /DTILR
│           └── ...PPMI+..._DTILR_...nii.gz
```



`antsxmm` seamlessly connects your data to the analysis.



`antsxmm` is a streamlined wrapper that acts as an intelligent adapter. It automates the tedious and error-prone process of translating BIDS-structured data into the format required by `ANTsPyMM`, enabling end-to-end processing with a single command.



Automation: Eliminate manual file renaming and restructuring.



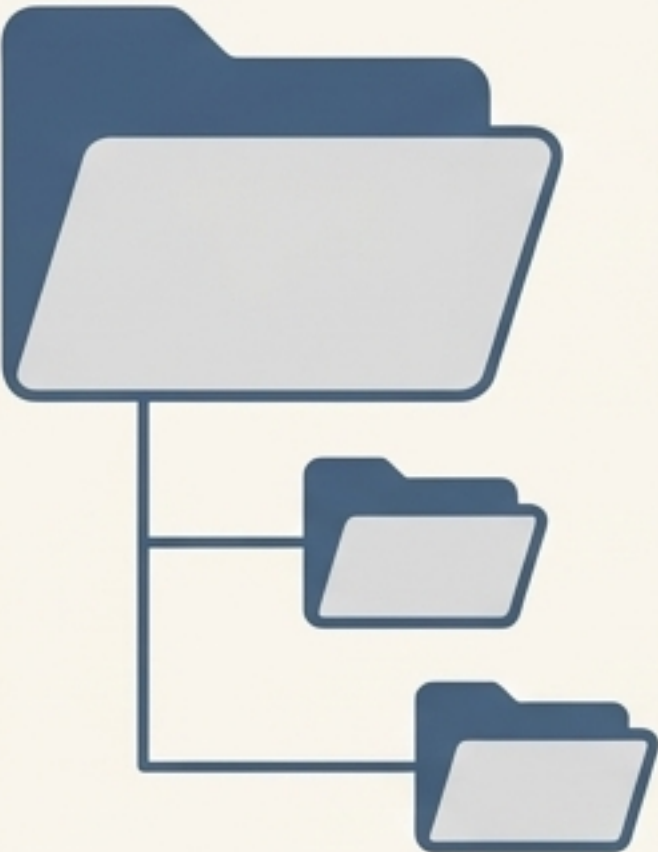
Standardization: Ensure data is perfectly formatted for `ANTsPyMM` every time.



Integration: Bridge two best-in-class tools into a single, cohesive workflow.

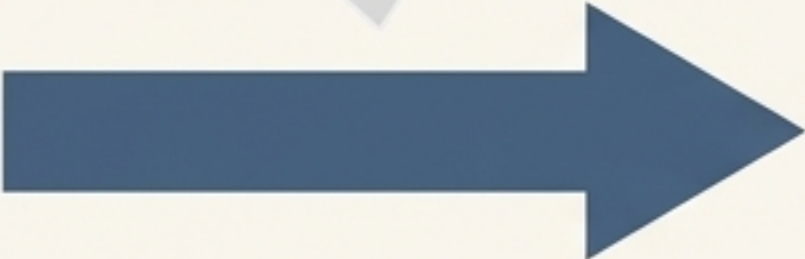
Step 1: It begins by parsing the BIDS layout.

`antsxmm` scans the entire BIDS directory, identifying every subject, session, and imaging modality. Using `parse_antspbids_layout`, it creates an internal manifest of all available data, ensuring no scan is missed.



BIDS Directory

```
Code Snippet:  
'layout_df = parse_antspbids_layout(bids_dir)'
```



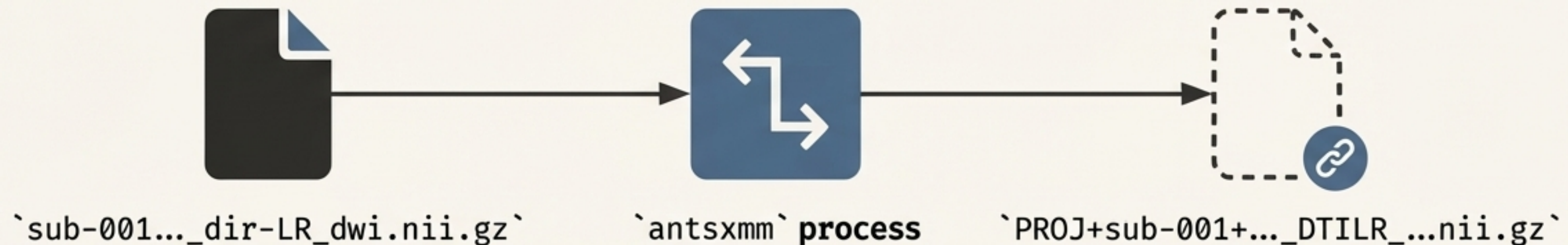
subjectID	date	t1_filename	dti_filenames	rsf_filenames	...
subjectID	2022-11-11	t1_filename	dti_filenames	rsf_filenames	...
subjectID	2022-11-11	t1_filename	dti_filenames	rsf_filenames	...
...

Internal Manifest (DataFrame)

Step 2: Files are staged and standardized without data duplication.

This is the core translation step. For each file, `antsxmm` performs two critical actions:

1. **Renames on-the-fly:** It creates the strict `Project-Subject-Date-Modality` filename that `ANTsPyMM` requires.
2. **Uses symlinks:** Instead of copying large image files, it creates symbolic links in a temporary staging area. This is fast and saves significant disk space.



Function: `sanitize_and_stage_file` using `os.symlink`

Step 3: The standardized data is processed by ANTsPyMM.

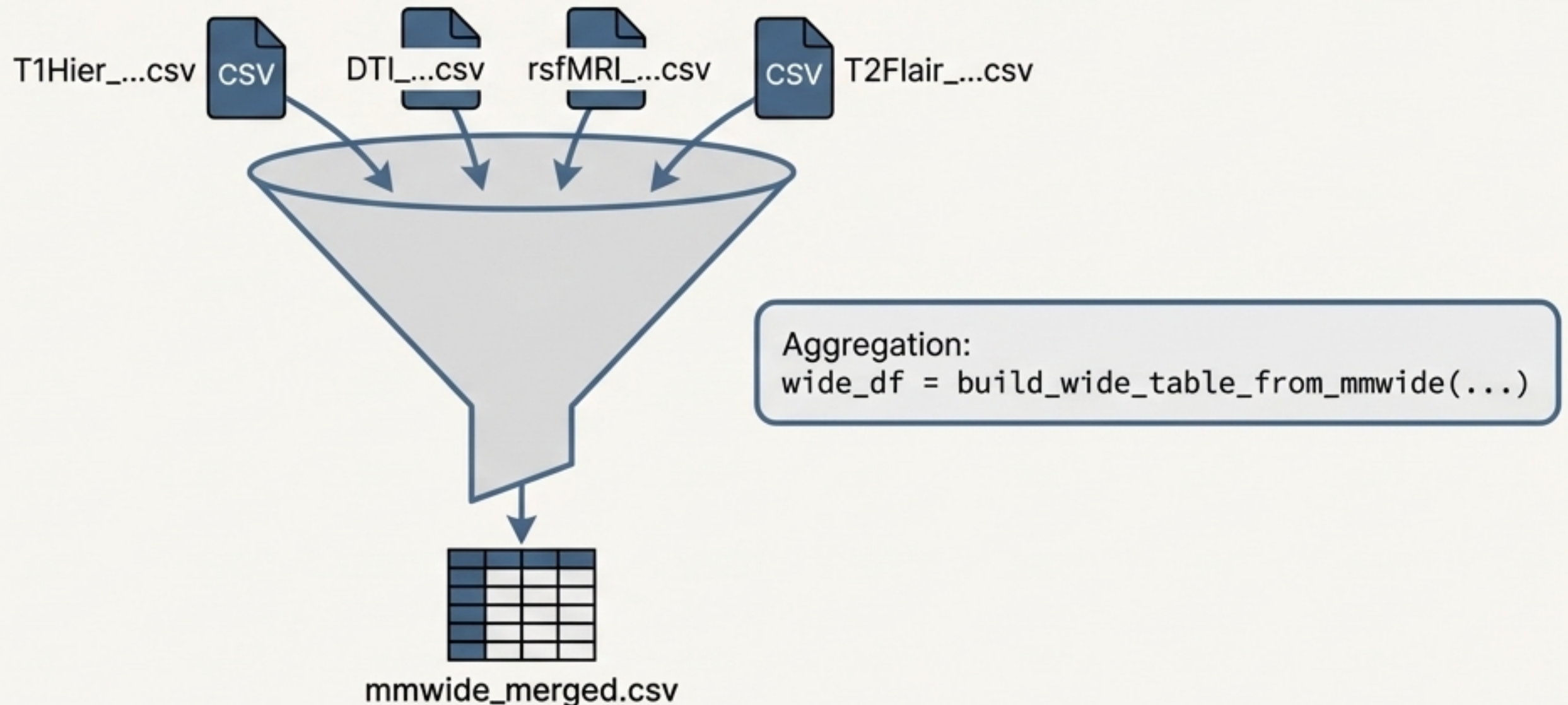
With the data perfectly staged, `antsxmm` executes the comprehensive ANTsPyMM pipeline. This is where the robust, validated neuroimaging algorithms run, performing tasks like motion correction, normalization, and quantitative metric extraction.



```
Execution: `antspymm.mm_csv(study_csv_clean, ...)`
```

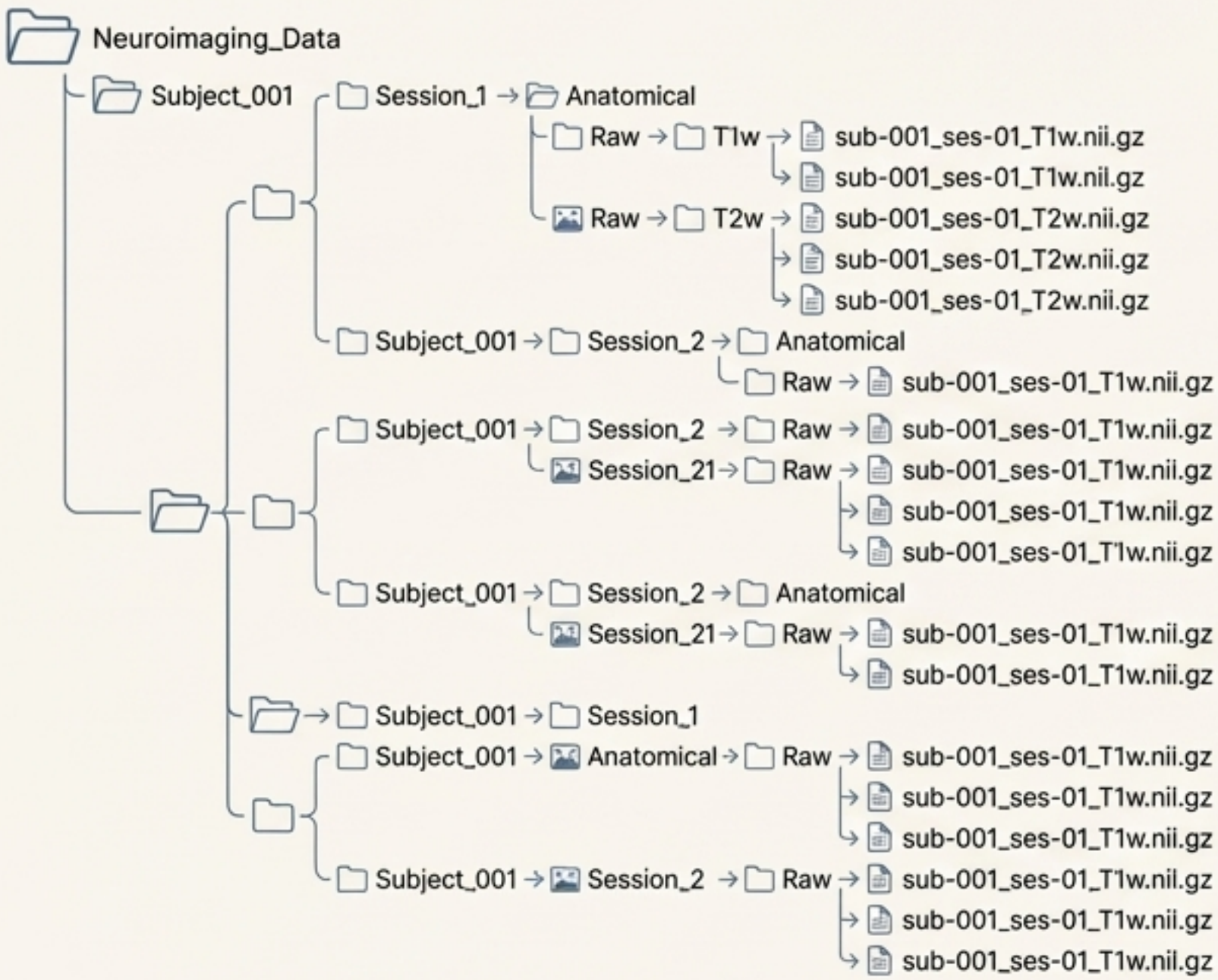

Step 4: Individual results are aggregated into one analysis-ready table.

Each modality processed by ANTsPyMM generates its own detailed output CSV. antsxmm finds all these `_mmwide.csv` files, intelligently merges them, de-duplicates columns, and sorts them into a logical order, creating a single, comprehensive “wide-format” table for the entire session.



From dozens of files to one comprehensive table.

Before



After

mmwide_merged.csv															
subject_id	T1Hier_volume	T1Hier_thickness	T1Hier_thickness	T1Hier_mets	DTI_FA	DTI_MD	DTI_RD	DTI_NT	rsfMRI_connectivity_matrix	rsfMRI_ALFF	rsfMRI_ALFF	T2Flair_volume	T2Flair_lesion_load	T2Flair_lesion	T2Flair_loss
subject_id	0.39	0.08	0.02	0.04
subject_02	0.37	0.03	0.92	0.09
subject_03	0.46	0.02	0.57	0.08
subject_04	0.54	0.02	0.29	0.37
subject_05	0.25	0.02	0.26	0.39
subject_06	0.54	0.03	0.26	0.54
subject_07	0.86	0.03	0.03	0.05
subject_08	0.70	0.68	0.22	0.88
...

Ready for statistical analysis in R, Python, or your tool of choice.

Get started in seconds.

``antsxmm`` is packaged for easy installation via pip.

A terminal window with a light gray title bar containing three white circular window control buttons. The main area is white and contains two lines of text: a comment line starting with a hash and a command line starting with a dollar sign.

```
# Install the package from the repository root  
$ pip install .
```

The installation handles all required dependencies, including ``antspymm``, ``pandas``, and ``click``.

Run a complete analysis with a single command.

```
antsxmm BIDS_TEST/ PROCESSED_OUTPUT/ --project PPMI  
--participant-label sub-211239
```

The command-line
entry point.

[Required]
Path to your BIDS
input directory.

[Required]
Where to save
all results.

A name for your
study, used in
output filenames.

[Optional]
Process only a
specific subject.

Resources & Acknowledgements

`antsxmm` is built on the shoulders of giants. We encourage you to explore the powerful libraries that make this workflow possible.

Explore the Code



github.com/ANTsX/ANTsXMM

Core Dependencies



This tool is a wrapper for the powerful **ANTsPyMM** library, which leverages the core **ANTs** toolkit for state-of-the-art image processing.