

# An Update on the FragPipe Galaxy Tool for Proteomics Workflows

Reid Wagner<sup>1</sup>, Fengchao Yu<sup>2</sup>, Subina Mehta<sup>3</sup>, Pratik Jagtap<sup>3</sup>, Alexey I. Nesvizhskii<sup>2</sup>, Timothy J. Griffin<sup>3</sup>



UNIVERSITY OF MINNESOTA

<sup>1</sup>Minnesota Supercomputing Institute, University of Minnesota; <sup>2</sup>Department of Pathology and Department of Computational Medicine and Bioinformatics, <sup>3</sup>Department of Biochemistry, Molecular Biology and Biophysics, University of Minnesota

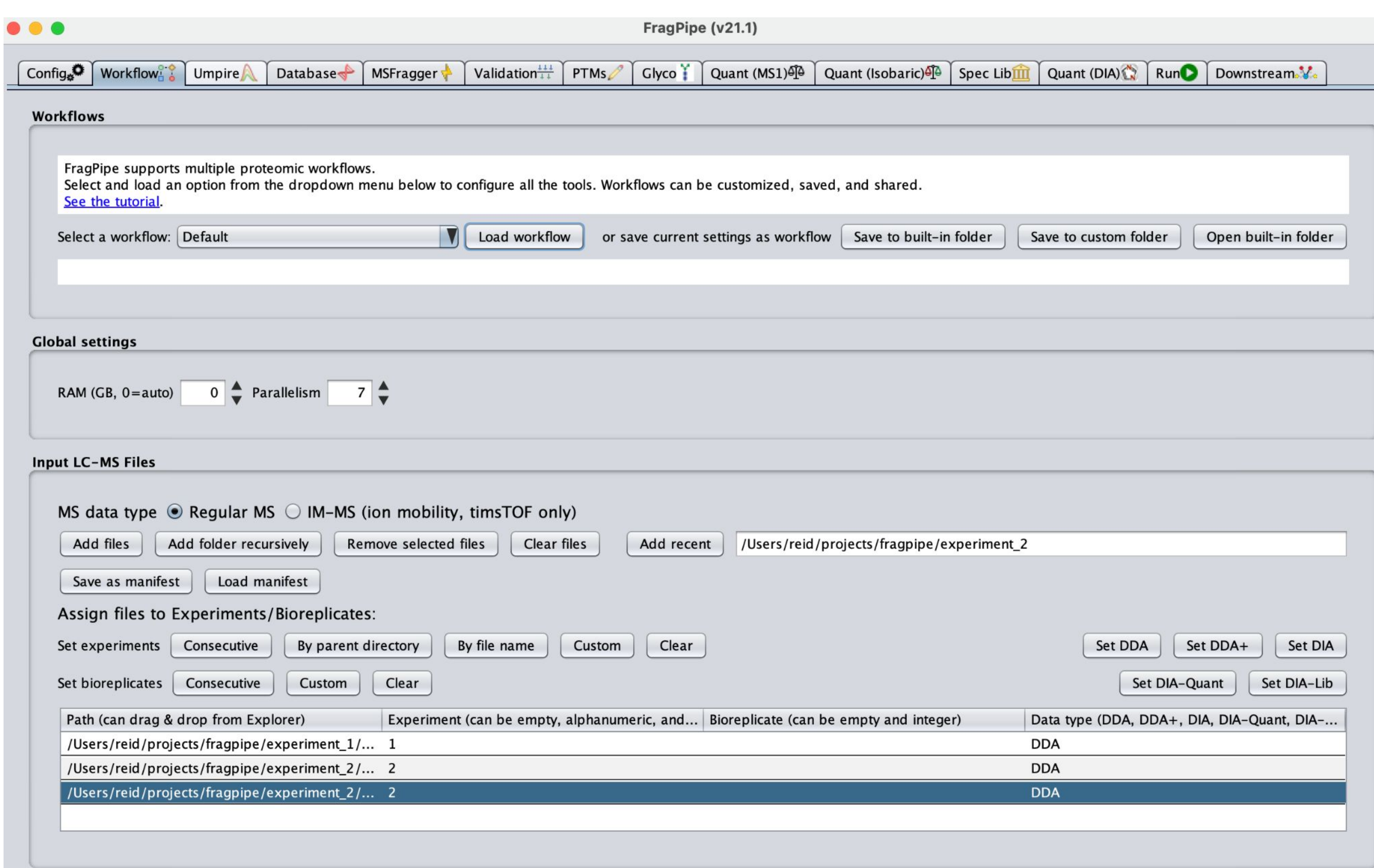


## Introduction

**FragPipe** ([fragpipe.nesvilab.org](http://fragpipe.nesvilab.org)) is a suite of tools for comprehensive analysis of mass spectrometry (MS) data. Originally developed as a GUI program, a more recent command line interface has allowed the creation of a Galaxy wrapper to expand access to the Galaxy ecosystem and HPC-backing.



Implementation of this wrapper presented various decisions for handling dependencies and replicating the GUI within the Galaxy interface. Three new Bioconda recipes - **MSFragger**<sup>1</sup>, **IonQuant**<sup>2</sup>, and **FragPipe** - were released to support the FragPipe Galaxy tool, and the FragPipe Galaxy tool has been submitted to the **tools-galaxy** repository for review. The original MSFragger, IonQuant, and FragPipe programs are maintained by Dr. Fengchao Yu and the lab of Dr. Alexey Nesvizhskii at the University of Michigan.



## Tool Implementation

- Primary input files include scan files in **mzML**, **mzXML**, or **Thermo.raw** formats, a search database in **FASTA** format, and a user-created Manifest file that includes metadata for the scan files.
  - While the Manifest file is automatically generated in the FragPipe GUI, handling the inputting of a collection of scan files, each with multiple associated fields, proved a challenge in Galaxy.
  - A TMT annotation file is required for TMT workflows.
- A user selects from five initially implemented workflows: **Default**, **Open**, **Nonspecific-HLA**, **LFQ-MBR**, **TMT-10**, and **TMT-11**, addressing the needs of most researchers. The parameters for each workflow are then made available to the user.
- The user may opt to use add decoys or decoys and contaminants to the database, and may input an integer for database splitting to reduce memory requirements.
- Workflow files provided by the FragPipe installation provide a useful canvas to construct parameters with. As seen in the figure below, the Cheetah templating in the tool XML file reads in the default workflow settings configuration for the user-selected workflow parameter, and overwrites only those parameters set by the user.
- A special **genericize\_db.py** script was included in the tool to ensure sequences headers are prefixed by an allowed subset of headers, or a controlled **>generic|** tag.

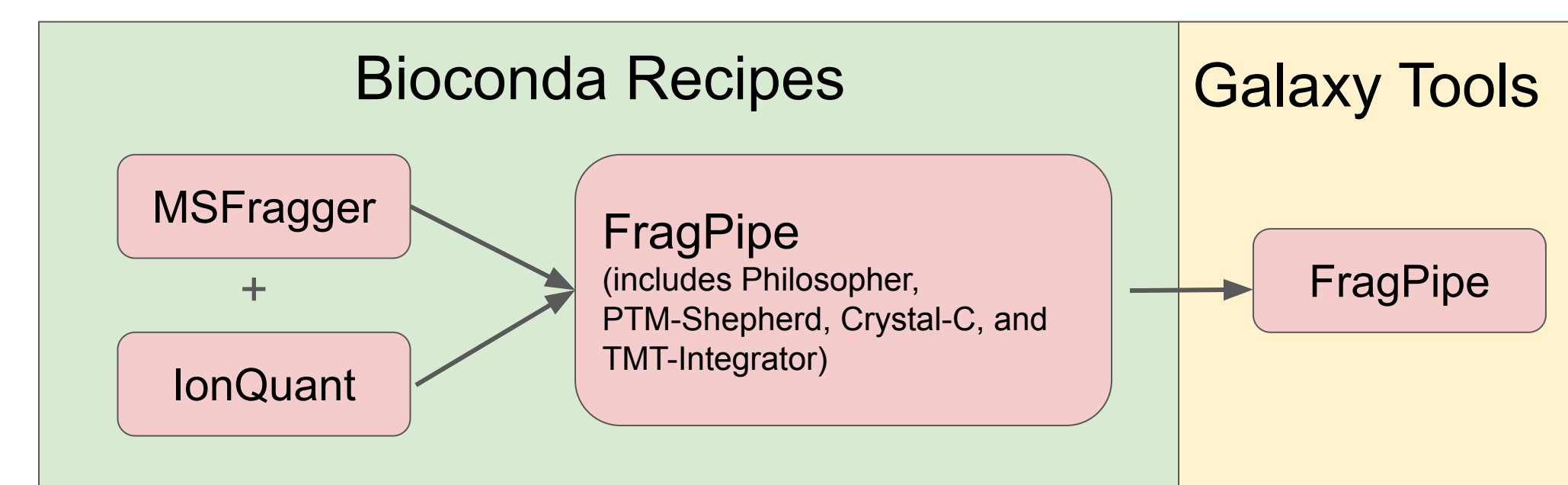
```
<configfile name="workflow_configfile"><![CDATA[#slurp
## Create a fragpipe params dictionary from a file
...
## Workflow: Default
crystalc.run-crystalc=false
database.decoy-tag=rev_
diann.fragpipe.cmd-opts=
diann.library=
diann.q-value=0.01
diann.quantification-strategy=0
diann.run-dia-nn=false
diaupire.AdjustFragIntensity=true
diaupire.BoostComplementaryIon=false
diaupire.CorrThreshold=0
diaupire.DeltaApex=0.2
diaupire.ExportPrecursorPeak=false
diaupire.Q1=true
diaupire.Q2=true
diaupire.Q3=true
diaupire.RFmax=500
diaupire.RFmax=25
diaupire.RTOverlap=0.3
...]]>
```

## Future Development

- Further workflows will be added from the existing 63 provided by FragPipe.
- Further parameters and parameter defaults will be added, as well as some software components that are not yet included due to licensing restrictions, e.g. DIA-NN.
- Implementation of MSFragger and IonQuant Galaxy tools.

## Bioconda Recipes

- MSFragger, IonQuant, and FragPipe recipes were published for the Galaxy tool. By releasing these three components separately, we open up the opportunity for standalone IonQuant and MSFragger Galaxy tools, and allow researchers to use the individual bioconda packages now.
- Additional programs and libraries are bundled within these recipes include **Philosopher**, **PTM-Shepherd**, **Crystal-C**, and **TMT-Integrator**.
- New license key functionality, described below, was added to the MSFragger and IonQuant Java programs to handle licensing in the bioconda recipes. Further flags and options were included in the FragPipe Bioconda recipe and Galaxy tool to propagate the academic license consent.

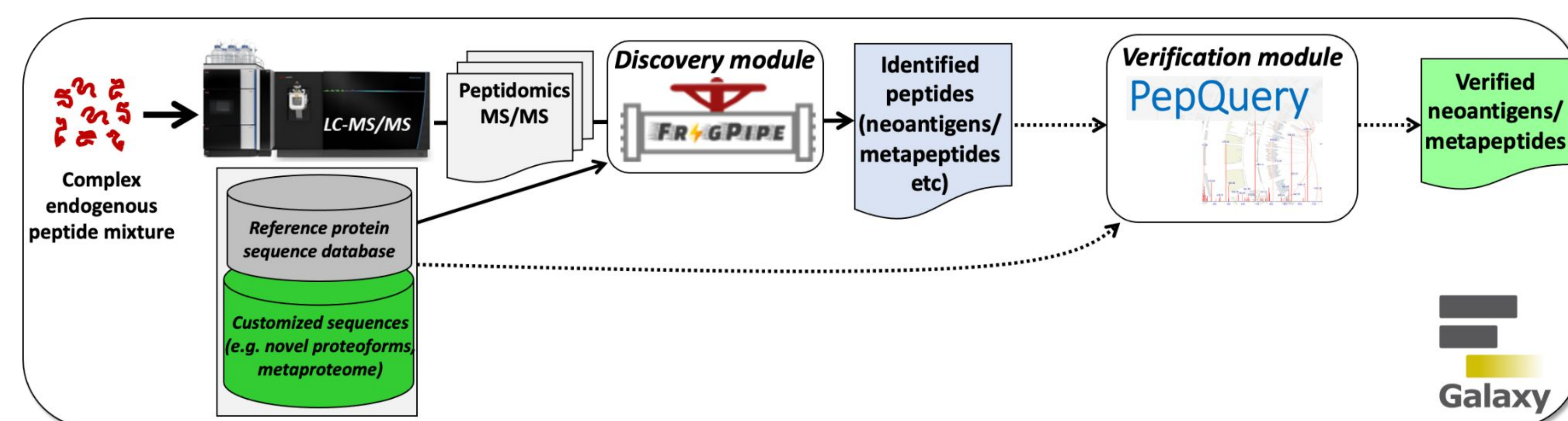


## Preliminary Results

Utilizing local HPC-backed FragPipe Galaxy installation, we searched two datasets that involved challenging non-tryptic searches - Orbitrap LC-MS/MS data from bronchoalveolar lavage fluid (BALF) endogenous peptides and immunopeptidomic data from Raji cells and a public melanoma cell dataset (PXD013649) - against large protein sequence databases. The following steps are shown in the below diagram:

- Database generation: A customized protein sequence databases is created with a variety of Galaxy tools.
- Discovery module: **FragPipe** is used to find peptide spectral matches using no enzyme settings.
- Verification module: PSMs generated by FragPipe are validated using the PepQuery2 Galaxy tool.

FragPipe was shown to be efficient and sensitive for peptidomic discovery analysis, producing 1166 peptide matches over 23 minutes. "No enzyme" analyses require significant resources, and are thus a good demonstration of the utility of HPC-backed FragPipe.



## Licensing

It was important to respect the academic-use only licensing of MSFragger and IonQuant in publishing this tool. Typically a user is expected to agree to the license conditions when downloading the MSFragger and IonQuant tools from **nesvilab.org**, so re-publishing these in Bioconda presented a challenge. Rather than making it strictly impossible run these tools without agreeing to the license, we determined it was important to ensure the user understands the limitations. To address this we implemented multiple solutions:

- License keys must be passed to the wrapper scripts for IonQuant, MSFragger, and FragPipe recipes. These keys are retrieved from the MSFragger and IonQuant download websites by reading and accepting the license terms, and are checked against the site for validity.
- A boolean parameter with a **<validator>** tag and license text was added to the FragPipe Galaxy tool. Valid keys are included in the tool XML, but a user must check the agreement parameter, shown below, to run FragPipe.
- The tool's name includes **Academic Research and Education User License (Non-Commercial)** to identify the conditions in the case where it is re-run or is used in a workflow.

**! You must agree to the MSFragger license!**  
**I understand that these tools, including MSFragger, IonQuant, Bruker, and Thermo Raw File Reader, are available freely for academic research and educational purposes only, and agree to the following terms.**

No

## References

- Kong, A. T., Leprevost, F. V., Avtonomov, D. M., Mellacheruvu, D., & Nesvizhskii, A. I. (2017). MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry-based proteomics. *Nature Methods*, 14(5), 513-520.
- Yu, F., Haynes, S. E., & Nesvizhskii, A. I. (2021). IonQuant enables accurate and sensitive label-free quantification with FDR-controlled match-between-runs. *Molecular & Cellular Proteomics*, 20, 100077.

## Acknowledgements

We acknowledge the Minnesota Supercomputing Institute for providing infrastructure and computational resources in the development of the FragPipe Galaxy tool, and the Center for Metabolomics and Proteomics at the University of Minnesota for support generating the data used in our preliminary results. This work was also supported in part by NIH grants UG3CA244687, 1U01CA288888, as well as grant P30CA077598 to the Masonic Cancer Center at the University of Minnesota.