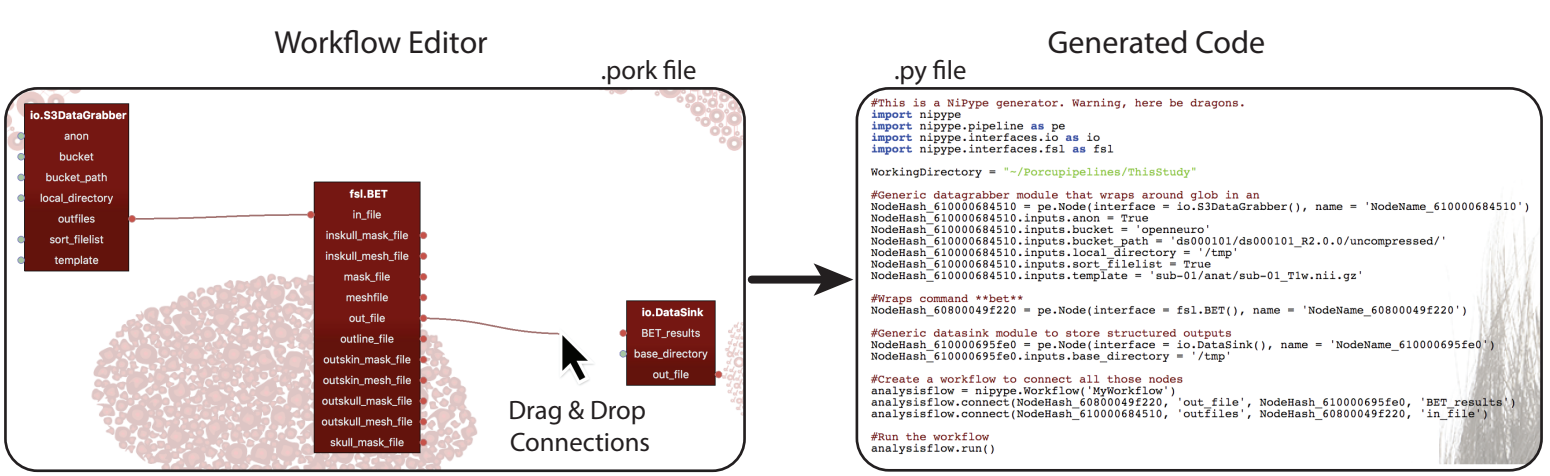
Workflow Editor
.pork file
io.S3DataGrabber
anon
bucket
bucket_path
local_directory
outfiles
sort_filelist
template
fsl.BET
in_file
inskull_mask_file
inskull_mesh_file
mask_file
meshfile
out_file
outline_file
outskin_mask_file
outskin_mesh_file
outskull_mask_file
outskull_mesh_file
skull_mask_file
io.DataSink
BET_results
base_directory
out_file
Drag & Drop
Connections
Generated Code
.py file
#This is a NiPype 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 = "~/Porcupipelines/ThisStudy"
#Generic datagrabber module that wraps around glob in an
NodeHash_610000684510 = pe.Node(interface = io.S3DataGrabber(), name = 'NodeName_610000684510')
NodeHash_610000684510.inputs.anon = True
NodeHash_610000684510.inputs.bucket = 'openneuro'
NodeHash_610000684510.inputs.bucket_path = 'ds000101/ds000101_R2.0.0/uncompressed/'
NodeHash_610000684510.inputs.local_directory = '/tmp'
NodeHash_610000684510.inputs.sort_filelist = True
NodeHash_610000684510.inputs.template = 'sub-01/anat/sub-01_T1w.nii.gz'
#Wraps command **bet**
NodeHash_60800049f220 = pe.Node(interface = fsl.BET(), name = 'NodeName_60800049f220')
#Generic datasink module to store structured outputs
NodeHash_610000695fe0 = pe.Node(interface = io.DataSink(), name = 'NodeName_610000695fe0')
NodeHash_610000695fe0.inputs.base_directory = '/tmp'
#Create a workflow to connect all those nodes
analysisflow = nipype.Workflow('MyWorkflow')
analysisflow.connect(NodeHash_60800049f220, 'out_file', NodeHash_610000695fe0, 'BET_results')
analysisflow.connect(NodeHash_610000684510, 'outfiles', NodeHash_60800049f220, 'in_file')
#Run the workflow
analysisflow.run()