Spring 2021 | COCOAN 101

Preprocessing and data QC 2

Nipype (4) : Nipype Basic - Nodes

Sungwoo Lee

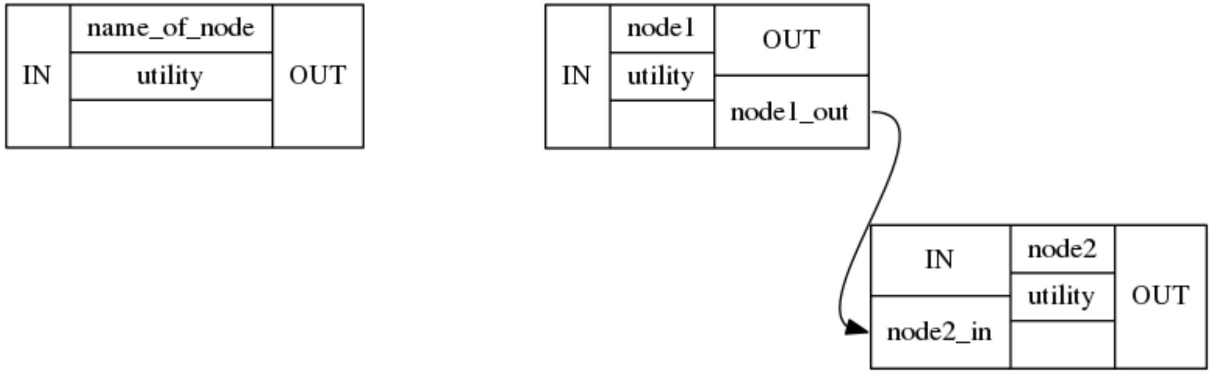
M.S-Ph.D. student

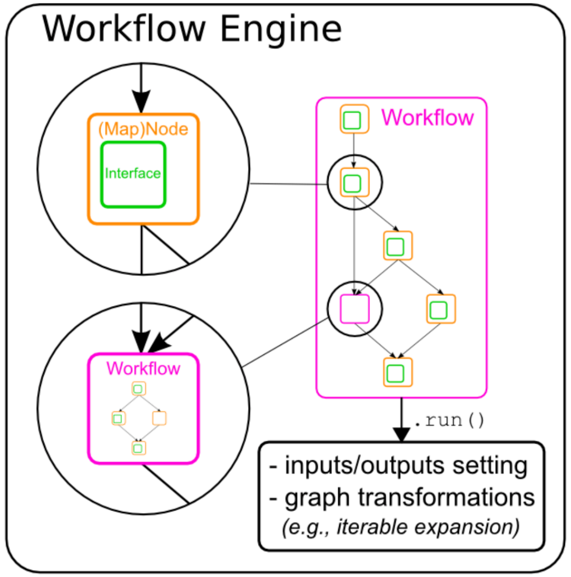
Computational Cognitive Affective Neuroscience Lab (Cocoan Lab) https://cocoanlab.github.io

Nipype (4) : Nipype Basic - Nodes

Nodes

In Nipype, a node is an object that executes a certain function. This function can be anything from a Nipype interface to a user-specified function or an external script. Each node consists of a name, an interface category and at least one input field, and at least one output field.



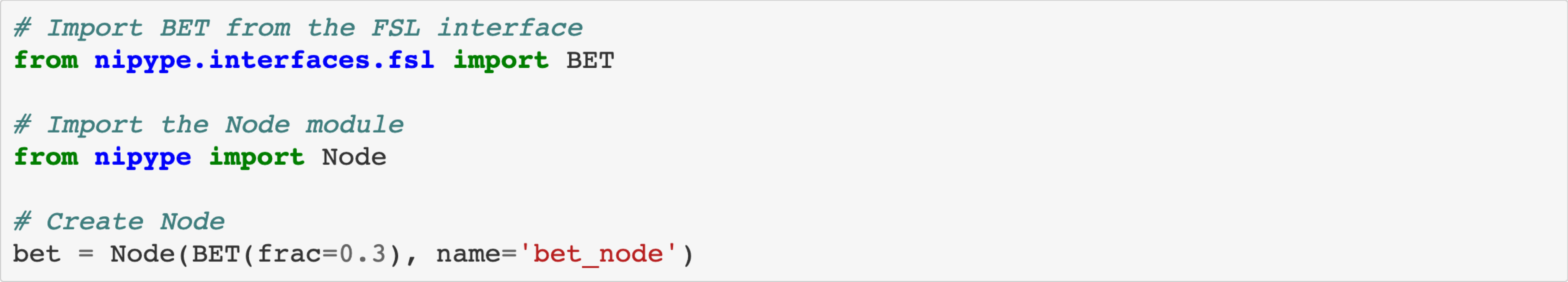


https://miykael.github.io/nipype_tutorial/notebooks/basic_nodes.html

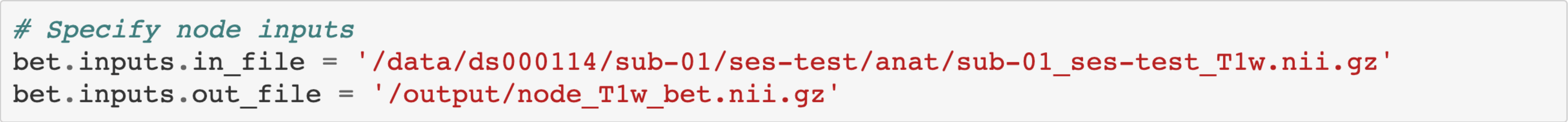
Nipype (4) : Nipype Basic - Nodes

Nodes : Example

- FSL : BET(Brain Extraction Tool)



```
# Import BET from the FSL interface
from nipype.interfaces.fsl import BET

# Import the Node module
from nipype import Node

# Create Node
bet = Node(BET(frac=0.3), name='bet_node')
```



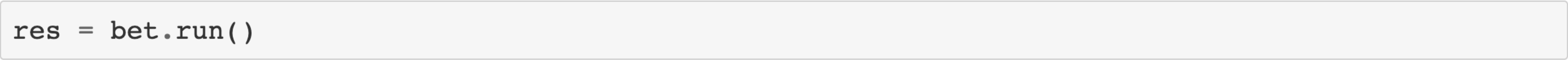
```
# Specify node inputs
bet.inputs.in_file = '/data/ds000114/sub-01/ses-test/anat/sub-01_ses-test_T1w.nii.gz'
bet.inputs.out_file = '/output/node_T1w_bet.nii.gz'
```

https://miykael.github.io/nipype_tutorial/notebooks/basic_nodes.html

Nipype (4) : Nipype Basic - Nodes

Nodes : Example

- FSL : BET(Brain Extraction Tool)



```
res = bet.run()
```

Result:

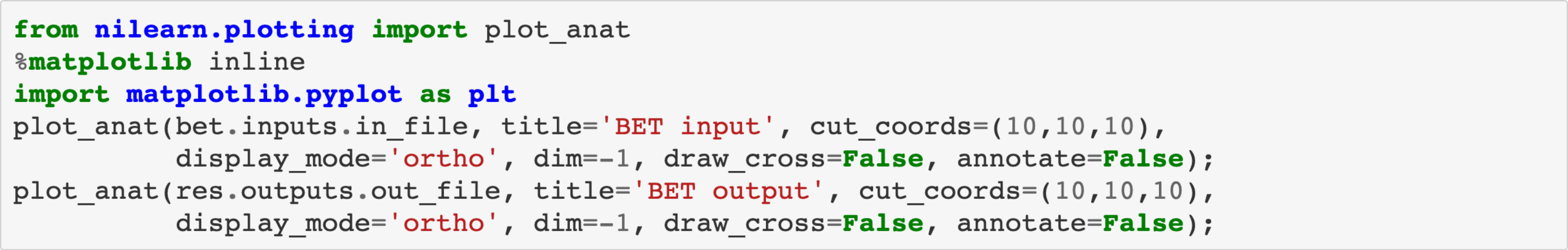
```
180514-09:27:40,948 workflow INFO:
         [Node] Setting-up "bet_node" in "/tmp/tmpr9t5iltq/bet_node".
180514-09:27:40,955 workflow INFO:
         [Node] Running "bet_node" ("nipype.interfaces.fsl.preprocess.BET"), a CommandLine Interface
with command:
bet /data/ds000114/sub-01/ses-test/anat/sub-01_ses-test_T1w.nii.gz /output/node_T1w_bet.nii.gz -f 0.30
180514-09:27:43,894 workflow INFO:
         [Node] Finished "bet_node".
```

https://miykael.github.io/nipype_tutorial/notebooks/basic_nodes.html

Nipype (4) : Nipype Basic - Nodes

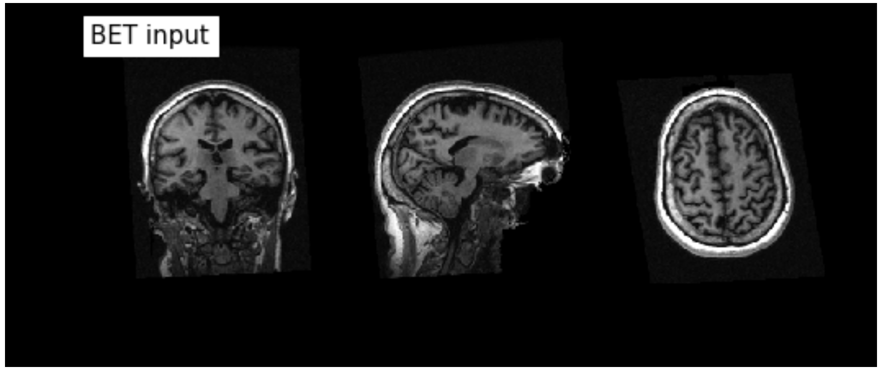
Nodes : Example

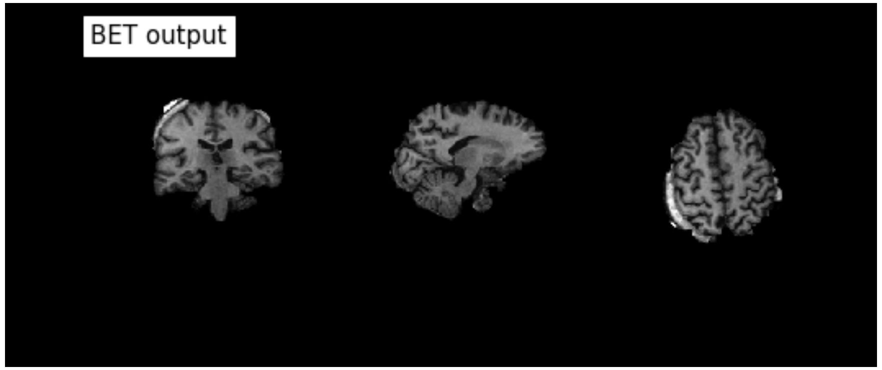
- FSL : BET(Brain Extraction Tool)



```
from nilearn.plotting import plot_anat
%matplotlib inline
import matplotlib.pyplot as plt
plot_anat(bet.inputs.in_file, title='BET input', cut_coords=(10,10,10),
          display_mode='ortho', dim=-1, draw_cross=False, annotate=False);
plot_anat(res.outputs.out_file, title='BET output', cut_coords=(10,10,10),
          display_mode='ortho', dim=-1, draw_cross=False, annotate=False);
```

Result:



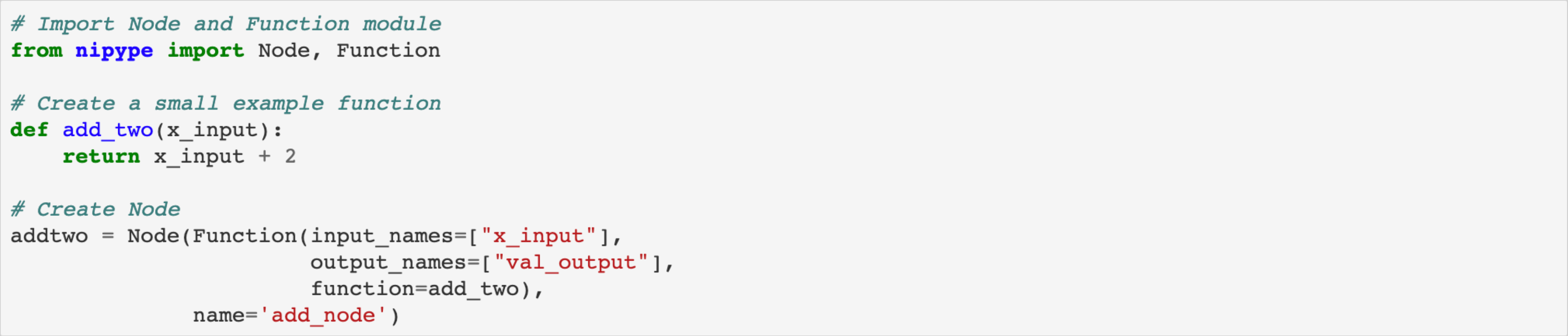


https://miykael.github.io/nipype_tutorial/notebooks/basic_nodes.html

Nipype (4) : Nipype Basic - Nodes

Nodes : Example

- Manual function



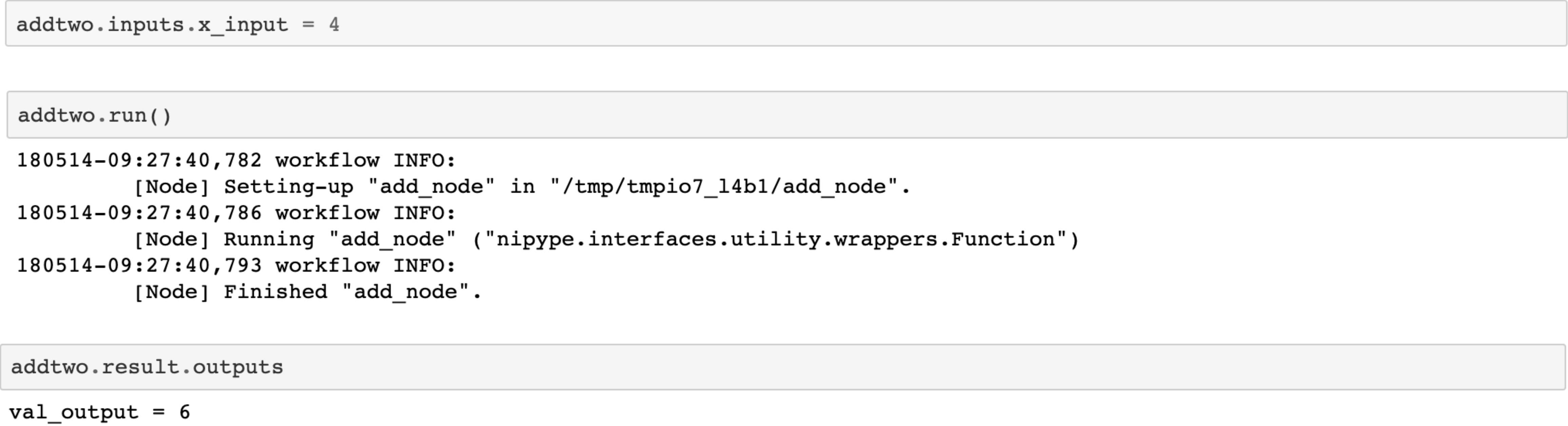
```
# Import Node and Function module
from nipype import Node, Function

# Create a small example function
def add_two(x_input):
    return x_input + 2

# Create Node
addtwo = Node(Function(input_names=["x_input"],
                       output_names=["val_output"],
                       function=add_two),
              name='add_node')
```

https://miykael.github.io/nipype_tutorial/notebooks/basic_nodes.html

Nipype (4) : Nipype Basic - Nodes

Nodes : Example

- Manual function



```
addtwo.inputs.x_input = 4
```

```
addtwo.run()
```

```
180514-09:27:40,782 workflow INFO:
         [Node] Setting-up "add_node" in "/tmp/tmpio7_l4b1/add_node".
180514-09:27:40,786 workflow INFO:
         [Node] Running "add_node" ("nipype.interfaces.utility.wrappers.Function")
180514-09:27:40,793 workflow INFO:
         [Node] Finished "add_node".
```

```
addtwo.result.outputs
```

```
val_output = 6
```

https://miykael.github.io/nipype_tutorial/notebooks/basic_nodes.html

Spring 2021 | COCOAN 101

Preprocessing and data QC 2

Nipype (4) : Nipype Basic - Nodes

Sungwoo Lee

M.S-Ph.D. student