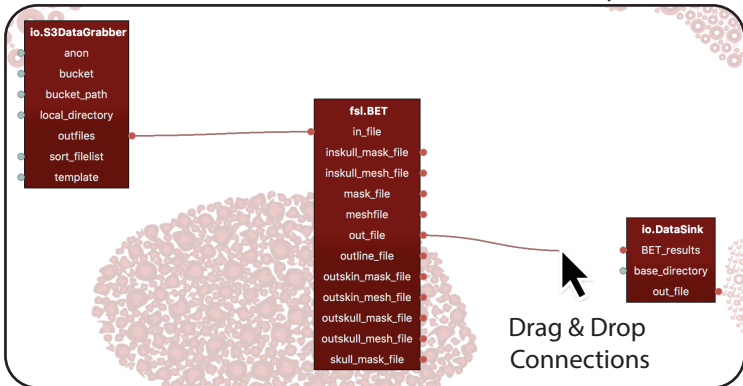


## Workflow Editor

.pork file



## Generated Code

.py file

```
#This is a NiType generator. Warning, here be dragons.
import nipype
import nipype.pipeline as pe
import nipype.interfaces.io as io
import nipype.interfaces.fsl as fsl

WorkingDirectory = '~/Huggingpipelines/ThisStudy'

#Generic datagrabber module that wraps around glob in an
NodeHash_610000484510 = pe.Node(interface = io.S3DataGrabber(), name = 'NodeHash_610000484510')
NodeHash_610000484510.inputs.anon = True
NodeHash_610000484510.inputs.bucket = 'openneuro'
NodeHash_610000484510.inputs.bucket_path = 'de000101/de000101_X2.0.0/uncompressed/'
NodeHash_610000484510.inputs.local_directory = '/tmp'
NodeHash_610000484510.inputs.sort_filelist = True
NodeHash_610000484510.inputs.template = 'sub-01/anat/sub-01_T1w.nii.gz'

#Wrap command **bet**
NodeHash_60800049f220 = pe.Node(interface = fsl.BET(), name = 'NodeHash_60800049f220')

#Generic datasink module to store structured outputs
NodeHash_610000495fe0 = pe.Node(interface = io.DataSink(), name = 'NodeHash_610000495fe0')
NodeHash_610000495fe0.inputs.base_directory = '/tmp'

#Create a workflow to connect all these nodes
analysiaflow = nipype.Workflow('MyWorkflow')
analysiaflow.connect(NodeHash_60800049f220, 'out_file', NodeHash_610000495fe0, 'BET_results')
analysiaflow.connect(NodeHash_610000484510, 'outfiles', NodeHash_60800049f220, 'in_file')

#Run the workflow
analysiaflow.run()
```