# Tutorial 3: Proteomics Informed by Transcriptomics (PIT)

## 1 Introduction to PIT

Our main motivation for developing GIO was to allow bench scientists to analyse data acquired using the Proteomics Informed by Transcriptomics (PIT) methodology that we published in *Nature Methods*[1]. The innovation of PIT is that the sample under study is analysed using both shotgun proteomic mass spectrometry and RNA-seq transcriptomics so that instead of searching the spectral data against a standard canonical proteome, the RNA-seq data can be used to generate a sample-specific search database.

Without GIO, PIT data analysis would be challenging for most biologists due to the number of proteomics and transcriptomics tools that need to be installed and connected together. To this end, we provide a number of ready-made PIT workflows within GIO (via *Published Workflows* in GIO's *Shared Data* menu). This document explains which workflows are currently available, and what they do. For details of how to access, use and modify these workflows in GIO, please see tutorials 1 and 2.

## 2 PIT workflow without reference genome

One of the main advantages conferred by PIT is that protein identification can be achieved in the absence of a reference proteome, or even in the absence of a reference genome. The workflow that we provide to support this is shown in Figure 1.

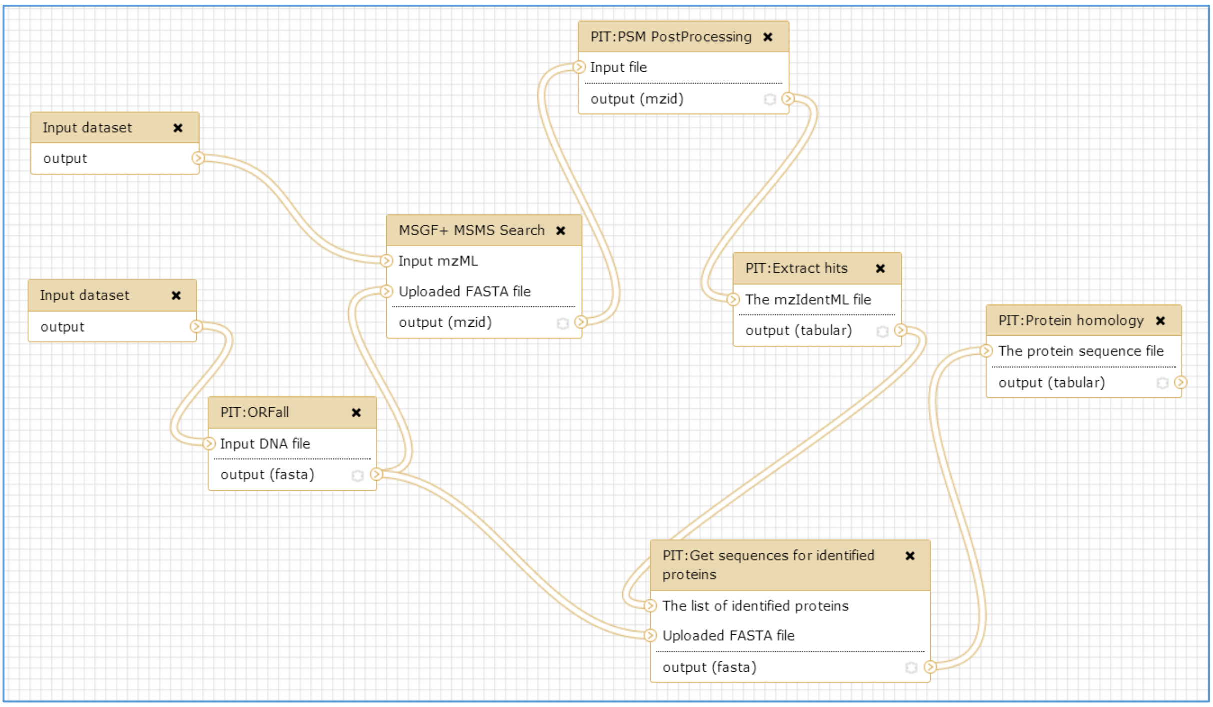

**Figure 1:** *PIT workflow without reference genome.*

There are two inputs to the workflow. One is a file containing spectral data which is expected to be in mzML format, but any common formats (including .raw) can be accommodated by using conversion tools within GIO, such as *MSConvert* from ProteoWizard[2]. The other input is a FASTA file

[1] http://www.ncbi.nlm.nih.gov/pubmed/23142869

[2] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3471674/

containing transcripts assembled *de novo* from RNA-seq data (assembled in a separate workflow, or outside GIO using a tool such as Trinity[3]). The longest open reading frames (ORFs) in all six frames for each of these transcripts are determined using the in-house *ORFall* tool. MSGF+[4] then uses the ORFs to find peptide spectrum matches (PSMs). These PSMs are then post-processed using a GIO tool that combines three tools from mzIdentML-lib[5] to calculate the false discovery rate (FDR) for each PSM, remove peptides below a specified threshold, and infer the identity of proteins from the remaining peptides. This results in a list of confidently identified ORFs in mzIdentML format. Because the ORFs identified from this process are anonymous, further post processing is needed to infer what they may be. To do this, all identified ORFs are BLASTed to find homologous proteins in selected species.

## 3 PIT genome annotation workflow

If a genome exists for the species under study, an alternative workflow can be used that includes additional steps to annotate the genome. In this workflow (Figure 2) transcripts from the input file are mapped to the genome using GMAP[6], in parallel with the peptide and protein identification. At the end of the workflow the in-house *PIT:Integrate* tool is used to integrate the identified transcriptomic and proteomic features into a single GFF3 genome annotation file so that these features can be viewed in their genomic context using a genome browser. This represents a very efficient way of annotating a recently sequenced genome, or enhancing existing genome annotation. As in the previous workflow, a homology search tool can be added at the end of this workflow to provide information as to what the identified proteins may be.

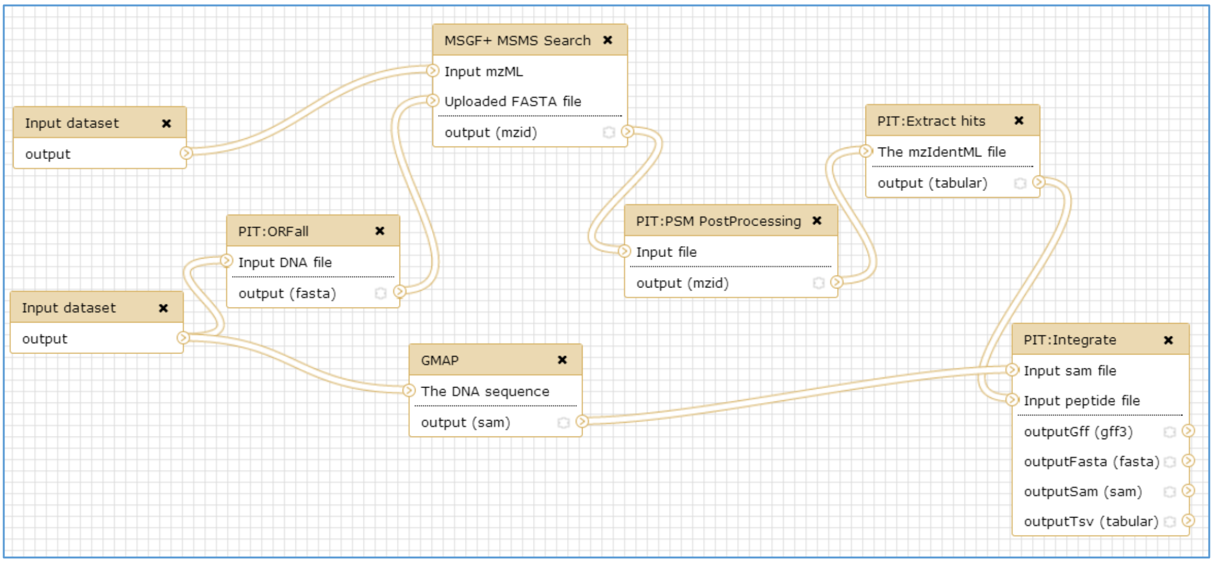

**Figure 2:** *Genome annotating proteomics workflow.*

## 4 PIT workflow with a reference genome

If a well annotated genome is available for the species under study, this genome can be used to maximise the quality of the ORFs that are used to build the database for protein identification. The workflow that we provide for this is shown in Figure 3. This latter part of this workflow is very similar to the genome annotation workflow shown in Figure 2. The difference is that instead of using a list of *de novo* assembled transcripts as our RNA-seq starting point, we begin with the FASTQ formatted

[3] http://www.ncbi.nlm.nih.gov/pubmed/21572440

[4] http://proteomics.ucsd.edu/software-tools/ms-gf/

[5] http://www.ncbi.nlm.nih.gov/pubmed/23813117

[6] http://www.ncbi.nlm.nih.gov/pubmed/15728110

short reads direct from the RNA-seq and assemble these using Tophat[7] followed by Cufflinks[8]. The resulting transcripts are used to generate ORFs that are used to search the spectral data, as in the previously described workflows.

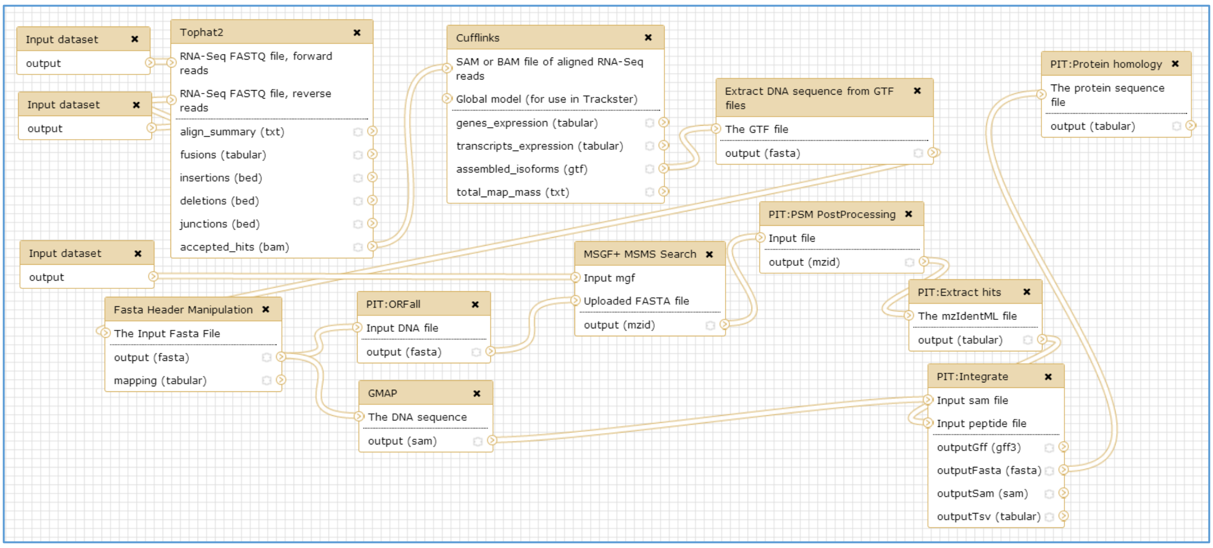

**Figure 3:** *PIT workflow with a well annotated reference genome.*

## 5 Standard identification workflow

For comparison with PIT we include a standard protein identification workflow (Figure 4), to which the inputs are a spectra file and a list of protein sequences to search against. The first step of the workflow is peptide spectrum matching, performed by MSGF+. Post-processing is then performed to refine the list of PSMs and infer proteins from these. The output of these steps is a mzIdentML file, from which the protein identifications are extracted and converted into a tabular (tab separated values - TSV) file for convenient viewing or downloading via the Galaxy interface.

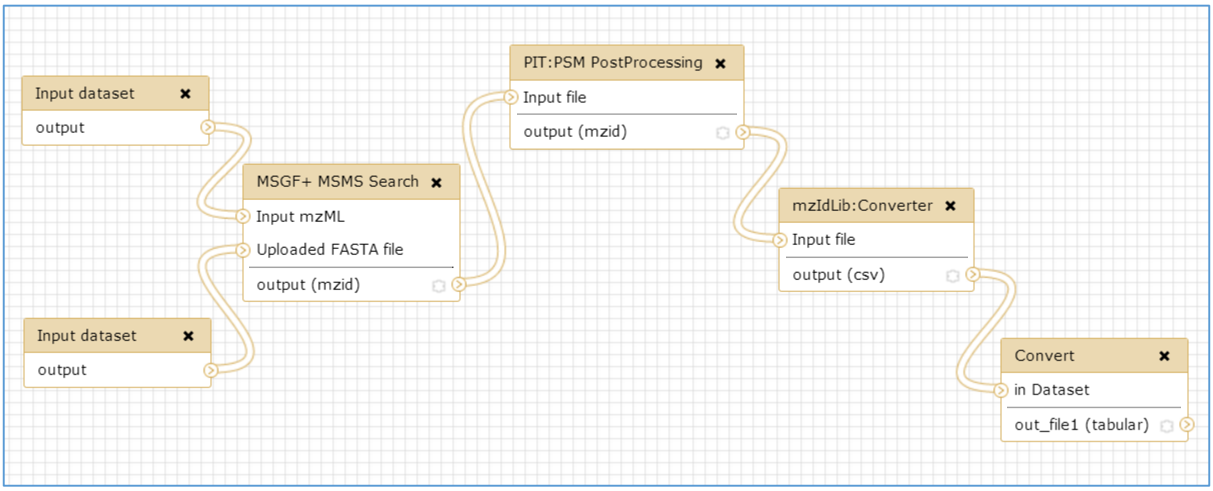

**Figure 4:** *Standard protein identification workflow.*

## 6 Summary and next steps

This tutorial has introduced the PIT workflows that are currently available within GIO. A key benefit of GIO is that these powerful workflows are easy to use, and their general structure and function is

[7] http://www.ncbi.nlm.nih.gov/pubmed/19289445

[8] http://www.ncbi.nlm.nih.gov/pubmed/20436464

easy to understand. However, it should be noted that each workflow is extremely customisable and most of the individual tools within a workflow have a large number of parameters that can be set according to the particular analysis being carried out. Rather than using our workflows as black boxes we strongly recommend that you investigate the functions of each individual tool and its parameters – the best way to do this is via the descriptive text and references that appear below each tool's parameters.

*Last updated 12 August 2014.*