FragPipe in Galaxy: Wrapper Implementation for a Complete Proteomics Tool Suite

Reid Wagner, James Johnson, Fengchao Yu, Subina Mehta, Pratik Jagtap, Alexey I. Nesvizhskii, Timothy J. Griffin

Galaxy Community Conference
July 18, 2022
FragPipe

- FragPipe is a software suite for spectrometry-based proteomics analysis.
  - Incorporates MS-Fagger, Philosopher, Crystal-C, PTM-Shephard, TMT-Integrator, IonQuant, EasyPQP, DIA-Umpire.
- Maintained by Dr. Fengchao Yu and Alexey Nesvizhskii’s lab at the University of Michigan.
- FragPipe includes both Java GUI and CLI headless mode.
FragPipe (v1.5.0)

Workflows
FragPipe and its collection of tools support multiple proteomic workflows. Select and load an option from the dropdown menu below to configure all the tools. Workflows can be customized and saved. Also see the tutorial.

Select an option to load config for: TMT10-phospho-bridge ▼ or save current settings as workflow Save Open in File Manager

TMT 10-plex workflow for phosphopeptide enriched data, with quantification from MS2. PTMProphet for site localization. TMT-Integrator with Bridge channel (labeled as 'pool' in the annotation files), median-centering normalization, data summarization at the gene/protein/peptide/site levels.

Global settings
RAM (GB, 0=auto) 0 ▼ Parallelism ▼

Input LC/MS Files
MS data type ▼ Regular MS ▼ IM-MS (ion mobility, timsTOF only)
Add files ▼ Add folder recursively ▼ Remove selected files ▼ Clear files ▼ Add recent ▼
Save as manifest ▼ Load manifest ▼

Assign files to Experiments/Groups (select rows to activate action buttons):
Consecutive By parent directory By file name Set experiment/replicate Clear groups

Path (can drag & drop from Explorer)
E: \hayse\MayInstitute\TMT\tmt-phospho-raw\TMT-P-01s\01OPTAC_CCRCC_P34U_20171106_LUMOS_F01.mzML
E: \hayse\MayInstitute\TMT\tmt-phospho-raw\TMT-P-01s\02OPTAC_CCRCC_P34U_20171106_LUMOS_F02.mzML
E: \hayse\MayInstitute\TMT\tmt-phospho-raw\TMT-P-02s\01OPTAC_CCRCC_P34U_20171108_LUMOS_F01.mzML
E: \hayse\MayInstitute\TMT\tmt-phospho-raw\TMT-P-02s\02OPTAC_CCRCC_P34U_20171108_LUMOS_F02.mzML

Experiment (can be empty, alphanumeric...)
TMT-P-01s
TMT-P-01s
TMT-P-02s
TMT-P-02s

Replicate (can be empty and integer)
Sample grouping
Sample grouping
### Sample grouping

**Input LC/MS Files**

- **MS data type**: Regular MS
- **Path (can drag & drop from Explorer)**
  - E:\hayse\MayInstitute\TMT\tmt-phospho-raw\TMT-P-01s\1CPTAC_CCRCC_P_3HU_20171106_LUMOS_f01.mzML
  - E:\hayse\MayInstitute\TMT\tmt-phospho-raw\TMT-P-01s\1CPTAC_CCRCC_P_3HU_20171106_LUMOS_f02.mzML
  - E:\hayse\MayInstitute\TMT\tmt-phospho-raw\TMT-P-02s\02CPTAC_CCRCC_P_3HU_20171108_LUMOS_f01.mzML
  - E:\hayse\MayInstitute\TMT\tmt-phospho-raw\TMT-P-02s\02CPTAC_CCRCC_P_3HU_20171108_LUMOS_f02.mzML

**Assign files to Experiments/Groups**

- **Consecutive**
- **By parent directory**
- **By file name**
- **Set experiment/replicate**
- **Clear groups**

**Experiment (can be empty, alphanumeric...)**

1. TMT-P-01s
2. TMT-P-02s

**Replicate (can be empty and integer)**

1. 1
2. 2
Sample grouping

<table>
<thead>
<tr>
<th>Scan files format</th>
<th>mzXML</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify sample groups</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Sample Groups

<table>
<thead>
<tr>
<th>1. Sample Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sample Name</strong></td>
</tr>
<tr>
<td>Only use letters, numbers, and chars: _-</td>
</tr>
</tbody>
</table>

#### Proteomics Spectrum files

| 6: O2CPTAC_CCRC_C_P-JHU_20171008_LUMOS_F02.mzXML |
| 5: O2CPTAC_CCRC_C_P-JHU_20171008_LUMOS_F01.mzXML |
| 3: 01CPTAC_CCRC_C_P-JHU_20171008_LUMOS_F02.mzXML |
| 2: 01CPTAC_CCRC_C_P-JHU_20171008_LUMOS_F01.mzXML |

**Thermo RAW, mZML, mzXML, MGF formats**

**File name prefix**

**Names inputs: prefix_rep#.mzXML Leave blank to use History names of inputs**

**Annotation file**

| 4: TMT-P-015_annotation.txt |

---

### 2. Sample Groups

| **Sample Name** | TMT-P-025 |
|------------------|
| Only use letters, numbers, and chars: _- |

#### Proteomics Spectrum files

| 6: O2CPTAC_CCRC_C_P-JHU_20171008_LUMOS_F02.mzXML |
| 5: O2CPTAC_CCRC_C_P-JHU_20171008_LUMOS_F01.mzXML |
| 3: 01CPTAC_CCRC_C_P-JHU_20171008_LUMOS_F02.mzXML |
| 2: 01CPTAC_CCRC_C_P-JHU_20171008_LUMOS_F01.mzXML |

**Thermo RAW, mZML, mzXML, MGF formats**

**File name prefix**

**Names inputs: prefix_rep#.mzXML Leave blank to use History names of inputs**

**Annotation file**

| 7: TMT-P-025_annotation.txt |

**Insert Sample Groups**
<repeat name="samples" title="Sample Groups" min="1">
  <expand macro="sample_input" scanformat="@SCANFORMAT@
  annotation_optional="false"/>
</repeat>
Workflows

- Headless FragPipe takes a configuration file defining a (FragPipe) workflow.
- How do we build this in the wrapper?
- How do we incorporate default settings?

```bash
# Workflow: Default

crystalc.run-crystalc=false
database.decoy-tag=rev_
diann.fragpipe.cmd-opts=
diann.library=
diann.q-value=0.01
diann.quantification-strategy=0
diann.run-dia-nn=false
diaumpire.AdjustFragIntensity=true
diaumpire.BoostComplementaryIon=false
diaumpire.CorrThreshold=0
diaumpire.DeltaApex=0.2
diaumpire.ExportPrecursorPeak=false
diaumpire.Q1=true
diaumpire.Q2=true
diaumpire.Q3=true
diaumpire.RFmax=500
diaumpire.RPmax=25
diaumpire.RTOverlap=0.3
...
```
Contents are written directly to file (after being evaluated).

- File can be referenced when invoking the tool.
- Workflow may come from:
  - History
  - Default provided by FragPipe
- User options overwrite defaults.

```
<configfile name="workflow_configfile"><![CDATA[#slurp

## Create a fragpipe params dictionary from a file

... #if $wf.workflow_name == 'history'
    #set $wfpath = str($wf.workflow)
#else
    #set $wfpath = $__tool_directory__ + '/workflows/' + str($wf.workflow_name) + '.workflow'
#end if
#set $fh = open($wfpath, 'r')
#for $line in $fh:
    #if $line.startswith('#')
        #silent $comments.append($line)
    #elif $line.strip() != ''
        #set $kv = $line.strip().split('=')
        #if len($kv) < 2
            #silent $kv.append('')
        #end if
        #set $wfdict[$kv[0]] = $kv[1]
    #end if
#end for

... #for $k in sorted($wfdict.keys())
    #set $kv = $k + '=' + str($wfdict[$k])
    $kv
#end for

#slurp]]>
</configfile>
```
Future

• We’ll be working on it at CoFest!
• MSFragger and FragPipe conda recipes:
  – https://github.com/galaxyproteomics/bioconda-recipes
• Further wrapper development:
  – Implementing workflows one-by-one.
  – https://github.com/galaxyproteomics/tools-galaxyp
Thank you!