



Virtual Earthquake and seismology Research Community e-science environment in Europe  
Project 283543 – FP7-INFRASTRUCTURES-2011-2 – [www.verce.eu](http://www.verce.eu) – [info@verce.eu](mailto:info@verce.eu)

# dispel4py + misfit

(dispel4py training)  
day 3

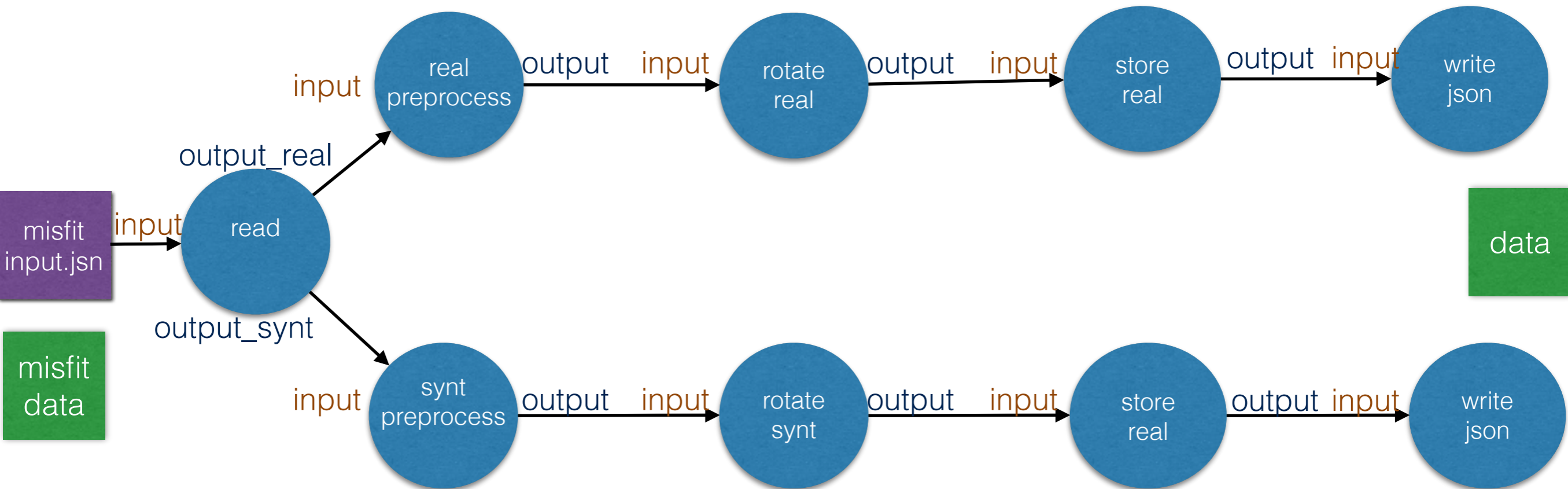
3 July 2015, Liverpool



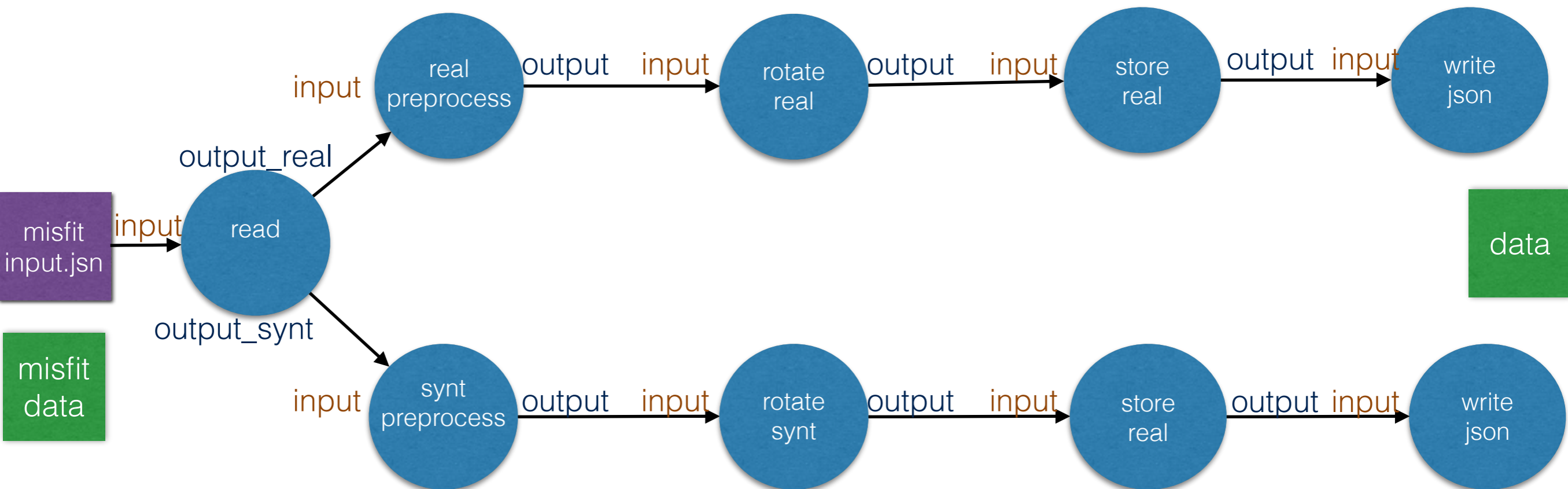
# Installation

- This is all you need:
  - **conda install -c obspy obspy**
  - **pip install dispel4py**
  - **pip install pyflex**

# Misfit Preprocess

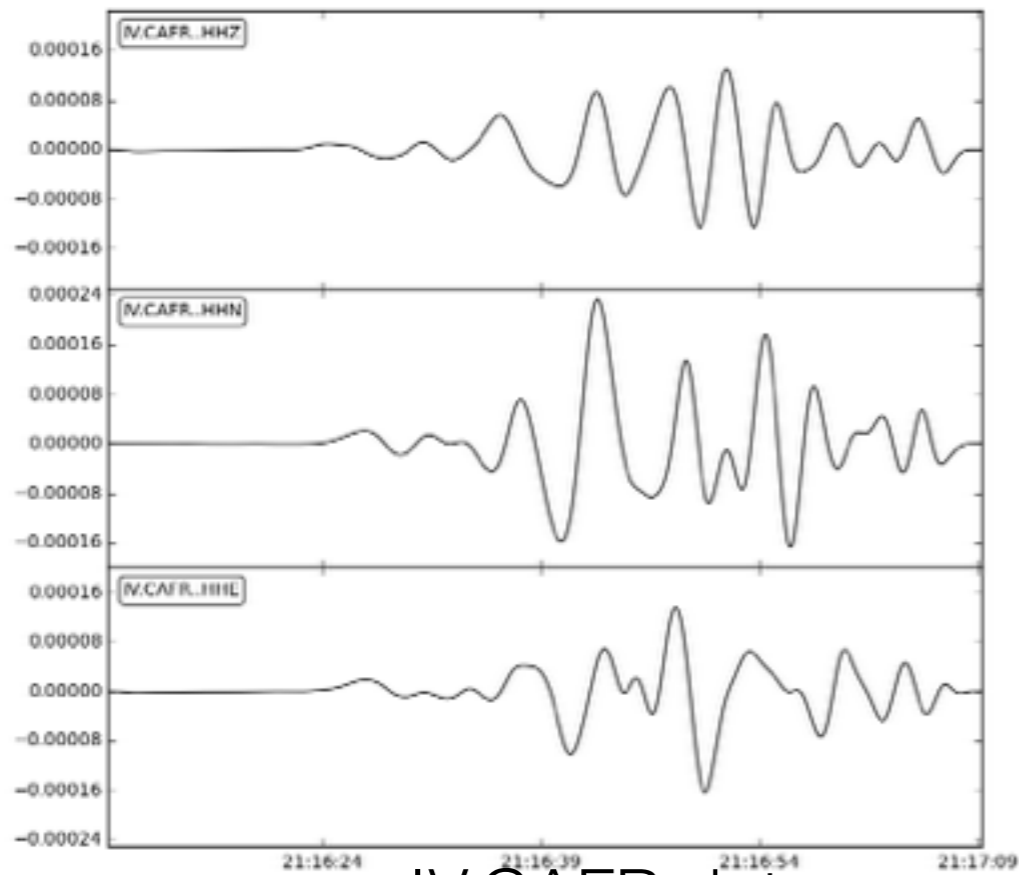


# Misfit Preprocess

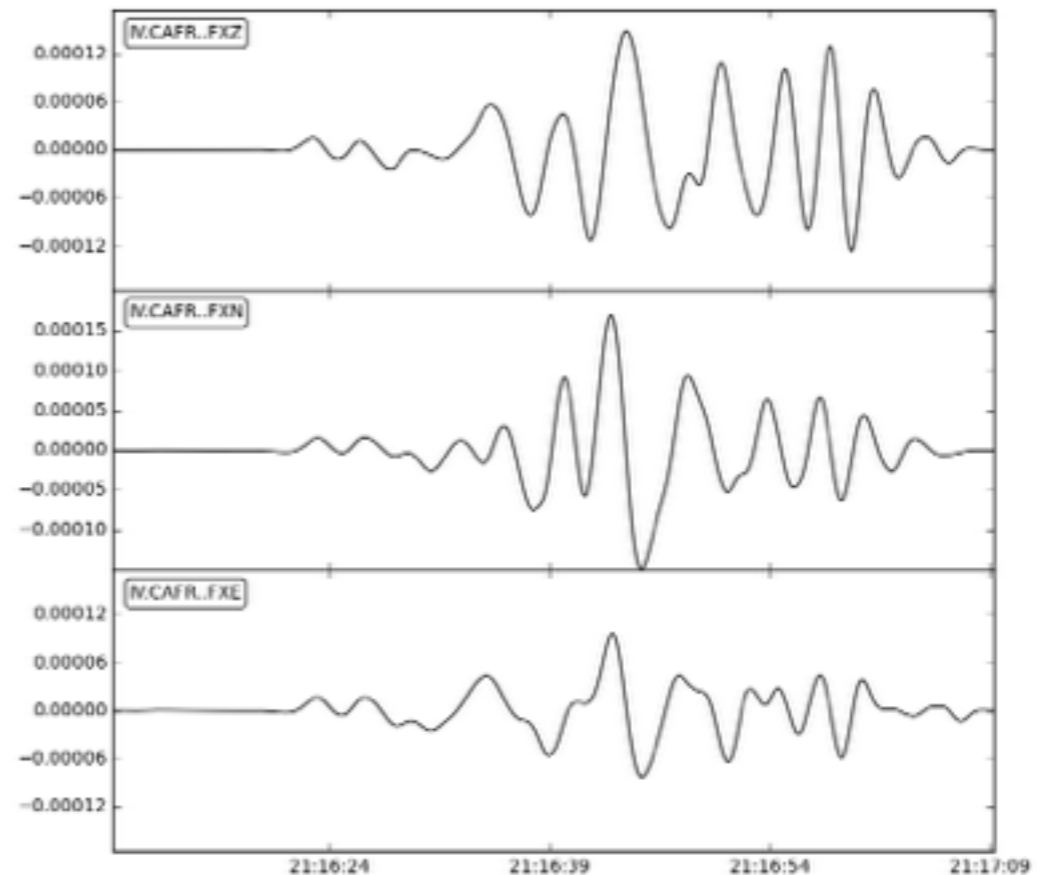


```
graph = WorkflowGraph()
```

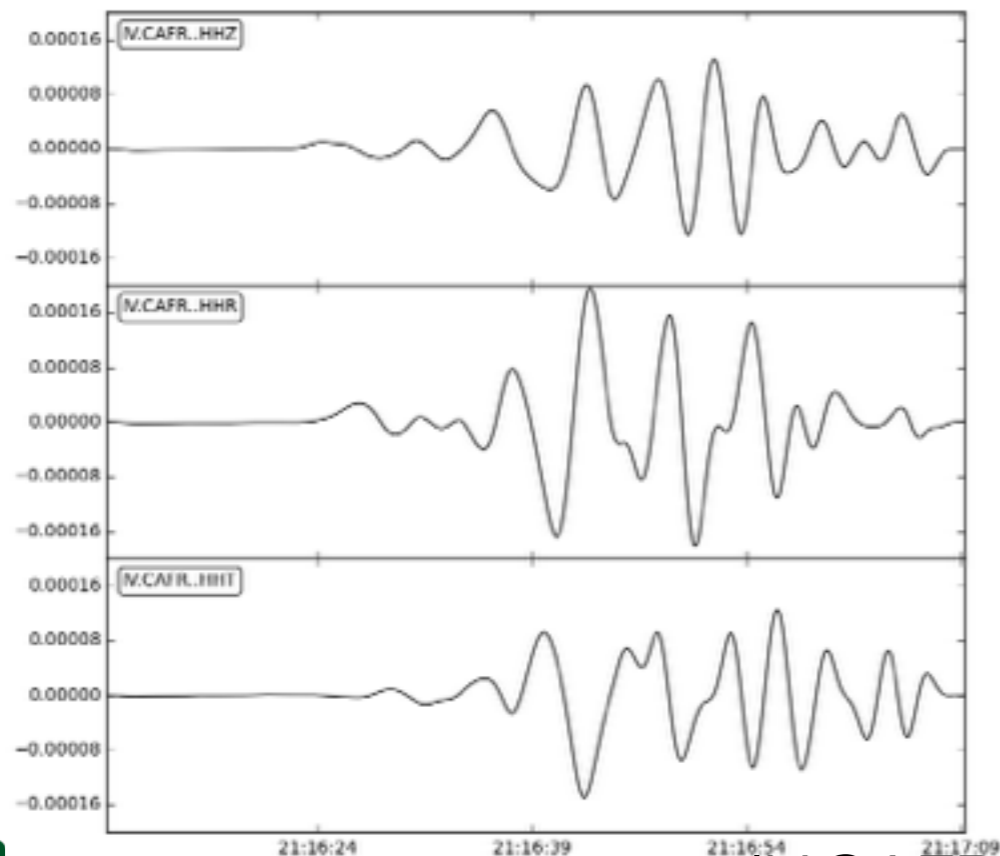
```
graph.connect(read, 'output_real', real_preprocess, 'input')
graph.connect(read, 'output_synt', synt_preprocess, 'input')
graph.connect(real_preprocess, 'output', rotate_real, 'input')
graph.connect(synt_preprocess, 'output', rotate_synt, 'input')
graph.connect(rotate_real, 'output', store_real, 'input')
graph.connect(rotate_synt, 'output', store_synt, 'input')
graph.connect(store_synt, 'output', writejson, 'input')
graph.connect(store_real, 'output', writejson, 'input')
```



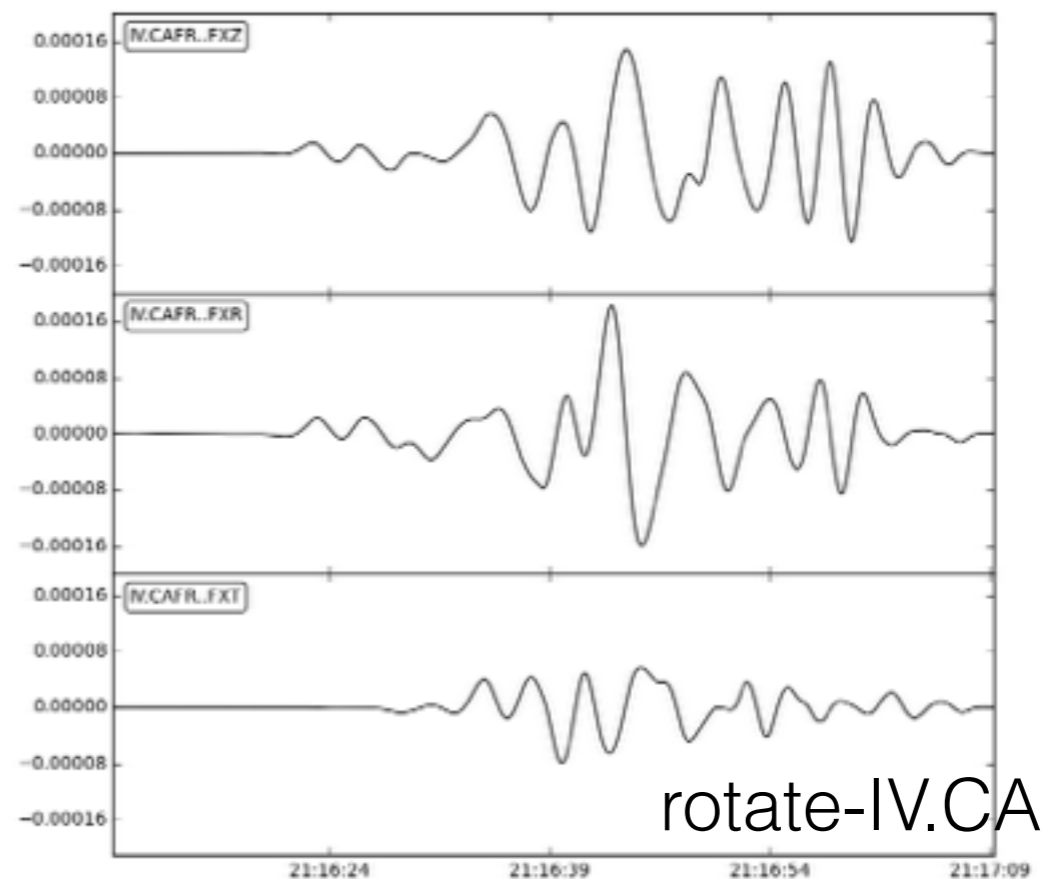
IV.CAFR.data



IV.CAFR.synth

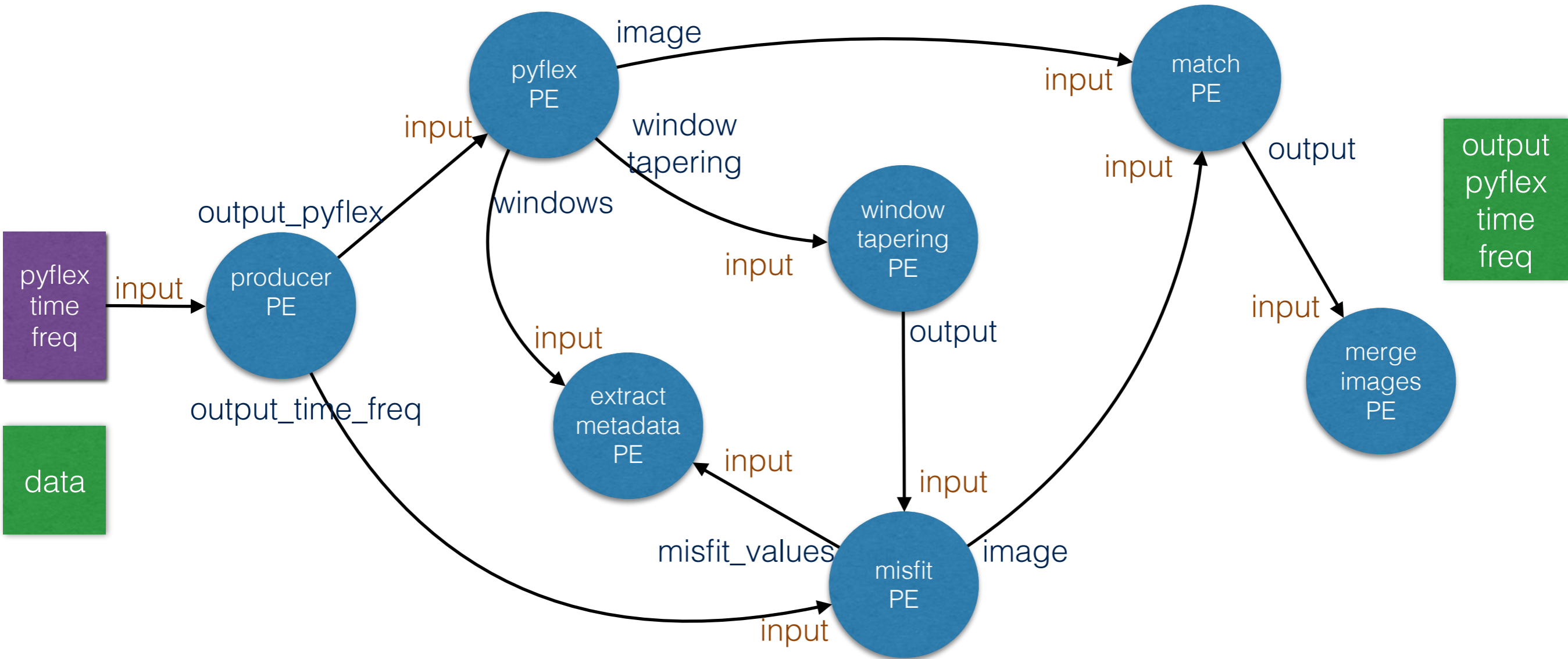


rotate-IV.CAFR.data

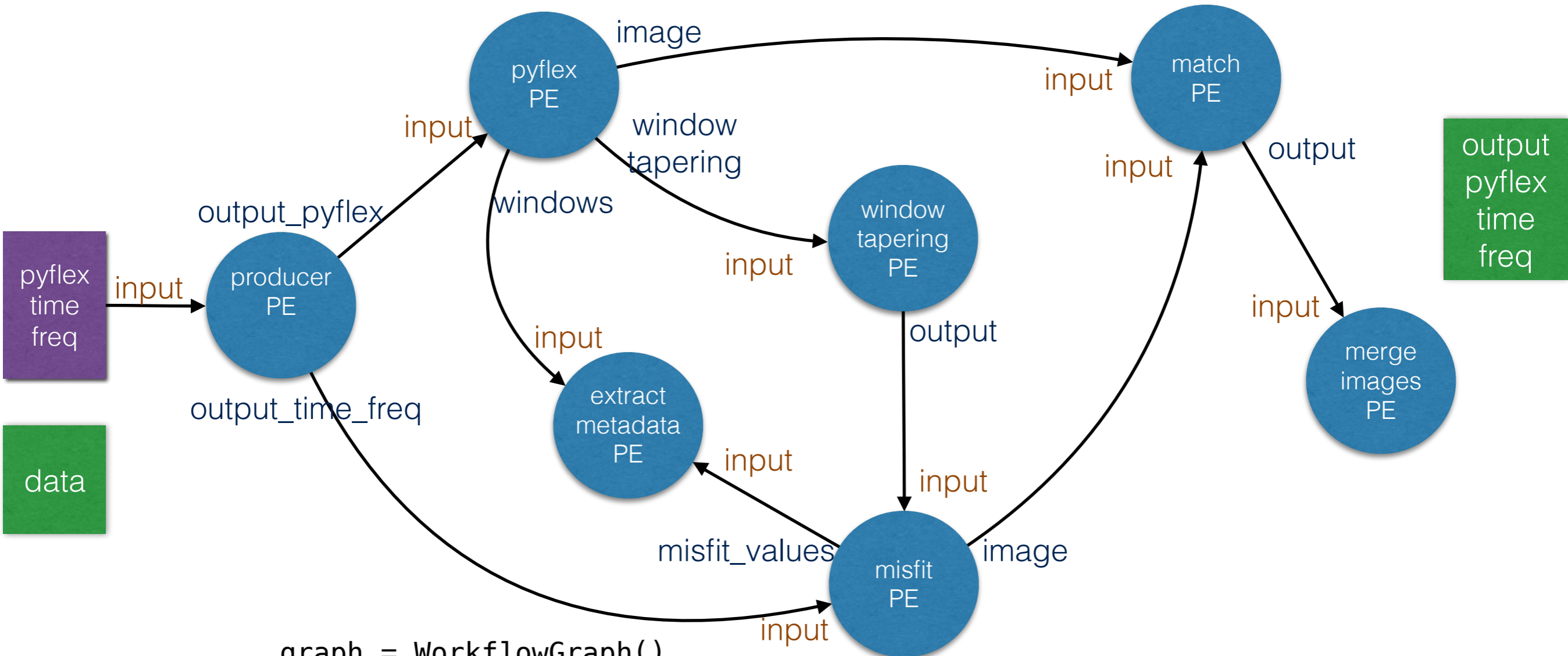


rotate-IV.CAFR.synth

# Misfit Postprocess



# Misfit Postprocess

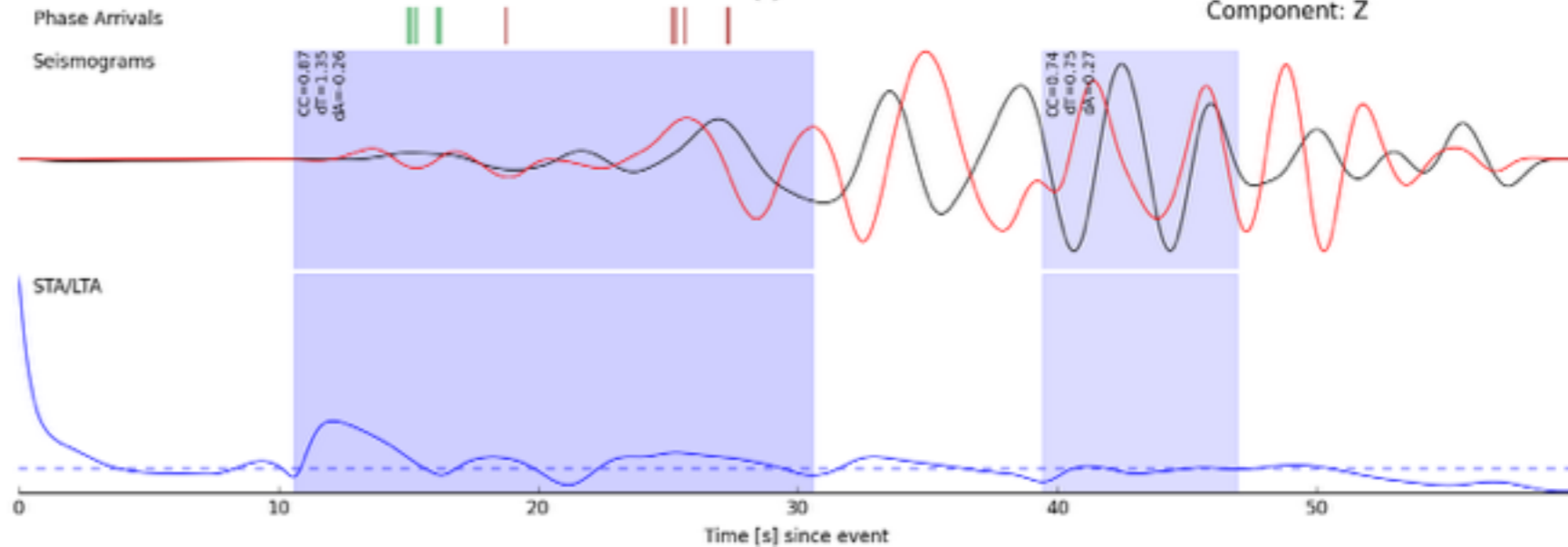
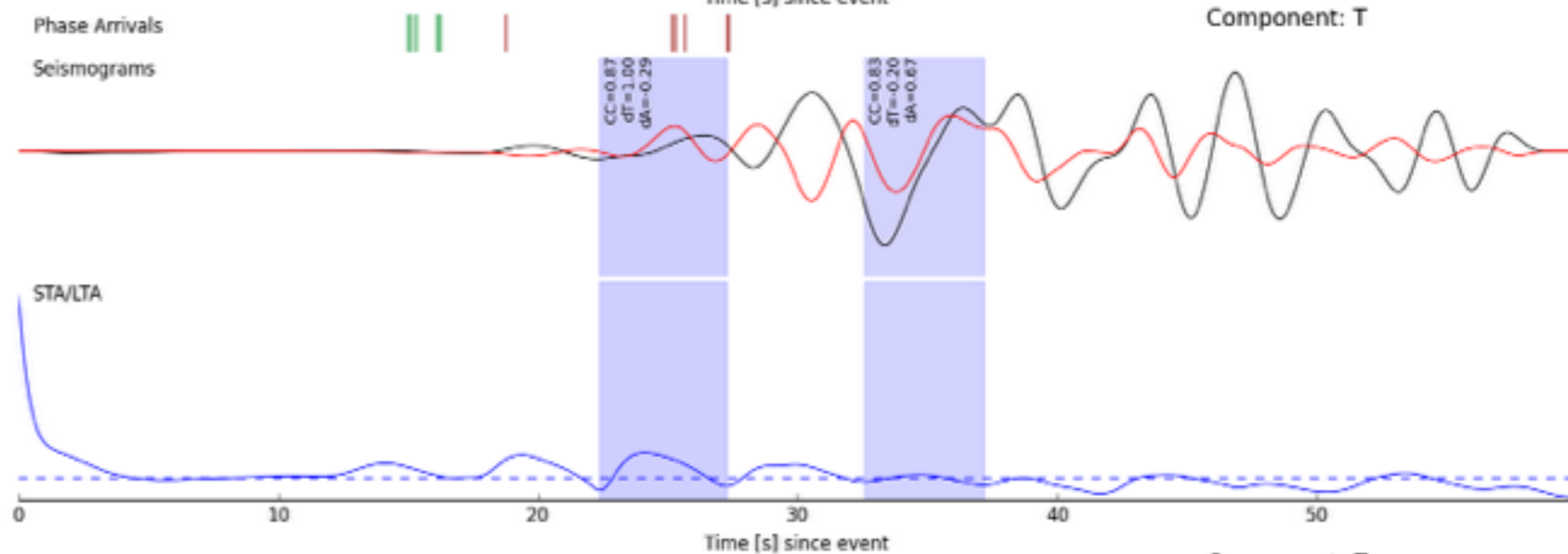
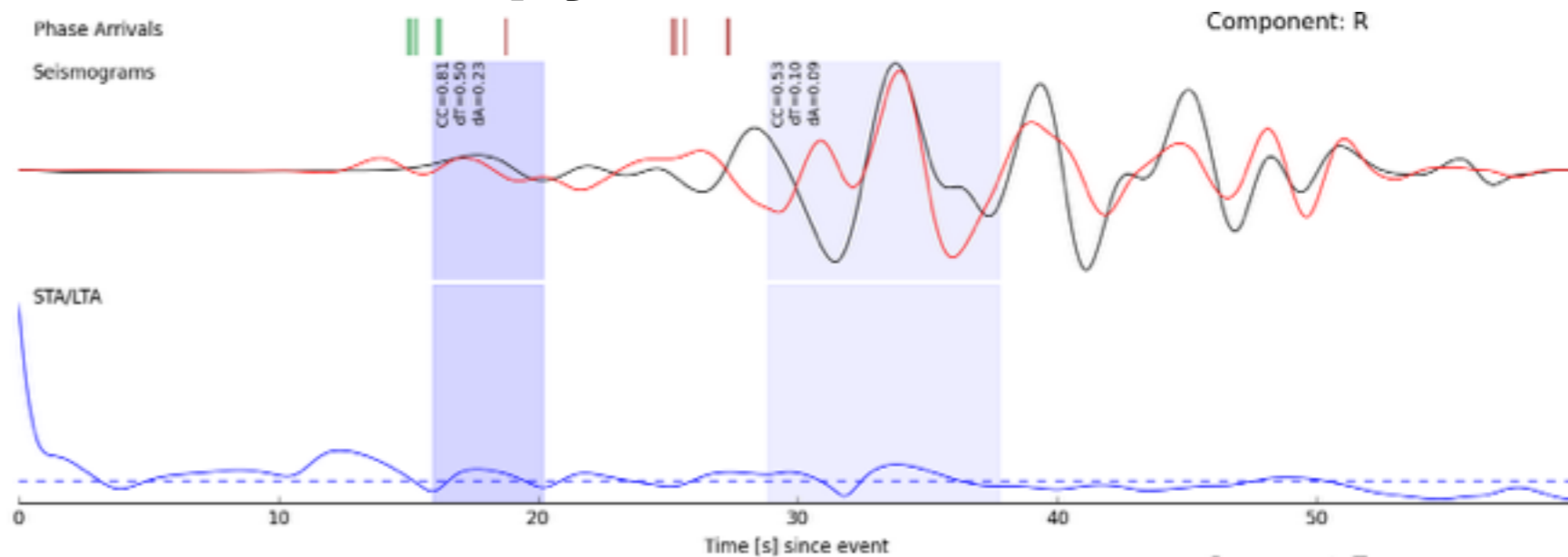
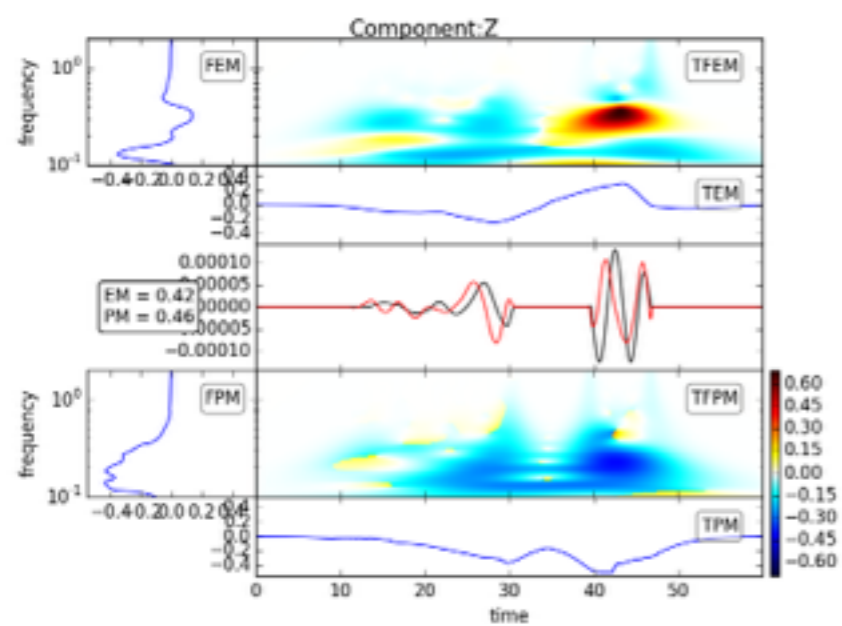
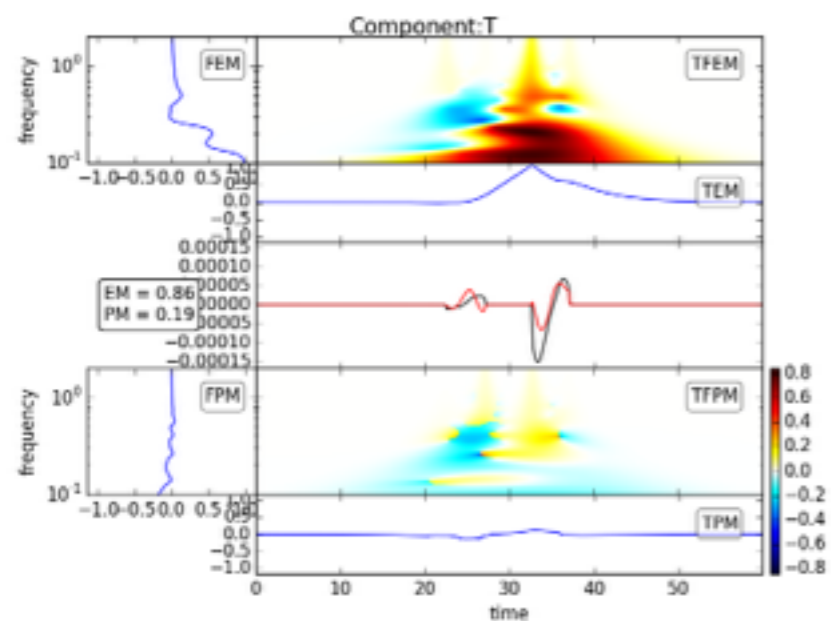
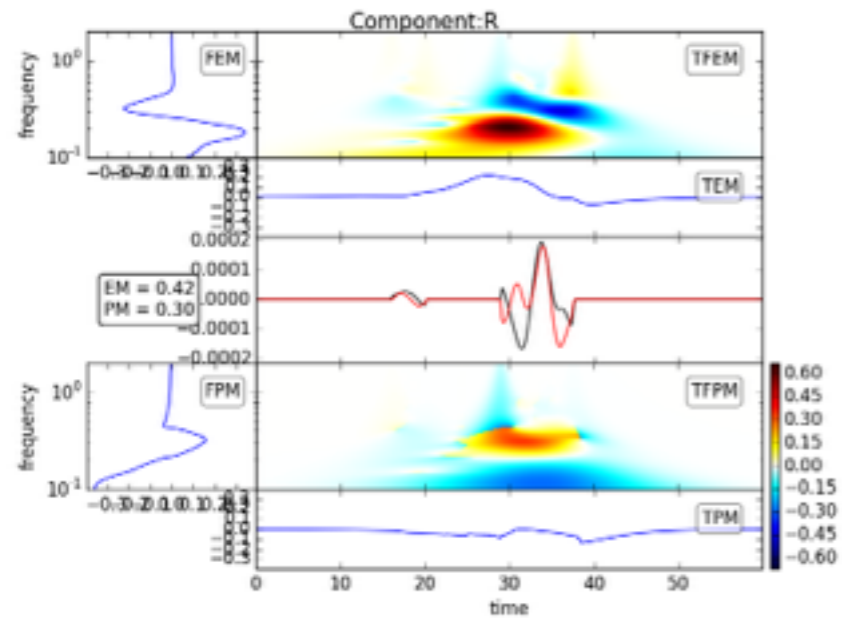


```

graph = WorkflowGraph()
graph.connect(producer_PE, "output_pyflex", pyflex_PE, "input")
graph.connect(producer_PE, "output_time_frequency", misfit_PE, "input")
graph.connect(pyflex_PE, "image", match_PE, "input")
graph.connect(pyflex_PE, "window_tapering", window_tapering_PE, "input")
graph.connect(pyflex_PE, "windows", extract_metadata_PE, "input")
graph.connect(window_tapering_PE, "output", misfit_PE, "input")
graph.connect(misfit_PE, "misfit_values", extract_metadata_PE, "input")
graph.connect(misfit_PE, "image", match_PE, "input")
graph.connect(match_PE, "output", merge_images_PE, "input")
  
```

# time frequency

# pyflex windows





# Run misfit

- preprocess (./run\_preprocess\_misfit.sh ):
  - dispel4py simple misfit\_preprocess.py -f misfit\_input.json
- postprocess (./run\_pyflex\_and\_time\_frequency.sh):
  - dispel4py simple misfit\_postprocess.py -f pyflex\_and\_time\_frequency.json