

Galaxy in Plant Pathology: Not everything is NGS data

Peter Cock & Leighton Pritchard
Galaxy Community Conference
Lunteren, The Netherlands
25 May 2011



The James
Hutton
Institute

JHI Plant Pathology

- We work on a range of organisms
 - Plant Viruses
 - Bacteria
 - Oomycetes
 - Fungi
 - Nematodes
 - Aphids (as virus vectors)
- Many genome sequences now available



**JHI Dundee site, formerly SCRI
(Scottish Crop Research Institute)**

Common themes – e.g. Effectors

- I will use “effector” to mean a pathogen produced protein which in some way manipulates the host plant
- The details depend on the type of organism, but we want to identify effector genes, e.g.
 - Similarity to known effectors (e.g. with BLAST)
 - Signal peptides
 - Localization signals
 - Possible horizontal gene transfer (e.g. different GC%)
- Part of larger task of automated gene annotation, e.g.
 - HMMER or RPS-BLAST domain searches

Why Galaxy?

- Hi Peter, could you run a big BLAST job for me?
 - Everyone using standalone BLAST is not practical
 - Want local BLAST web interface with multiple-query support
- Group XXX have just published the YYY genome – could you look for ZZZ proteins please?
 - With a suitable interface, lots of analyses are simple enough for non-bioinformaticians to run and interpret
- You remember that analysis we did last year? I want to do it again on this new genome
 - Running old scripts on new data is tedious
 - Workflows should be reproducible

Why Galaxy?

- Could you run tool XXXX on this data please?
 - Getting the tool:
 - ▶ Many tools are Unix/Linux only (mostly Windows at JHI)
 - Running the tool:
 - ▶ Most tools lack any GUI or web interface
 - Using the results:
 - ▶ Many tools produce their own output formats
- So, we run it via Galaxy instead

Why Galaxy?

■ Plus points for us:

- Don't have to worry about local software installation
- Uniform web based GUI for wrapped tools
- Coupling tools together as sharable repeatable workflows
- Sharing the same data version (better than email/shared drives)
- Open Source (extendable, free)
- Almost any tool can be added

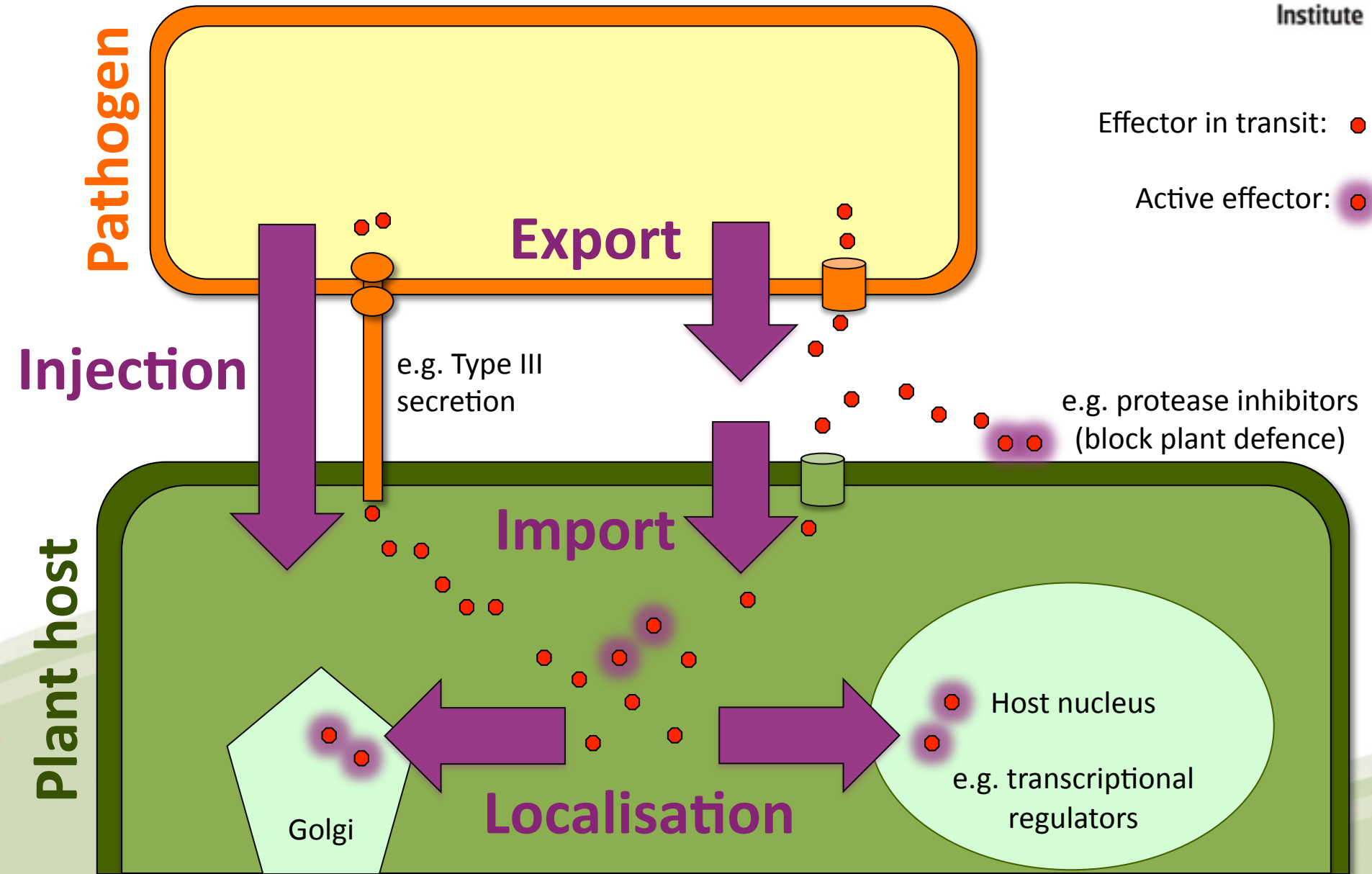
■ Downsides:

- Missing tools (have to invest time wrapping them)
- Bugs in Galaxy (both for end users and tool wrapping)
- Investment in training users

JHI Plant Pathology Server

- Dedicated Linux server (16 core, 32 GB RAM)
- Wiki (general and Lab specific)
- GBrowse (Genome browser)
- Local BLAST databases (wwwblast)
- Galaxy
 - PostgreSQL
 - Python 2.6 (not CentOS provided Python 2.4)
- Compute cluster (not used yet due to firewall issues)

Effector Protein Analysis



Protein Analysis Tools in our Galaxy

All take a FASTA protein file as input, return a tabular file.

- Sequence similarity

- BLAST

- Transmembrane domains

- TMHMM

- Signal Peptides/Motifs

- SignalP
- EffectiveT3
- RXLR

- Nuclear Localisation

- PredictNLS
- NLStradamus

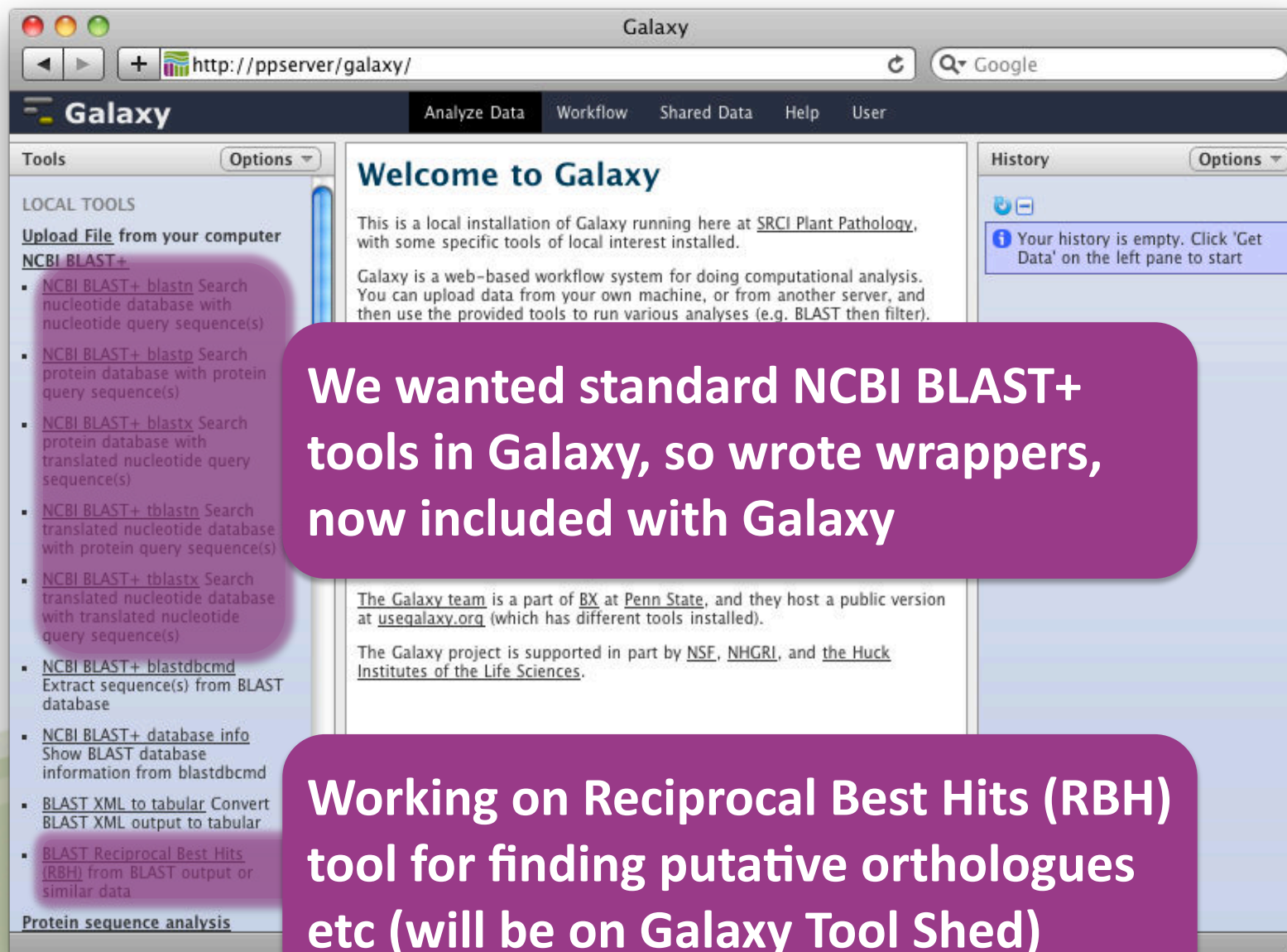
- Nucleolus Localisation

- NoD

- Sub-cellular Localisation

- PSORTB
- WoLF PSORT

NCBI BLAST+



Galaxy

http://ppserver/galaxy/

Galaxy

Analyze Data Workflow Shared Data Help User

Tools Options

LOCAL TOOLS

Upload File from your computer

NCBI BLAST+

- [NCBI BLAST+ blastn](#) Search nucleotide database with nucleotide query sequence(s)
- [NCBI BLAST+ blastp](#) Search protein database with protein query sequence(s)
- [NCBI BLAST+ blastx](#) Search protein database with translated nucleotide query sequence(s)
- [NCBI BLAST+ tblastn](#) Search translated nucleotide database with protein query sequence(s)
- [NCBI BLAST+ tblastx](#) Search translated nucleotide database with translated nucleotide query sequence(s)
- [NCBI BLAST+ blastdbcmd](#) Extract sequence(s) from BLAST database
- [NCBI BLAST+ database info](#) Show BLAST database information from blastdbcmd
- [BLAST XML to tabular](#) Convert BLAST XML output to tabular
- [BLAST Reciprocal Best Hits \(RBH\)](#) from BLAST output or similar data

Protein sequence analysis

Welcome to Galaxy

This is a local installation of Galaxy running here at [SRCI Plant Pathology](#), with some specific tools of local interest installed.

Galaxy is a web-based workflow system for doing computational analysis. You can upload data from your own machine, or from another server, and then use the provided tools to run various analyses (e.g. BLAST then filter).

History Options

Your history is empty. Click 'Get Data' on the left pane to start

The Galaxy team is a part of [BX](#) at [Penn State](#), and they host a public version at [usegalaxy.org](#) (which has different tools installed).

The Galaxy project is supported in part by [NSF](#), [NHGRI](#), and [the Huck Institutes of the Life Sciences](#).

We wanted standard NCBI BLAST+ tools in Galaxy, so wrote wrappers, now included with Galaxy

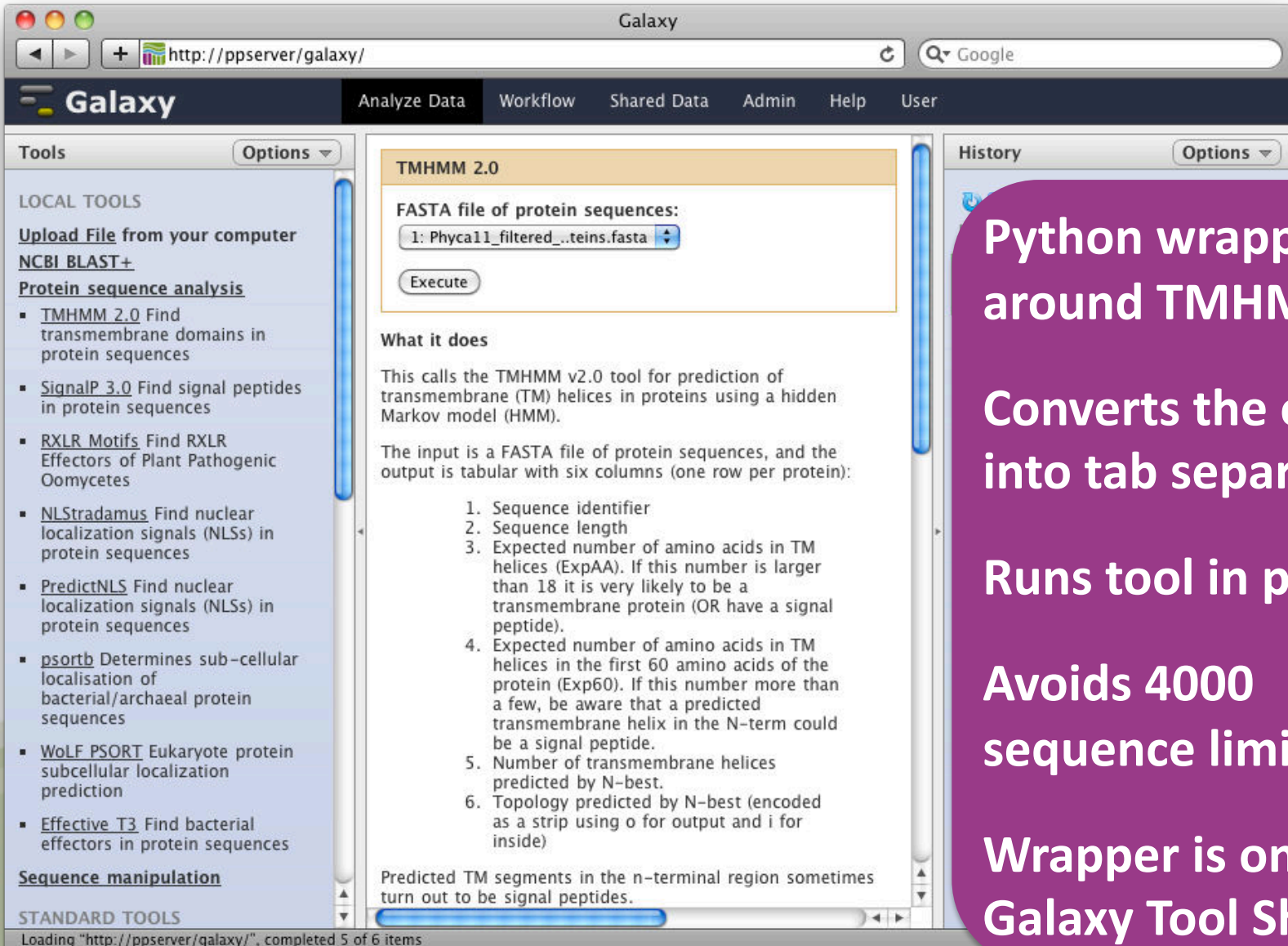
Working on Reciprocal Best Hits (RBH) tool for finding putative orthologues etc (will be on Galaxy Tool Shed)

Protein Sequence analysis tools



The screenshot shows the Galaxy web interface in a browser window. The address bar displays <http://ppserver/galaxy/root>. The main navigation bar includes links for **Analyze Data**, **Workflow**, **Shared Data**, **Help**, and **User**. On the left, the **Tools** panel is expanded, showing categories like **Upload File from your computer**, **NCBI BLAST+**, **Protein sequence analysis**, and **Sequence manipulation**. The **Protein sequence analysis** section lists several tools: **TMHMM 2.0** (Find transmembrane domains), **SignalP 3.0** (Find signal peptides), **RXLR Motifs** (Find RXLR Effectors), **NLStradamus** (Find nuclear localization signals), **PredictNLS** (Find nuclear localization signals), **Nucleolar localization sequence Detector (NoD)** (Find nucleolar localization signals), **psortb** (Determines sub-cellular localisation), **WoLF PSORT** (Eukaryote protein subcellular localization prediction), and **Effective T3** (Find bacterial effectors). The main content area, titled **Welcome to Galaxy**, provides an overview of the system, explaining that it is a local installation at [SRCI Plant Pathology](#). It describes Galaxy as a web-based workflow system for computational analysis, allowing users to upload data and run various analyses like BLAST. It also mentions that analysis pipelines can be saved as workflows and shared. A **Galaxy Screencasts** logo is displayed. The bottom section of the main area mentions that the Galaxy team is part of [BX at Penn State](#) and hosts a public version at usegalaxy.org. It also notes that the project is supported by [NSE](#), [NHGRI](#), and [the Huck Institutes of the Life Sciences](#). On the right, the **History** panel shows a message: "Your history is empty. Click 'Get Data' on the left pane to start".

Transmembrane Domains (TMHMM)



The screenshot shows the Galaxy web interface. The top navigation bar includes 'Galaxy', 'Analyze Data', 'Workflow', 'Shared Data', 'Admin', 'Help', and 'User'. The left sidebar lists various tools under 'LOCAL TOOLS' and 'STANDARD TOOLS'. The main panel displays the 'TMHMM 2.0' tool configuration. It shows a 'FASTA file of protein sequences:' input field with a dropdown menu containing '1: Phyca11_filtered...teins.fasta' and an 'Execute' button. Below the input field, the 'What it does' section explains that the tool calls the TMHMM v2.0 tool for prediction of transmembrane (TM) helices in proteins using a hidden Markov model (HMM). It also states that the input is a FASTA file of protein sequences, and the output is tabular with six columns (one row per protein):

1. Sequence identifier
2. Sequence length
3. Expected number of amino acids in TM helices (ExpAA). If this number is larger than 18 it is very likely to be a transmembrane protein (OR have a signal peptide).
4. Expected number of amino acids in TM helices in the first 60 amino acids of the protein (Exp60). If this number more than a few, be aware that a predicted transmembrane helix in the N-term could be a signal peptide.
5. Number of transmembrane helices predicted by N-best.
6. Topology predicted by N-best (encoded as a strip using o for output and i for inside)

At the bottom of the main panel, it notes: 'Predicted TM segments in the n-terminal region sometimes turn out to be signal peptides.'

The right sidebar shows a 'History' section with an 'Options' dropdown.

Python wrapper
around TMHMM 2.0

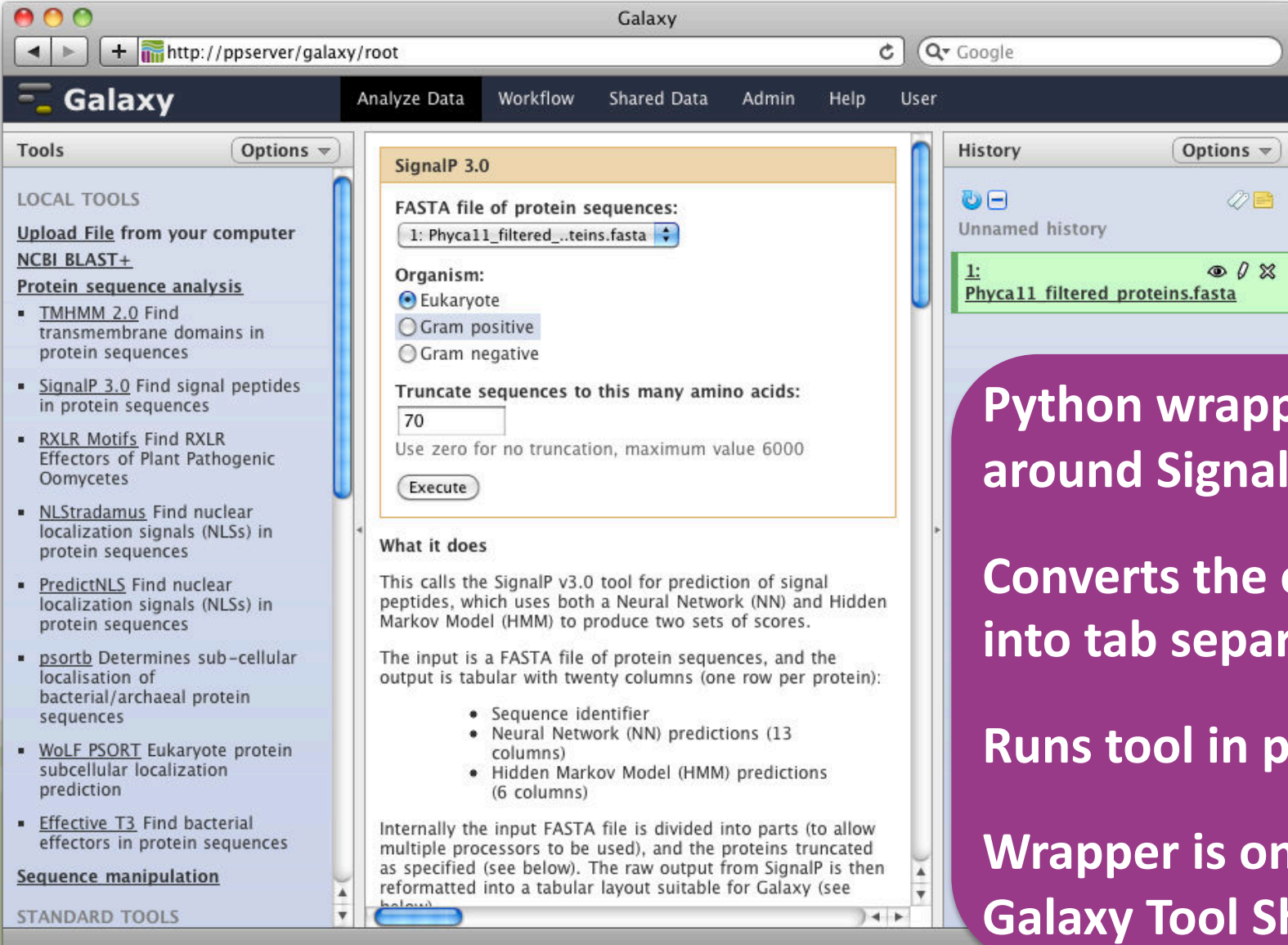
Converts the output
into tab separated

Runs tool in parallel

Avoids 4000
sequence limit

Wrapper is on the
Galaxy Tool Shed

Signal Peptides (SignalP)



The screenshot shows the Galaxy web interface with the SignalP 3.0 tool selected. The interface includes a top navigation bar with 'Galaxy' and links to 'Analyze Data', 'Workflow', 'Shared Data', 'Admin', 'Help', and 'User'. The left sidebar lists various tools under 'LOCAL TOOLS' and 'STANDARD TOOLS'. The main panel displays the 'SignalP 3.0' tool configuration, which includes a 'FASTA file of protein sequences' field with a dropdown menu showing '1: Phyca11_filtered_proteins.fasta'. Below this, the 'Organism' section has radio buttons for 'Eukaryote' (selected), 'Gram positive', and 'Gram negative'. The 'Truncate sequences to this many amino acids' field is set to '70'. An 'Execute' button is at the bottom of the configuration panel. To the right of the configuration panel is a 'History' panel showing a list of jobs, with the current job '1: Phyca11 filtered proteins.fasta' highlighted. The bottom of the interface shows a progress bar and a 'What it does' section describing the tool's function.

SignalP 3.0

FASTA file of protein sequences:
1: Phyca11_filtered_proteins.fasta

Organism:
☒ Eukaryote
☐ Gram positive
☐ Gram negative

Truncate sequences to this many amino acids:
70
Use zero for no truncation, maximum value 6000

Execute

What it does

This calls the SignalP v3.0 tool for prediction of signal peptides, which uses both a Neural Network (NN) and Hidden Markov Model (HMM) to produce two sets of scores.

The input is a FASTA file of protein sequences, and the output is tabular with twenty columns (one row per protein):

- Sequence identifier
- Neural Network (NN) predictions (13 columns)
- Hidden Markov Model (HMM) predictions (6 columns)

Internally the input FASTA file is divided into parts (to allow multiple processors to be used), and the proteins truncated as specified (see below). The raw output from SignalP is then reformatted into a tabular layout suitable for Galaxy (see below).

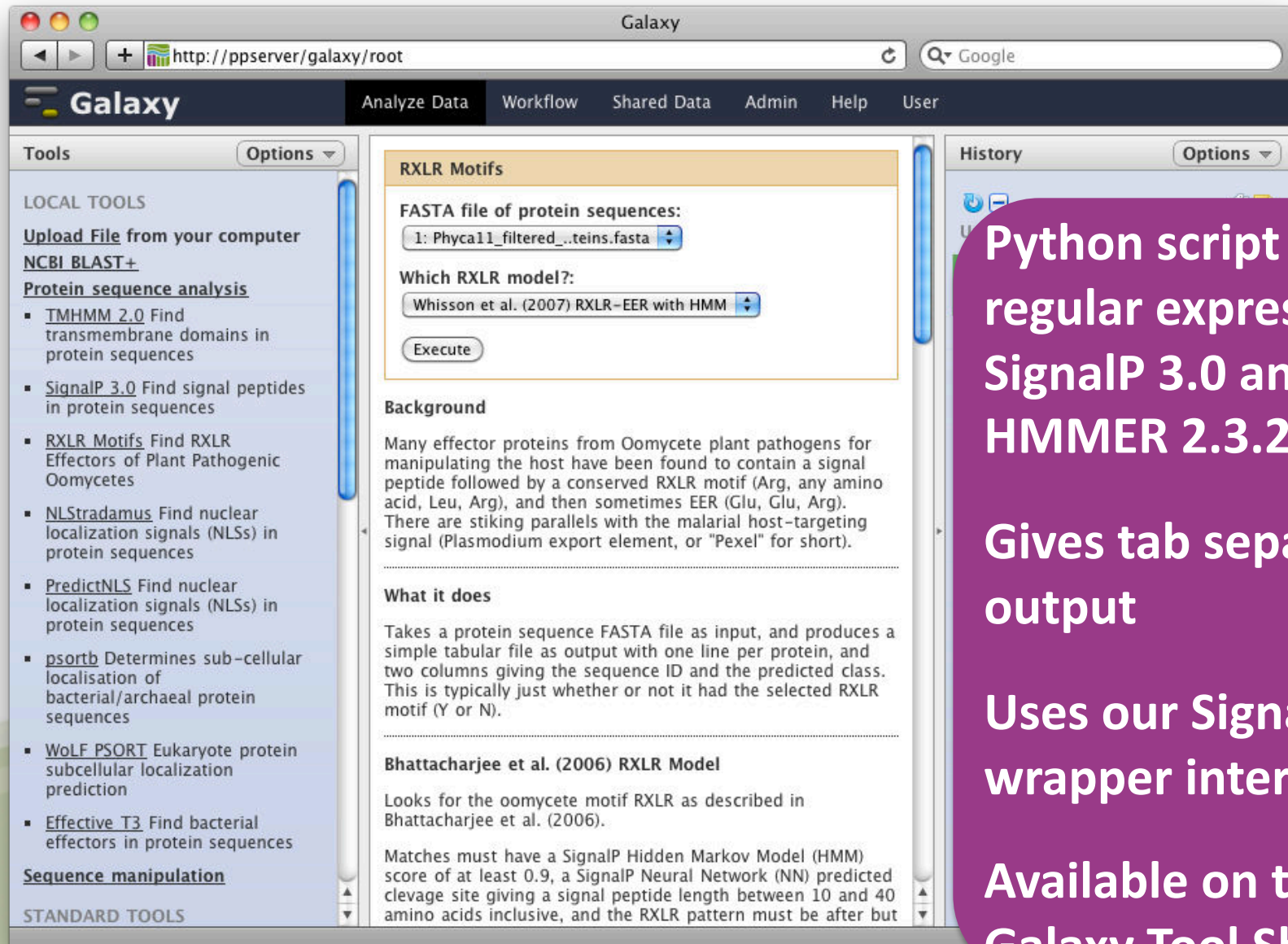
Python wrapper
around SignalP 3.0

Converts the output
into tab separated

Runs tool in parallel

Wrapper is on the
Galaxy Tool Shed

Oomycete RXLR Motifs



The screenshot shows the Galaxy web interface. The top navigation bar includes 'Galaxy', 'Analyze Data', 'Workflow', 'Shared Data', 'Admin', 'Help', and 'User'. The left sidebar lists various tools under 'LOCAL TOOLS' and 'STANDARD TOOLS'. The main panel displays the 'RXLR Motifs' tool configuration. It includes a text input for the 'FASTA file of protein sequences' (set to '1: Phyca11_filtered...teins.fasta'), a dropdown for 'Which RXLR model?' (set to 'Whisson et al. (2007) RXLR-EER with HMM'), and an 'Execute' button. Below the configuration, there is a 'Background' section with text about Oomycete plant pathogens, a 'What it does' section describing the tool's output, and a 'Bhattacharjee et al. (2006) RXLR Model' section with details on the motif and matching criteria.

RXLR Motifs

FASTA file of protein sequences:
1: Phyca11_filtered...teins.fasta

Which RXLR model?:
Whisson et al. (2007) RXLR-EER with HMM

Execute

Background

Many effector proteins from Oomycete plant pathogens for manipulating the host have been found to contain a signal peptide followed by a conserved RXLR motif (Arg, any amino acid, Leu, Arg), and then sometimes EER (Glu, Glu, Arg). There are striking parallels with the malarial host-targeting signal (Plasmodium export element, or "Pexel" for short).

What it does

Takes a protein sequence FASTA file as input, and produces a simple tabular file as output with one line per protein, and two columns giving the sequence ID and the predicted class. This is typically just whether or not it had the selected RXLR motif (Y or N).

Bhattacharjee et al. (2006) RXLR Model

Looks for the oomycete motif RXLR as described in Bhattacharjee et al. (2006).

Matches must have a SignalP Hidden Markov Model (HMM) score of at least 0.9, a SignalP Neural Network (NN) predicted cleavage site giving a signal peptide length between 10 and 40 amino acids inclusive, and the RXLR pattern must be after but

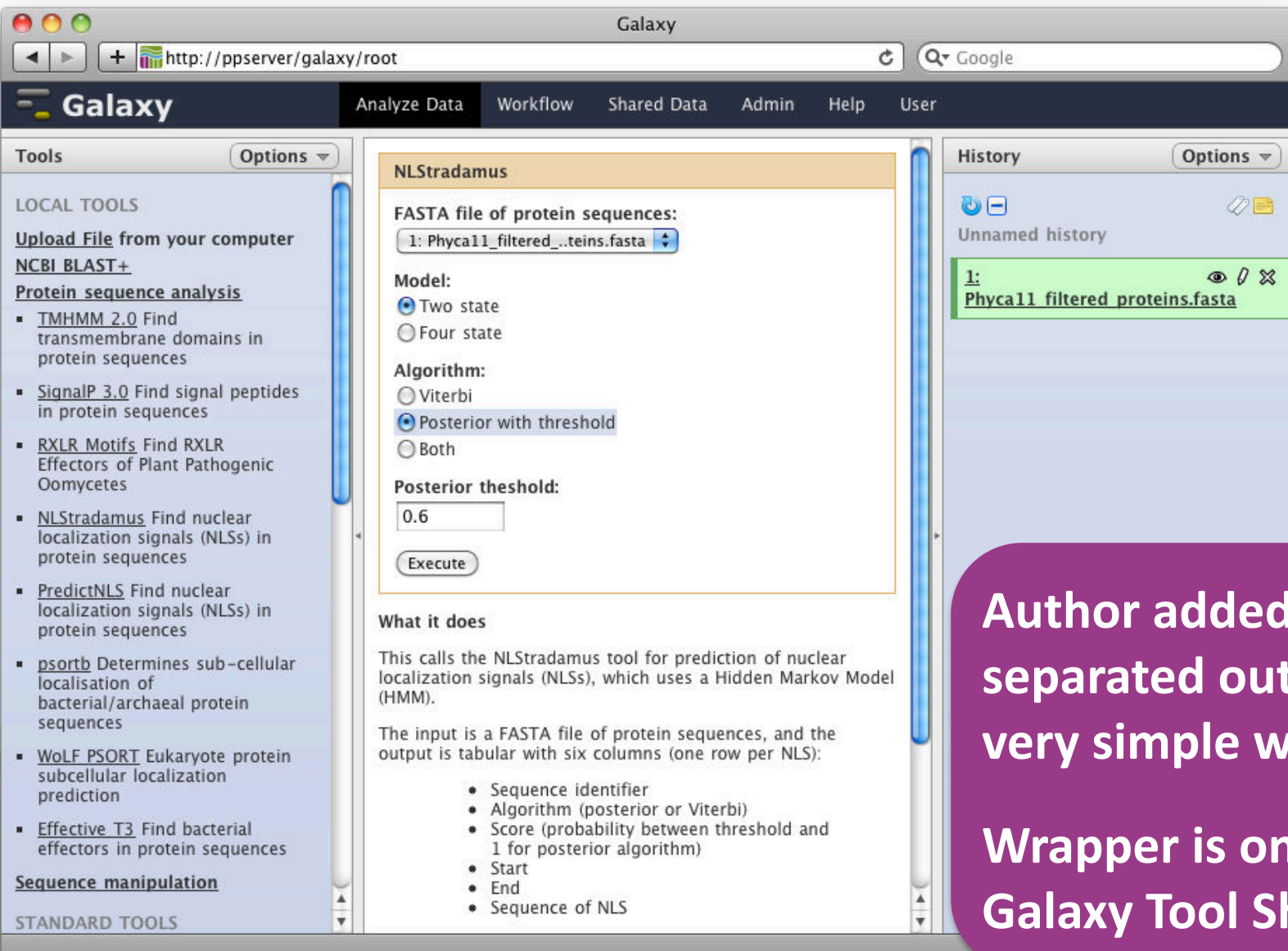
Python script using
regular expressions,
SignalP 3.0 and
HMMER 2.3.2

Gives tab separated
output

Uses our SignalP
wrapper internally

Available on the
Galaxy Tool Shed*

Nuclear Localisation Signals: NLStradamus



Galaxy

http://ppserver/galaxy/root

Google

Galaxy

Analyze Data Workflow Shared Data Admin Help User

Tools Options

LOCAL TOOLS

Upload File from your computer

NCBI BLAST+

Protein sequence analysis

- TMHMM 2.0 Find transmembrane domains in protein sequences
- SignalP 3.0 Find signal peptides in protein sequences
- RXLR Motifs Find RXLR Effectors of Plant Pathogenic Oomycetes
- NLStradamus Find nuclear localization signals (NLSs) in protein sequences
- PredictNLS Find nuclear localization signals (NLSs) in protein sequences
- psortb Determines sub-cellular localisation of bacterial/archaeal protein sequences
- WoLF PSORT Eukaryote protein subcellular localization prediction
- Effective T3 Find bacterial effectors in protein sequences

Sequence manipulation

STANDARD TOOLS

NLStradamus

FASTA file of protein sequences:

1: Phyca11_filtered...teins.fasta

Model:

☒ Two state

☐ Four state

Algorithm:

☐ Viterbi

☒ Posterior with threshold

☐ Both

Posterior threshold:

0.6

Execute

What it does

This calls the NLStradamus tool for prediction of nuclear localization signals (NLSs), which uses a Hidden Markov Model (HMM).

The input is a FASTA file of protein sequences, and the output is tabular with six columns (one row per NLS):

- Sequence identifier
- Algorithm (posterior or Viterbi)
- Score (probability between threshold and 1 for posterior algorithm)
- Start
- End
- Sequence of NLS

History Options

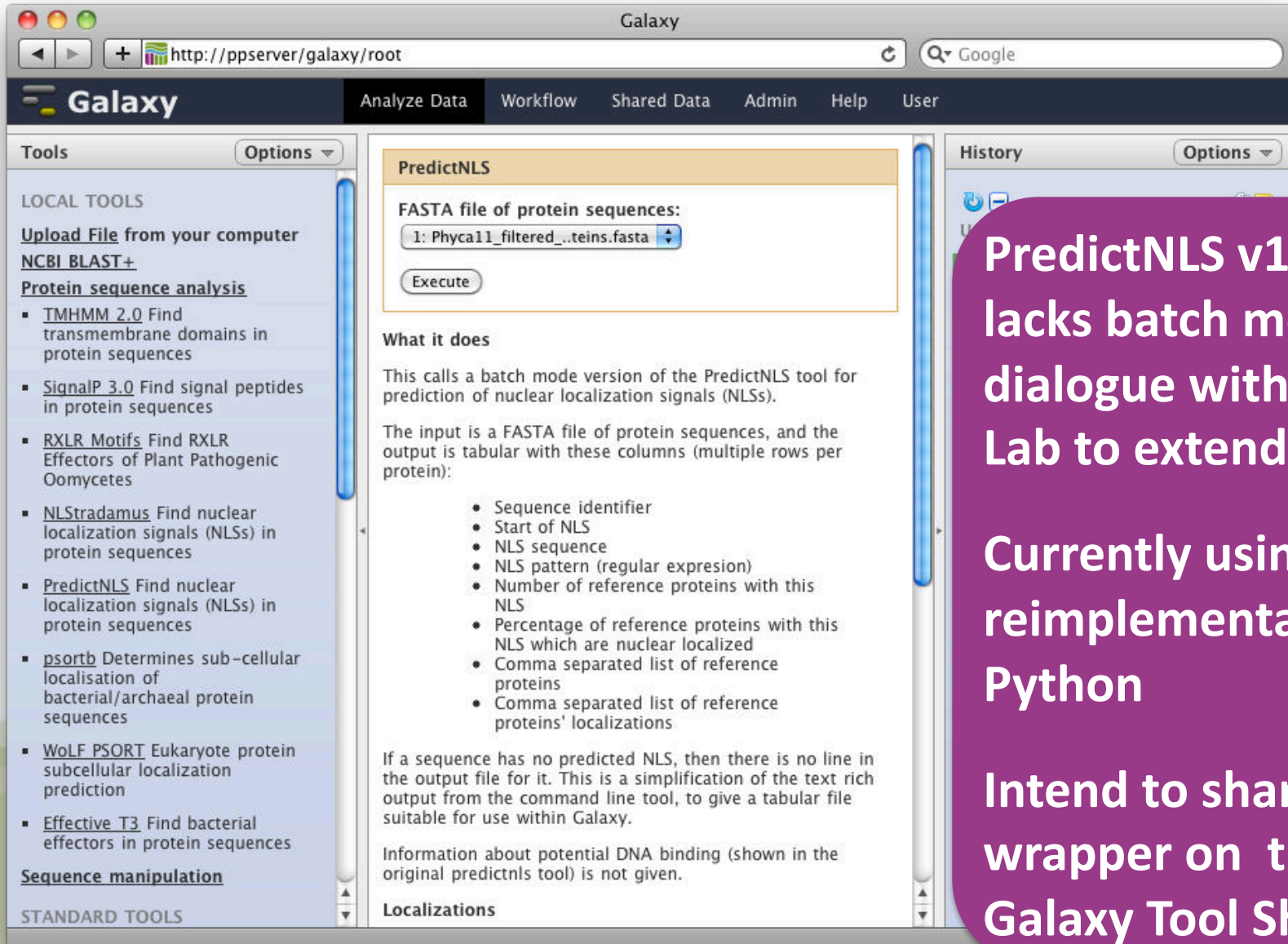
Unnamed history

1: Phyca11 filtered proteins.fasta

Author added tab
separated output, so
very simple wrapper

Wrapper is on the
Galaxy Tool Shed

Nuclear Localisation Signals: PredictNLS



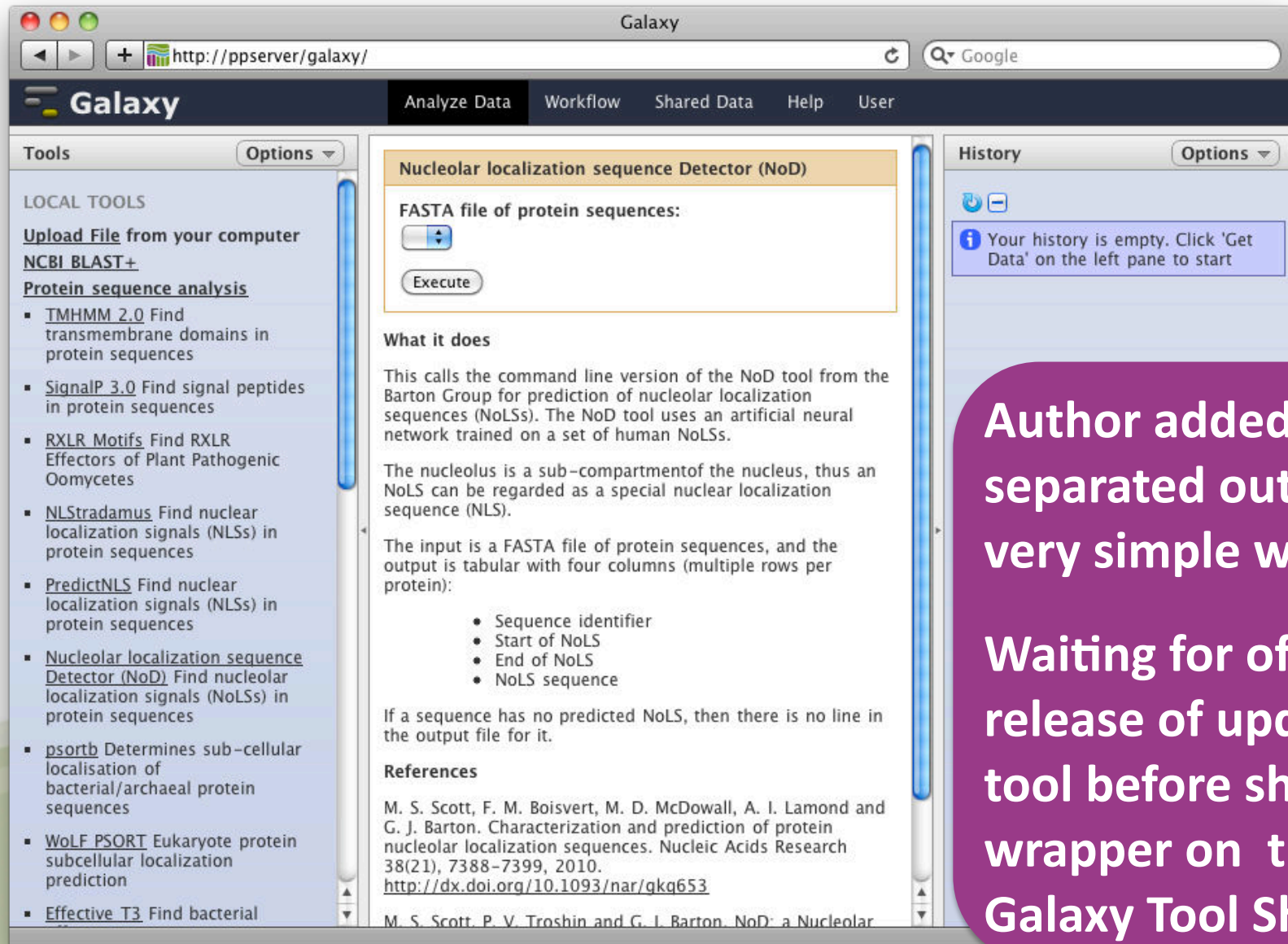
The screenshot shows the Galaxy web interface with the PredictNLS tool selected. The left sidebar lists various tools under 'LOCAL TOOLS', including 'PredictNLS'. The main panel displays the tool's configuration, which includes a text input for a 'FASTA file of protein sequences' (currently set to '1: Phyca11_filtered...teins.fasta') and an 'Execute' button. Below the configuration, the 'What it does' section explains that the tool calls a batch mode version of PredictNLS for predicting nuclear localization signals (NLSs). It also lists the columns in the output file: Sequence identifier, Start of NLS, NLS sequence, NLS pattern (regular expression), Number of reference proteins with this NLS, Percentage of reference proteins with this NLS which are nuclear localized, Comma separated list of reference proteins, and Comma separated list of reference proteins' localizations. A note states that sequences with no predicted NLS will have no line in the output file. The bottom section, 'Localizations', mentions that information about potential DNA binding is not given. The right sidebar shows a 'History' panel.

PredictNLS v1.0.17
lacks batch mode, in
dialogue with Rost
Lab to extend this...

**Currently using own
reimplementation in
Python**

**Intend to share
wrapper on the
Galaxy Tool Shed**

Nucleolar Localisation Signals: NoD

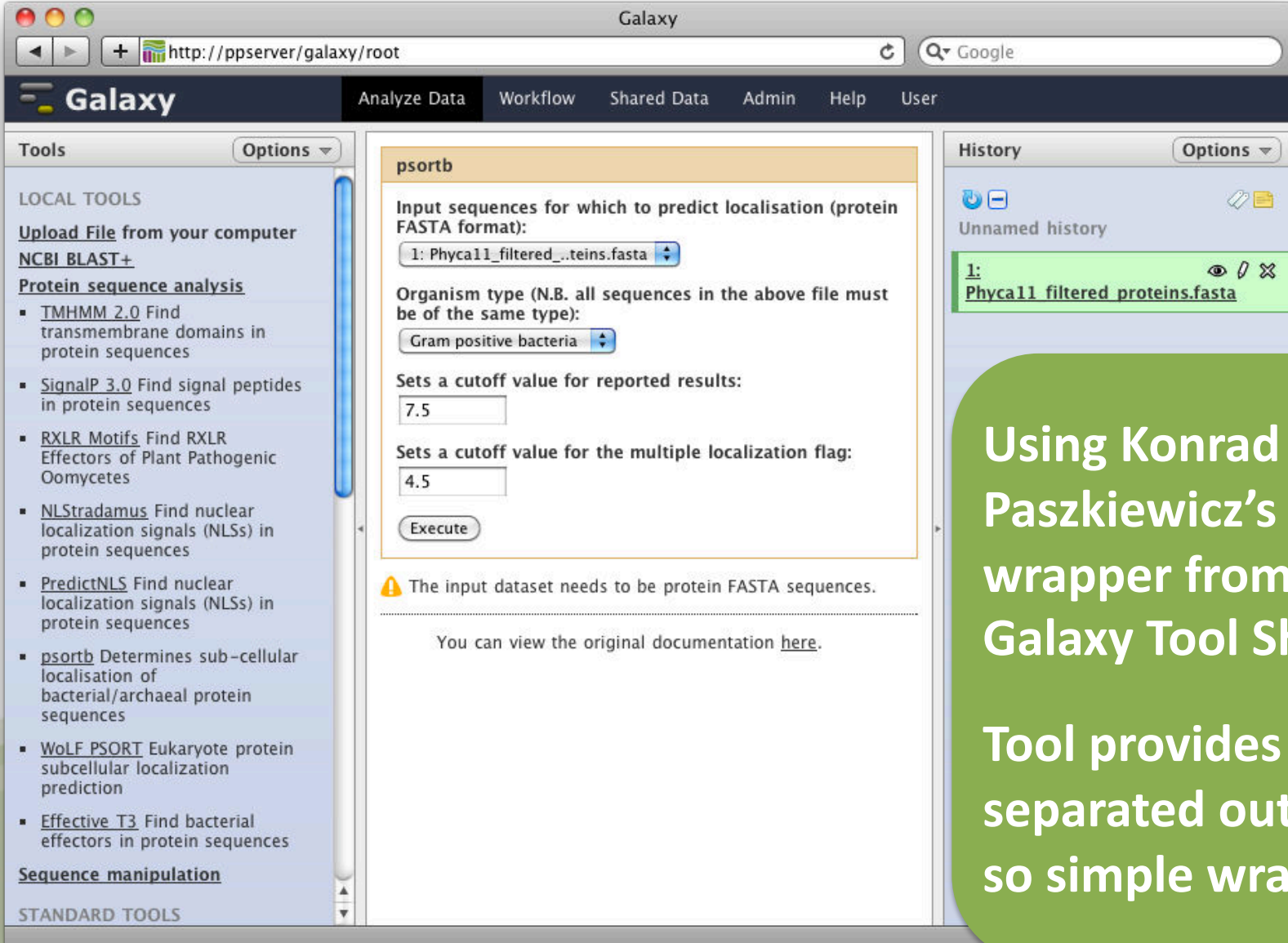


The screenshot shows the Galaxy web interface with the 'Nucleolar localization sequence Detector (NoD)' tool selected. The interface includes a top navigation bar with 'Analyze Data', 'Workflow', 'Shared Data', 'Help', and 'User' tabs. The left sidebar lists various tools under 'LOCAL TOOLS', including 'Upload File from your computer', 'NCBI BLAST+', 'Protein sequence analysis', and 'Nucleolar localization sequence Detector (NoD)'. The main panel displays the tool's description, which states that it calls the command line version of the NoD tool from the Barton Group for prediction of nucleolar localization sequences (NoLSs). It explains that the nucleolus is a sub-compartment of the nucleus and that an NoLS can be regarded as a special nuclear localization sequence (NLS). The input is a FASTA file of protein sequences, and the output is tabular with four columns: Sequence identifier, Start of NoLS, End of NoLS, and NoLS sequence. A note indicates that if a sequence has no predicted NoLS, there is no line in the output file for it. The 'References' section lists two publications: Scott et al. (2010) and Scott et al. (submitted). The right sidebar shows the 'History' tab, which is currently empty.

Author added tab separated output, so very simple wrapper

Waiting for official release of updated tool before sharing wrapper on the Galaxy Tool Shed

Prokaryotic Sub-cellular Localisation: PSORTb

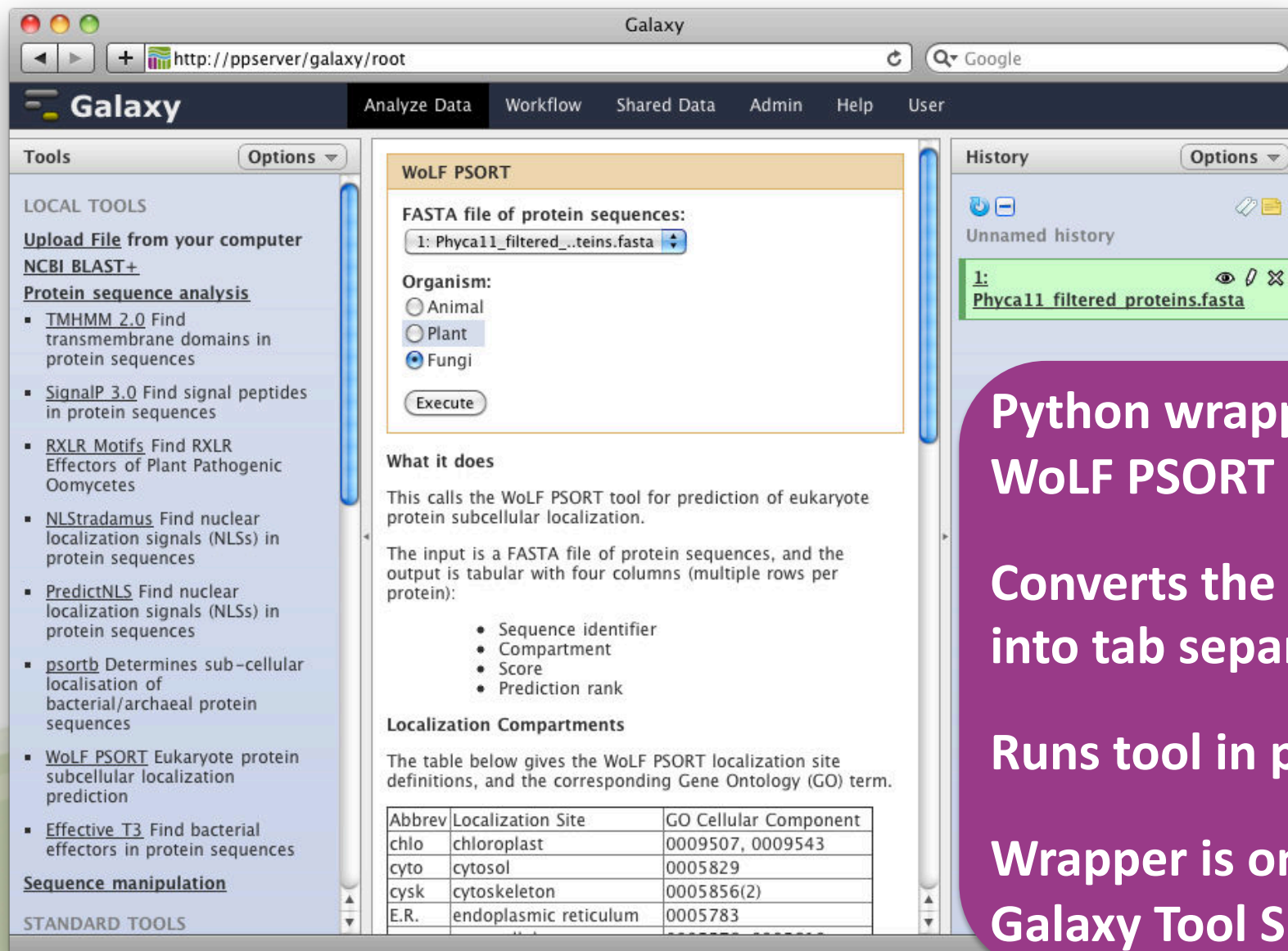


The screenshot shows the Galaxy web interface. The top navigation bar includes 'Galaxy', 'Analyze Data', 'Workflow', 'Shared Data', 'Admin', 'Help', and 'User'. The left sidebar lists various tools under 'LOCAL TOOLS', including 'Upload File from your computer', 'NCBI BLAST+', 'Protein sequence analysis' (with sub-items like TMHMM 2.0, SignalP 3.0, RXLR Motifs, NLStradamus, PredictNLS, psortb, WoLF PSORT, and Effective T3), and 'Sequence manipulation'. The main panel displays the 'psortb' tool configuration. It has a title bar 'psortb' and a description: 'Input sequences for which to predict localisation (protein FASTA format):'. The input field shows '1: Phyca11_filtered_proteins.fasta'. Below this, it asks for 'Organism type (N.B. all sequences in the above file must be of the same type):' with a dropdown set to 'Gram positive bacteria'. There are two input fields for cutoff values: 'Sets a cutoff value for reported results:' (7.5) and 'Sets a cutoff value for the multiple localization flag:' (4.5). An 'Execute' button is at the bottom. A warning message states: 'The input dataset needs to be protein FASTA sequences. You can view the original documentation [here](#).' The right sidebar shows a 'History' panel with 'Unnamed history' and a single entry '1: Phyca11 filtered proteins.fasta'.

Using Konrad Paszkiewicz's wrapper from Galaxy Tool Shed

Tool provides tab separated output, so simple wrapper

Eukaryotic Sub-cellular Localisation: WoLF PSORT



The screenshot shows the Galaxy web interface with the WoLF PSORT tool selected. The tool configuration includes a FASTA file upload, organism selection (Fungi is chosen), and an 'Execute' button. The 'What it does' section explains that the tool predicts eukaryote protein subcellular localization. The 'Localization Compartments' section provides a table of definitions.

WoLF PSORT

FASTA file of protein sequences:
1: Phyca11_filtered_proteins.fasta

Organism:
☐ Animal
☐ Plant
☒ Fungi

Execute

What it does

This calls the WoLF PSORT tool for prediction of eukaryote protein subcellular localization.

The input is a FASTA file of protein sequences, and the output is tabular with four columns (multiple rows per protein):

- Sequence identifier
- Compartment
- Score
- Prediction rank

Localization Compartments

The table below gives the WoLF PSORT localization site definitions, and the corresponding Gene Ontology (GO) term.

Abbrev	Localization Site	GO Cellular Component
chlo	chloroplast	0009507, 0009543
cyto	cytosol	0005829
cysk	cytoskeleton	0005856(2)
E.R.	endoplasmic reticulum	0005783

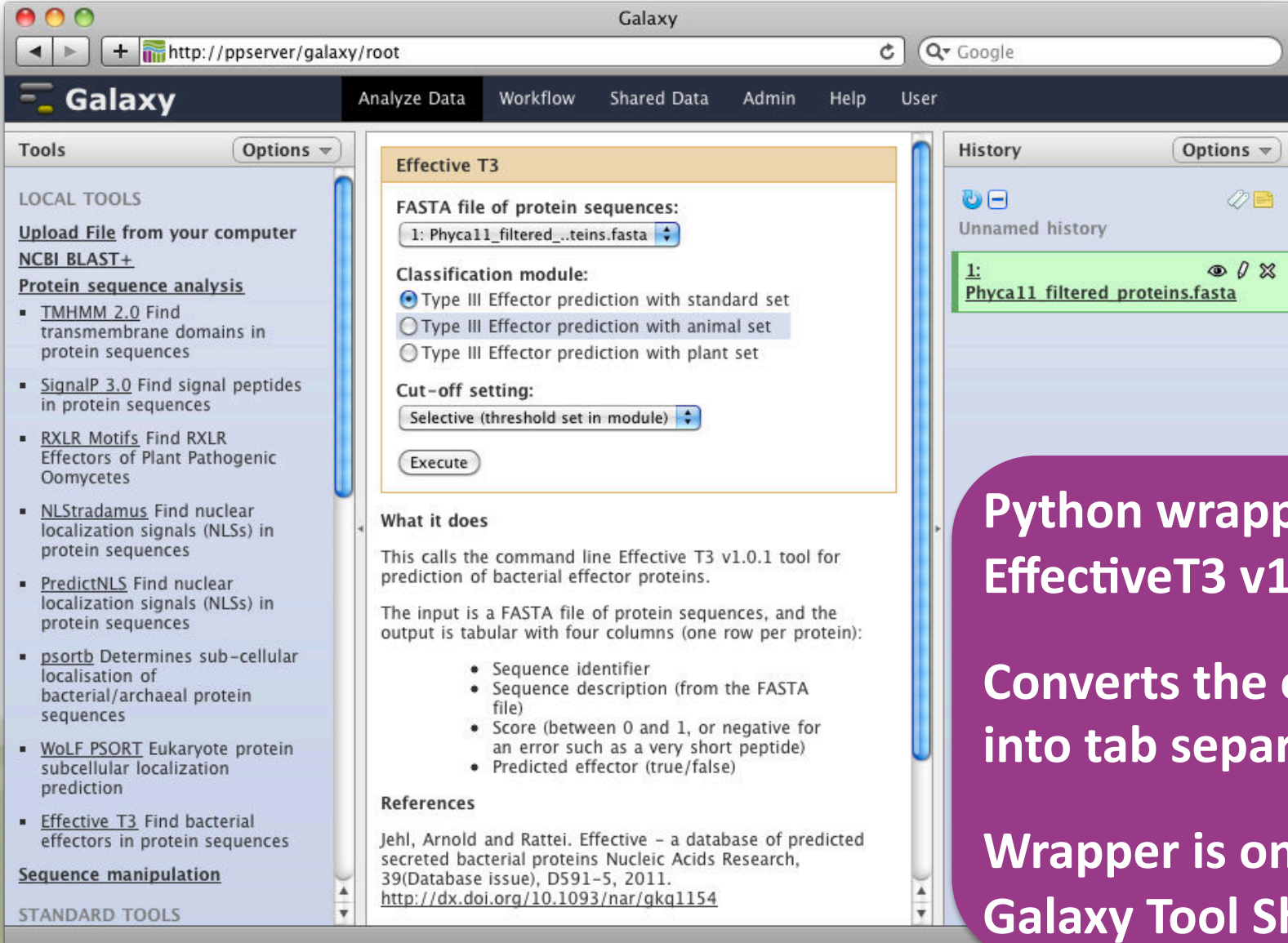
Python wrapper for
WoLF PSORT 2.0

Converts the output
into tab separated

Runs tool in parallel

Wrapper is on the
Galaxy Tool Shed

Type III Secretion Signals: EffectiveT3



The screenshot shows the Galaxy web interface with the Effective T3 tool selected. The interface includes a top navigation bar with links for Analyze Data, Workflow, Shared Data, Admin, Help, and User. The left sidebar lists various tools under 'LOCAL TOOLS' and 'STANDARD TOOLS'. The main panel displays the Effective T3 tool configuration, which includes a FASTA file input, classification module options, and a cut-off setting. The right sidebar shows the history of the tool execution.

Effective T3

FASTA file of protein sequences:
1: Phyca11_filtered_proteins.fasta

Classification module:
☒ Type III Effector prediction with standard set
☐ Type III Effector prediction with animal set
☐ Type III Effector prediction with plant set

Cut-off setting:
Selective (threshold set in module)

Execute

What it does

This calls the command line Effective T3 v1.0.1 tool for prediction of bacterial effector proteins.

The input is a FASTA file of protein sequences, and the output is tabular with four columns (one row per protein):

- Sequence identifier
- Sequence description (from the FASTA file)
- Score (between 0 and 1, or negative for an error such as a very short peptide)
- Predicted effector (true/false)

References

Jehl, Arnold and Rattei. Effective – a database of predicted secreted bacterial proteins Nucleic Acids Research, 39(Database issue), D591–5, 2011.
<http://dx.doi.org/10.1093/nar/gkq1154>

History

Unnamed history

1: Phyca11 filtered proteins.fasta

Python wrapper for
EffectiveT3 v1.0.1

Converts the output
into tab separated

Wrapper is on the
Galaxy Tool Shed*

Observations from Wrapping Tools

- Tabular output for Galaxy
 - Most tools' output needed reformatting
- Some tools are not threaded
 - I use a Python wrapper script to divide the input (using subprocess rather than multiprocessing module for Python 2.4 compatibility)
- Interaction with tool authors can be productive and informative, and improve their tools
- To tool authors
 - Offer tabular output (if appropriate)
 - Better error handling (e.g. zero length sequences)

Workflow example - RXLR motifs

- Important translocation motif in oomycetes
- We have implemented three methods in Galaxy:
 - Bhattacharjee et al. (2006)
 - Win et al. (2007)
 - Whisson et al. (2007)
- Venn Diagram comparing the three methods

Tools

Options ▾

LOCAL TOOLS

[Upload File from your computer](#)[NCBI BLAST+](#)[Protein sequence analysis](#)

- [TMHMM 2.0](#) Find transmembrane domains in protein sequences
- [SignalP 3.0](#) Find signal peptides in protein sequences
- [RXLR Motifs](#) Find RXLR Effectors of Plant Pathogenic Oomycetes
- [NLStradamus](#) Find nuclear localization signals (NLSs) in protein sequences
- [PredictNLS](#) Find nuclear localization signals (NLSs) in protein sequences
- [psortb](#) Determines sub-cellular localisation of bacterial/archaeal protein sequences
- [WoLF PSORT](#) Eukaryote protein subcellular localization prediction
- [Effective T3](#) Find bacterial effectors in protein sequences

[Sequence manipulation](#)

STANDARD TOOLS

RXLR Motifs

FASTA file of protein sequences:

1: Phyca11_filtered_...teins.fasta ▾

Which RXLR model?:

Whisson et al. (2007) RXLR-EER with HMM ▾

Execute

Background

Many effector proteins from Oomycete plant pathogens for manipulating the host have been found to contain a signal peptide followed by a conserved RXLR motif (Arg, any amino acid, Leu, Arg), and then sometimes EER (Glu, Glu, Arg). There are striking parallels with the malarial host-targeting signal (Plasmodium export element, or "Pexel" for short).

What it does

Takes a protein sequence FASTA file as input, and produces a simple tabular file as output with one line per protein, and two columns giving the sequence ID and the predicted class. This is typically just whether or not it had the selected RXLR motif (Y or N).

Bhattacharjee et al. (2006) RXLR Model

Looks for the oomycete motif RXLR as described in Bhattacharjee et al. (2006).

Matches must have a SignalP Hidden Markov Model (HMM) score of at least 0.9, a SignalP Neural Network (NN) predicted cleavage site giving a signal peptide length between 10 and 40 amino acids inclusive, and the RXLR pattern must be after but

History

Options ▾



Unnamed history

1: [Phyca11 filtered proteins.fasta](#)

19,805 sequences
format: fasta, database: 2
Info: uploaded fasta file

```
>jgi|Phyca11|532085|estExt2_fgenes1_
MGNVYSTSSSSDTQQVERPEEKLHSLSDTTVTSSQ
GIRYTDEQTKRKQGGNSPFLVSGVLTWIKACAGGSS
LNDVVL
>jgi|Phyca11|80978|gw1.4.1034.1
DVFLDIGSGVGNVVAQFALSTKVRACIGIEIRRVLAD
```


Tools

Options ▾

LOCAL TOOLS

[Upload File from your computer](#)[NCBI BLAST+](#)[Protein sequence analysis](#)

- [TMHMM 2.0](#) Find transmembrane domains in protein sequences
- [SignalP 3.0](#) Find signal peptides in protein sequences
- [RXLR Motifs](#) Find RXLR Effectors of Plant Pathogenic Oomycetes
- [NLStradamus](#) Find nuclear localization signals (NLSs) in protein sequences
- [PredictNLS](#) Find nuclear localization signals (NLSs) in protein sequences
- [psortb](#) Determines sub-cellular localisation of bacterial/archaeal protein sequences
- [WoLF PSORT](#) Eukaryote protein subcellular localization prediction
- [Effective T3](#) Find bacterial effectors in protein sequences

[Sequence manipulation](#)

STANDARD TOOLS



This dataset is large and only the first megabyte is shown below.

[Show all](#) | [Save](#)

```
#ID Whisson2007
jgi|Phyca11|532085|estExt2_fgenes1_pg.C.PHYCAscaffold_40001
jgi|Phyca11|80978|gw1.4.1034.1 neither
jgi|Phyca11|99641|e_gw1.4.982.1 neither
jgi|Phyca11|13479|fgenes1_pg.PHYCAscaffold_4_#_2 neither
jgi|Phyca11|99637|e_gw1.4.173.1 neither
jgi|Phyca11|503463|fgenes2_kg.PHYCAscaffold_4_#_2_#_Contig4450
jgi|Phyca11|99784|e_gw1.4.911.1 neither
jgi|Phyca11|525476|estExt2_fgenes1_pm.C.PHYCAscaffold_40002
jgi|Phyca11|100524|e_gw1.4.603.1 neither
jgi|Phyca11|539761|estExt2_Genewise1Plus.C.PHYCAscaffold_40015
jgi|Phyca11|503470|fgenes2_kg.PHYCAscaffold_4_#_9_#_4100341:2
jgi|Phyca11|503471|fgenes2_kg.PHYCAscaffold_4_#_10_#_gi|189084
jgi|Phyca11|503473|fgenes2_kg.PHYCAscaffold_4_#_12_#_4098755:1
jgi|Phyca11|539767|estExt2_Genewise1Plus.C.PHYCAscaffold_40021
jgi|Phyca11|559729|estExt2_Genewise1.C.PHYCAscaffold_40022
jgi|Phyca11|4920|fgenes1_pm.PHYCAscaffold_4_#_8 neither
jgi|Phyca11|13487|fgenes1_pg.PHYCAscaffold_4_#_10 neither
jgi|Phyca11|99638|e_gw1.4.611.1 neither
jgi|Phyca11|13489|fgenes1_pg.PHYCAscaffold_4_#_12 neither
jgi|Phyca11|503477|fgenes2_kg.PHYCAscaffold_4_#_16_#_4099940:1
jgi|Phyca11|4923|fgenes1_pm.PHYCAscaffold_4_#_11 neither
jgi|Phyca11|503479|fgenes2_kg.PHYCAscaffold_4_#_18_#_gi|189083
jgi|Phyca11|539773|estExt2_Genewise1Plus.C.PHYCAscaffold_40033
jgi|Phyca11|503483|fgenes2_kg.PHYCAscaffold_4_#_22_#_Contig874
jgi|Phyca11|539778|estExt2_Genewise1Plus.C.PHYCAscaffold_40038
jgi|Phyca11|503485|fgenes2_kg.PHYCAscaffold_4_#_24_#_Contig592
jgi|Phyca11|559743|estExt2_Genewise1.C.PHYCAscaffold_40042
jgi|Phyca11|539783|estExt2_Genewise1Plus.C.PHYCAscaffold_40043
jgi|Phyca11|525486|estExt2_fgenes1_pm.C.PHYCAscaffold_40015
jgi|Phyca11|525487|estExt2_fgenes1_pm.C.PHYCAscaffold_40016
jgi|Phyca11|99772|e_gw1.4.1100.1 neither
jgi|Phyca11|99723|e_gw1.4.865.1 neither
jgi|Phyca11|559752|estExt2_Genewise1.C.PHYCAscaffold_40052
jgi|Phyca11|13498|fgenes1_pg.PHYCAscaffold_4_#_21 neither
jgi|Phyca11|525489|estExt2_fgenes1_pm.C.PHYCAscaffold_40018
jgi|Phyca11|559759|estExt2_Genewise1.C.PHYCAscaffold_40059
jgi|Phyca11|539800|estExt2_Genewise1Plus.C.PHYCAscaffold_40061
jgi|Phyca11|99569|e_gw1.4.1190.1 neither
jgi|Phyca11|13501|fgenes1_pg.PHYCAscaffold_4_#_24 neither
jgi|Phyca11|503494|fgenes2_kg.PHYCAscaffold_4_#_33_#_gi|189084
jgi|Phyca11|503496|fgenes2_kg.PHYCAscaffold_4_#_35_#_Contig282
jgi|Phyca11|100304|e_gw1.4.1152.1 neither
jgi|Phyca11|503499|fgenes2_kg.PHYCAscaffold_4_#_38_#_Contig292
jgi|Phyca11|525496|estExt2_fgenes1_pm.C.PHYCAscaffold_40025
jgi|Phyca11|503501|fgenes2_kg.PHYCAscaffold_4_#_40_#_Contig435
jgi|Phyca11|299261|gw1.4.967.1 neither
```

History

Options ▾

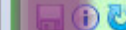


Unnamed history

2: Whisson et al. (2007)

RXLR-EER with HMM

19,805 lines, 1 comments
format: tabular, database: ?
Info: Whisson2007 for 19805 sequences:
Y = 89, hmm = 43, neither = 19657, re = 16



1

```
#ID
jgi|Phyca11|532085|estExt2_fgenes1_pg.C.PHYCAscaffold_40001
jgi|Phyca11|80978|gw1.4.1034.1
jgi|Phyca11|99641|e_gw1.4.982.1
jgi|Phyca11|13479|fgenes1_pg.PHYCAscaffold_4_#_2
jgi|Phyca11|99637|e_gw1.4.173.1
```

1:

Phyca11 filtered proteins.fasta

19,805 sequences
format: fasta, database: ?
Info: uploaded fasta file



```
>jgi|Phyca11|532085|estExt2_fgenes1_pg.C.PHYCAscaffold_40001
MGNVYSTDSSSDTQQQVERPEEKLHSLSDTTVT:
GIRYTDQETKRRQGGNSPFLVSGVLTWIKACAG
```


Galaxy2-[Whisson_et_al._(2007)_RXLR-EER_with_HMM].tabular.txt						
New Open Save Print Import Copy Paste Format Undo Redo AutoSum Sort A-Z Sort Z-A Gallery Toolbox Zoom Help						
Sheets Charts SmartArt Graphics WordArt						
	A	B	C	D	E	F
1	#ID	Whisson2007				
2	jgi Phyca11 532085 estExt2_fgenesh1_pg.C_PHYCAscaffold_40001	neither				
3	jgi Phyca11 80978 gw1.4.1034.1	neither				
4	jgi Phyca11 99641 e_gw1.4.982.1	neither				
5	jgi Phyca11 13479 fgenesh1_pg.PHYCAscaffold_4_#_2	neither				
6	jgi Phyca11 99637 e_gw1.4.173.1	neither				
7	jgi Phyca11 503463 fgenesh2_kg.PHYCAscaffold_4_#_2_#_Contig4450.1	neither				
8	jgi Phyca11 99784 e_gw1.4.911.1	neither				
9	jgi Phyca11 525476 estExt2_fgenesh1_pm.C_PHYCAscaffold_40002	neither				
10	jgi Phyca11 100524 e_gw1.4.603.1	neither				
11	jgi Phyca11 539761 estExt2_Genewise1Plus.C_PHYCAscaffold_40015	neither				
12	jgi Phyca11 503470 fgenesh2_kg.PHYCAscaffold_4_#_9_#_4100341:2	neither				
13	jgi Phyca11 503471 fgenesh2_kg.PHYCAscaffold_4_#_10_#_gi 189084718 gb BT032236.1	neither				
14	jgi Phyca11 503473 fgenesh2_kg.PHYCAscaffold_4_#_12_#_4098755:1	neither				
15	jgi Phyca11 539767 estExt2_Genewise1Plus.C_PHYCAscaffold_40021	neither				
16	jgi Phyca11 559729 estExt2_Genewise1.C_PHYCAscaffold_40022	neither				
17	jgi Phyca11 4920 fgenesh1_pm.PHYCAscaffold_4_#_8	neither				
18	jgi Phyca11 13487 fgenesh1_pg.PHYCAscaffold_4_#_10	neither				
19	jgi Phyca11 99638 e_gw1.4.611.1	neither				
20	jgi Phyca11 13489 fgenesh1_pg.PHYCAscaffold_4_#_12	neither				
21	jgi Phyca11 503477 fgenesh2_kg.PHYCAscaffold_4_#_16_#_4099940:1	neither				
22	jgi Phyca11 4923 fgenesh1_pm.PHYCAscaffold_4_#_11	neither				
23	jgi Phyca11 503479 fgenesh2_kg.PHYCAscaffold_4_#_18_#_gi 189083978 gb BT031494.1	neither				
24	jgi Phyca11 539773 estExt2_Genewise1Plus.C_PHYCAscaffold_40033	neither				
25	jgi Phyca11 503483 fgenesh2_kg.PHYCAscaffold_4_#_22_#_Contig874.1	neither				
26	jgi Phyca11 539778 estExt2_Genewise1Plus.C_PHYCAscaffold_40038	neither				
27	jgi Phyca11 503485 fgenesh2_kg.PHYCAscaffold_4_#_24_#_Contig5923.1	neither				
28	jgi Phyca11 559743 estExt2_Genewise1.C_PHYCAscaffold_40042	neither				
29	jgi Phyca11 539783 estExt2_Genewise1Plus.C_PHYCAscaffold_40043	neither				
30	jgi Phyca11 525486 estExt2_fgenesh1_pm.C_PHYCAscaffold_40015	neither				
31	jgi Phyca11 525487 estExt2_fgenesh1_pm.C_PHYCAscaffold_40016	neither				
32	jgi Phyca11 99772 e_gw1.4.1100.1	neither				
33	jgi Phyca11 99723 e_gw1.4.865.1	neither				
34	jgi Phyca11 559752 estExt2_Genewise1.C_PHYCAscaffold_40052	neither				
35	jgi Phyca11 13498 fgenesh1_pg.PHYCAscaffold_4_#_21	neither				
36	jgi Phyca11 525489 estExt2_fgenesh1_pm.C_PHYCAscaffold_40018	neither				
37	jgi Phyca11 559759 estExt2_Genewise1.C_PHYCAscaffold_40059	neither				
38	jgi Phyca11 539800 estExt2_Genewise1Plus.C_PHYCAscaffold_40061	neither				
39	jgi Phyca11 99569 e_gw1.4.1190.1	neither				
40	jgi Phyca11 13501 fgenesh1_pg.PHYCAscaffold_4_#_24	neither				
41	jgi Phyca11 503494 fgenesh2_kg.PHYCAscaffold_4_#_33_#_gi 189084545 gb BT032061.1	neither				

Tools

Options ▾

LOCAL TOOLS

[Upload File from your computer](#)[NCBI BLAST+](#)[Protein sequence analysis](#)[Sequence manipulation](#)

STANDARD TOOLS

[Get Data](#)[Send Data](#)[ENCODE Tools](#)[Lift-Over](#)[Text Manipulation](#)[Filter and Sort](#)

- [Filter](#) data on any column using simple expressions
- [Sort](#) data in ascending or descending order
- [Select](#) lines that match an expression

GFF

- [Extract features](#) from GFF file
- [Filter GFF file by attribute](#) using simple expressions
- [Filter GFF file by feature count](#) using simple expressions

[Join, Subtract and Group](#)[Convert Formats](#)

Filter

Filter:

2: Whisson et al. (2..ER with HMM)

Dataset missing? See TIP below.

With following condition:

c2 == 'Y'

Double equal signs, ==, must be used as shown above.
To filter for an arbitrary string, use the Select tool.

Execute

⚠ Double equal signs, ==, must be used as "equal to" (e.g., c1 == 'chr22')

ℹ **TIP:** Attempting to apply a filtering condition may throw exceptions if the data type (e.g., string, integer) in every line of the columns being filtered is not appropriate for the condition (e.g., attempting certain numerical calculations on strings). If an exception is thrown when applying the condition to a line, that line is skipped as invalid for the filter condition. The number of invalid skipped lines is documented in the resulting history item as a "Condition/data issue".

ℹ **TIP:** If your data is not TAB delimited, use [Text Manipulation->Convert](#)

Syntax

The filter tool allows you to restrict the dataset using simple conditional statements.

- Columns are referenced with **c** and a **number**. For example, **c1** refers to the first column of a tab-delimited file

History

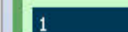
Options ▾



Unnamed history

2: Whisson et al. (2007)
RXLR-EER with HMM

19,805 lines, 1 comments
 format: tabular, database: ?
 Info: Whisson2007 for 19805 sequences:
 Y = 89, hmm = 43, neither = 19657, re = 16



1
 #ID
 jgi|Phyca11|532085|estExt2_fgenes1
 jgi|Phyca11|80978|gw1.4.1034.1
 jgi|Phyca11|99641|e_gw1.4.982.1
 jgi|Phyca11|13479|fgenes1_pg.PHYC
 jgi|Phyca11|99637|e_gw1.4.173.1

1:
Phyca11 filtered proteins.fasta

19,805 sequences
 format: fasta, database: ?
 Info: uploaded fasta file



>jgi|Phyca11|532085|estExt2_fgenes1
 MGNVYSTDSSSDTQQVERPEEKLHSLSDTTVT:
 GIRYTDQETKRKQGGNSPFLVSGVLTWIKACAG

Tools

Options ▾

LOCAL TOOLS

[Upload File from your computer](#)[NCBI BLAST+](#)[Protein sequence analysis](#)[Sequence manipulation](#)

STANDARD TOOLS

[Get Data](#)[Send Data](#)[ENCODE Tools](#)[Lift-Over](#)[Text Manipulation](#)[Filter and Sort](#)

- [Filter](#) data on any column using simple expressions
- [Sort](#) data in ascending or descending order
- [Select](#) lines that match an expression

GFF

- [Extract features](#) from GFF file
- [Filter GFF file by attribute](#) using simple expressions
- [Filter GFF file by feature count](#) using simple expressions

[Join, Subtract and Group](#)[Convert Formats](#)

```
jgi|Phycall|5186|fgenesl_pm.PHYCAscaffold_4_#
_274 Y
jgi|Phycall|130393|e_gwl.93.9.1 Y
jgi|Phycall|109432|e_gwl.16.672.1 Y
jgi|Phycall|129643|e_gwl.86.164.1 Y
jgi|Phycall|533084|estExt2_fgenesl_pg.C_PHYCA
scaffold_100082 Y
jgi|Phycall|14941|fgenesl_pg.PHYCAscaffold_10
_#_87 Y
jgi|Phycall|14944|fgenesl_pg.PHYCAscaffold_10
_#_90 Y
jgi|Phycall|14948|fgenesl_pg.PHYCAscaffold_10
_#_94 Y
jgi|Phycall|129113|e_gwl.81.47.1 Y
jgi|Phycall|129145|e_gwl.81.173.1 Y
jgi|Phycall|129044|e_gwl.81.43.1 Y
jgi|Phycall|15117|fgenesl_pg.PHYCAscaffold_11
_#_103 Y
jgi|Phycall|102742|e_gwl.7.224.1 Y
jgi|Phycall|116585|e_gwl.31.283.1 Y
jgi|Phycall|116645|e_gwl.31.473.1 Y
jgi|Phycall|39353|gwl.107.45.1 Y
jgi|Phycall|97196|e_gwl.1.556.1 Y
jgi|Phycall|538116|estExt2_GenewiselPlus.C_PHY
CAscaffold_10381 Y
jgi|Phycall|4454|fgenesl_pm.PHYCAscaffold_2_#
_50 Y
jgi|Phycall|118417|e_gwl.36.500.1 Y
jgi|Phycall|103340|e_gwl.8.893.1 Y
jgi|Phycall|20942|fgenesl_pg.PHYCAscaffold_77
_#_10 Y
jgi|Phycall|20944|fgenesl_pg.PHYCAscaffold_77
_#_12 Y
jgi|Phycall|102326|e_gwl.6.392.1 Y
jgi|Phycall|101904|e_gwl.6.942.1 Y
jgi|Phycall|14853|fgenesl_pg.PHYCAscaffold_9_
_#_229 Y
```

History

Options ▾



Unnamed history

3: Filter on data 2



89 lines

format: tabular, database: ?

Info: Filtering with c2=='Y',

kept 0.45% of 19806 lines.

Skipped 1 invalid lines starting at

line #1: "#ID Whisson2007"



1

jgi|Phycall|5186|fgenesl_pm.PHYCAs

jgi|Phycall|130393|e_gwl.93.9.1

jgi|Phycall|109432|e_gwl.16.672.1

jgi|Phycall|129643|e_gwl.86.164.1

jgi|Phycall|533084|estExt2_fgenesl

jgi|Phycall|14941|fgenesl_pg.PHYC

2: Whisson et al. (2007)



RXLR-EER with HMM

19,805 lines, 1 comments

format: tabular, database: ?

Info: Whisson2007 for 19805

sequences:

Y = 89, hmm = 43, neither =

19657, re = 16

1

Tools

Options ▾

LOCAL TOOLS

[Upload File from your computer](#)[NCBI BLAST+](#)[Protein sequence analysis](#)[Sequence manipulation](#)

STANDARD TOOLS

[Get Data](#)[Send Data](#)[ENCODE Tools](#)[Lift-Over](#)[Text Manipulation](#)[Filter and Sort](#)

- [Filter](#) data on any column using simple expressions
- [Sort](#) data in ascending or descending order
- [Select](#) lines that match an expression

GFF

- [Extract features](#) from GFF file
- [Filter GFF file by attribute](#) using simple expressions
- [Filter GFF file by feature count](#) using simple expressions

[Join, Subtract and Group](#)[Convert Formats](#)

RXLR Motifs

FASTA file of protein sequences:

1: Phyca11_filtered_teins.fasta ▾

Which RXLR model?:

Whisson et al. (2007) RXLR-EER with HMM ▾

Execute

Background

Many effector proteins from Oomycete plant pathogens for manipulating the host have been found to contain a signal peptide followed by a conserved RXLR motif (Arg, any amino acid, Leu, Arg), and then sometimes EER (Glu, Glu, Arg). There are striking parallels with the malarial host-targeting signal (Plasmodium export element, or "Pexel" for short).

What it does

Takes a protein sequence FASTA file as input, and produces a simple tabular file as output with one line per protein, and two columns giving the sequence ID and the predicted class. This is typically just whether or not it had the selected RXLR motif (Y or N).

Bhattacharjee et al. (2006) RXLR Model

Looks for the oomycete motif RXLR as described in Bhattacharjee et al. (2006).

Matches must have a SignalP Hidden Markov Model (HMM) score of at least 0.9, a SignalP Neural Network (NN) predicted cleavage site giving a signal peptide length between 10 and 40 amino acids inclusive, and the RXLR pattern must be after but

History

Options ▾

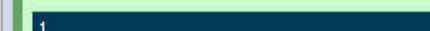


Unnamed history

3: Filter on data 2



89 lines
format: tabular, database: ?
Info: Filtering with c2=='Y', kept 0.45% of 19806 lines. Skipped 1 invalid lines starting at line #1: "#ID Whisson2007"



1

jgi|Phyca11|5186|fgenes1_pm.PHYCA

jgi|Phyca11|130393|e_gw1.93.9.1

jgi|Phyca11|109432|e_gw1.16.672.1

jgi|Phyca11|129643|e_gw1.86.164.1

jgi|Phyca11|533084|estExt2_fgensesh

jgi|Phyca11|14941|fgensesh1_pg.PHYC

Tools

Options ▾

LOCAL TOOLS

[Upload File from your computer](#)[NCBI BLAST+](#)[Protein sequence analysis](#)[Sequence manipulation](#)

STANDARD TOOLS

[Get Data](#)[Send Data](#)[ENCODE Tools](#)[Lift-Over](#)[Text Manipulation](#)[Filter and Sort](#)

- [Filter](#) data on any column using simple expressions

- [Sort](#) data in ascending or descending order

- [Select](#) lines that match an expression

GFF

- [Extract features](#) from GFF file

- [Filter GFF file by attribute](#) using simple expressions

- [Filter GFF file by feature count](#) using simple expressions

[Join, Subtract and Group](#)[Convert Formats](#)

RXLR Motifs

FASTA file of protein sequences:

1: Phyca11_filtered_teins.fasta ▾

Which RXLR model?:

Win et al. (2007) RXLR ▾

Execute

Background

Many effector proteins from Oomycete plant pathogens for manipulating the host have been found to contain a signal peptide followed by a conserved RXLR motif (Arg, any amino acid, Leu, Arg), and then sometimes EER (Glu, Glu, Arg). There are striking parallels with the malarial host-targeting signal (Plasmodium export element, or "Pexel" for short).

What it does

Takes a protein sequence FASTA file as input, and produces a simple tabular file as output with one line per protein, and two columns giving the sequence ID and the predicted class. This is typically just whether or not it had the selected RXLR motif (Y or N).

Bhattacharjee et al. (2006) RXLR Model

Looks for the oomycete motif RXLR as described in Bhattacharjee et al. (2006).

Matches must have a SignalP Hidden Markov Model (HMM) score of at least 0.9, a SignalP Neural Network (NN) predicted cleavage site giving a signal peptide length between 10 and 40 amino acids inclusive, and the RXLR pattern must be after but

History

Options ▾



Unnamed history

3: Filter on data 2

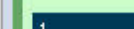


89 lines

format: tabular, database: ?

Info: Filtering with c2=='Y', kept 0.45% of 19806 lines.

Skipped 1 invalid lines starting at line #1: "#ID Whisson2007"



```
1
jgi|Phyca11|5186|fgenes1_pm.PHYCA
jgi|Phyca11|130393|e_gw1.93.9.1
jgi|Phyca11|109432|e_gw1.16.672.1
jgi|Phyca11|129643|e_gw1.86.164.1
jgi|Phyca11|533084|estExt2_fgenes1
jgi|Phyca11|14941|fgenes1_pg.PHYCA
```

2: Whisson et al. (2007)



RXLR-EER with HMM

19,805 lines, 1 comments

format: tabular, database: ?

Info: Whisson2007 for 19805 sequences:

Y = 89, hmm = 43, neither = 19657, re = 16



```
1
ATP
```

Next few steps omitted...

- Repeated RXLR search & filter using other two models
- Labelled some history entries

Galaxy

http://ppserver/galaxy/root

Galaxy

Analyze Data Workflow Shared Data Admin Help User

Tools Options

- Fetch Sequences
- Fetch Alignments
- Statistics
- Wavelet Analysis
- Graph/Display Data
 - Histogram of a numeric column
 - Scatterplot of two numeric columns
 - Bar chart for multiple columns
 - Plotting tool for multiple series and graph types
 - Boxplot of quality statistics
 - GMAJ Multiple Alignment Viewer
 - LAI Pairwise Alignment Viewer
 - Build custom track for UCSC genome browser
 - VCF to MAF Custom Track for display at UCSC
 - Mutation Visualization
 - Venn Diagram from lists
- Regional Variation
- Multiple regression
- Multivariate Analysis
- Evolution
- Metagenomic analyses
- FASTA manipulation

Attributes updated

History Options

Unnamed history

- 7: Bhattacharjee matches
- 6: Bhattacharjee et al. (2006) RXLR
- 5: Win matches
- 4: Win et al. (2007) RXLR
- 3: Whisson matches
- 2: Whisson et al. (2007) RXLR-EER with HMM
- 1: Phyca11 filtered proteins.fasta
 - 19,805 sequences
 - format: fasta, database: ?
 - Info: uploaded fasta file
 - >jgi|Phyca11|532085|estExt2_fgenes1
MGNVYSTDSSSSDTQQVERPEEKLHSLSDTTVT:
GIRYTDQTKRKQGGNSPFLVSGVLTWIKACAG
LNDVVL
 - >jgi|Phyca11|80978|gw1.4.1034.1
DVFLDIGSGVGNVVAQFALSTKTRACIGIEIRRV

Python script using rpy

R library limma handles the plotting

Wrapper is on the Galaxy Tool Shed

Tools

Options ▾

Fetch SequencesFetch AlignmentsStatisticsWavelet AnalysisGraph/Display Data

- Histogram of a numeric column
- Scatterplot of two numeric columns
- Bar chart for multiple columns
- Plotting tool for multiple series and graph types
- Boxplot of quality statistics
- GMAJ Multiple Alignment Viewer
- LAI Pairwise Alignment Viewer
- Build custom track for UCSC genome browser
- VCF to MAF Custom Track for display at UCSC
- Mutation Visualization
- Venn Diagram from lists

Regional VariationMultiple regressionMultivariate AnalysisEvolutionMetagenomic analysesFASTA manipulation

Caption for set:

Bhattacharjee et al

Remove Sets 1

Sets 2

Members of set:

5: Win matches

Tabular file (uses column one), FASTA, FASTQ or SFF file.

Caption for set:

Win et al

Remove Sets 2

Sets 3

Members of set:

3: Whisson matches

Tabular file (uses column one), FASTA, FASTQ or SFF file.

Caption for set:

Whisson et al

Remove Sets 3

Add new Sets

Execute

History

Options ▾



Unnamed history

7: Bhattacharjee matches



6: Bhattacharjee et al. (2006) RXLR



5: Win matches



4: Win et al. (2007) RXLR



3: Whisson matches



2: Whisson et al. (2007) RXLR-EER with HMM



1: Phyca11 filtered proteins.fasta



19,805 sequences
format: fasta, database: 2
Info: uploaded fasta file



```
> jgi | Phyca11 | 532085 | estExt2_fgenes1
MGNVYSTDSSSSDTQQVERPEEKLHSLSDTTVT:
GIRYTDQTKRKQGGNSPFLVSGVLTWIKACAG
LNDVVL
```

```
> jgi | Phyca11 | 80978 | gw1.4.1034.1
DVFLDIGSGVGNVVAQFALSTKTRACIGIEIRRV
```


Tools

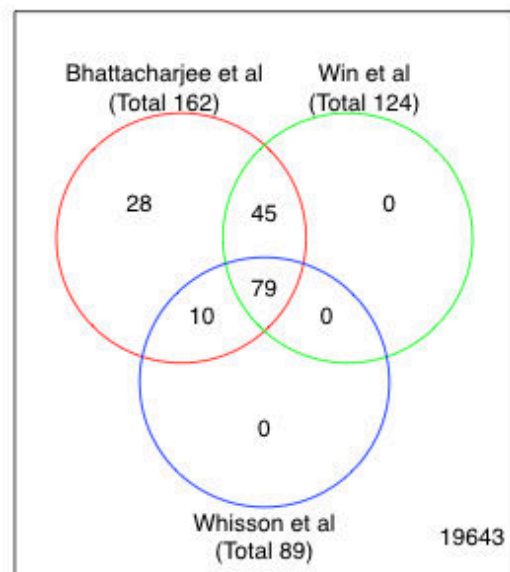
Options ▾

Fetch SequencesFetch AlignmentsStatisticsWavelet AnalysisGraph/Display Data

- Histogram of a numeric column
- Scatterplot of two numeric columns
- Bar chart for multiple columns
- Plotting tool for multiple series and graph types
- Boxplot of quality statistics
- GMAJ Multiple Alignment Viewer
- LAI Pairwise Alignment Viewer
- Build custom track for UCSC genome browser
- VCF to MAF Custom Track for display at UCSC
- Mutation Visualization
- Venn Diagram from lists

Regional VariationMultiple regressionMultivariate AnalysisEvolutionMetagenomic analysesFASTA manipulation

Venn Diagram



History

Options ▾



Unnamed history

8: Venn Diagram on data 7, data 1, and others

19.5 Kb
 format: pdf, database: ?
 Info: Doing 3-way Venn Diagram
 Total of 19805 IDs
 162 in Bhattacharjee et al
 124 in Win et al
 89 in Whisson et al

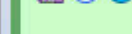
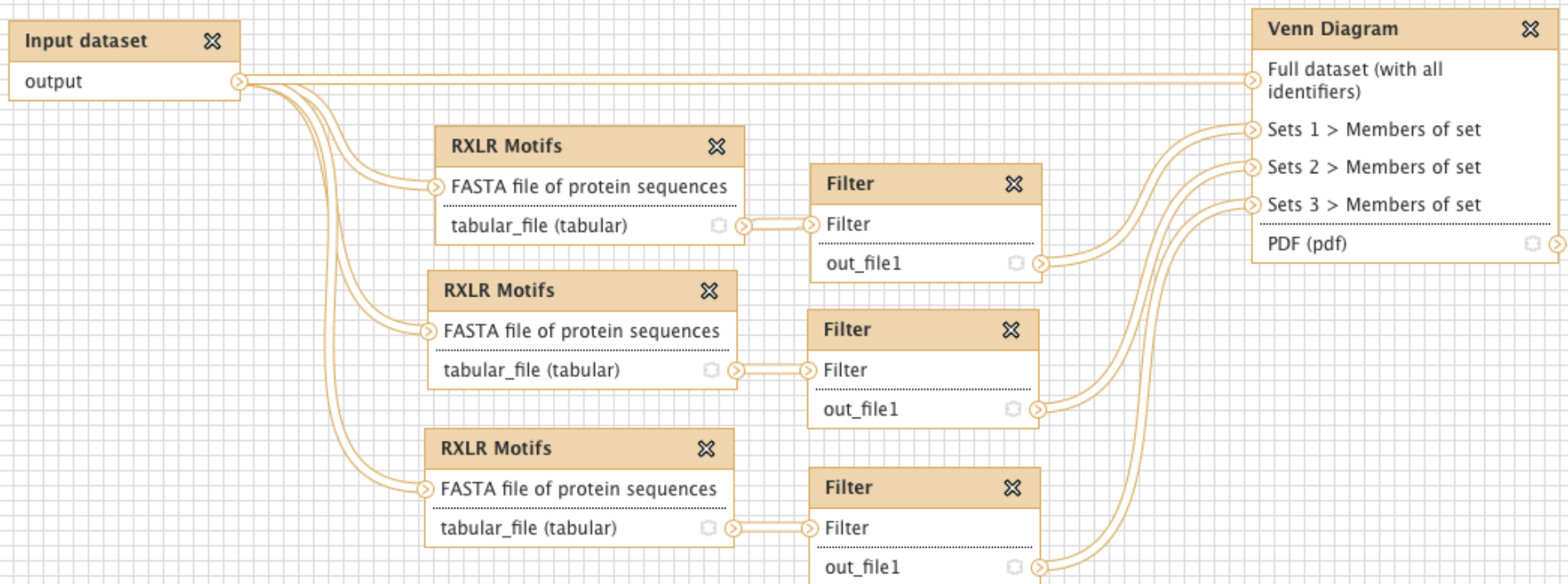


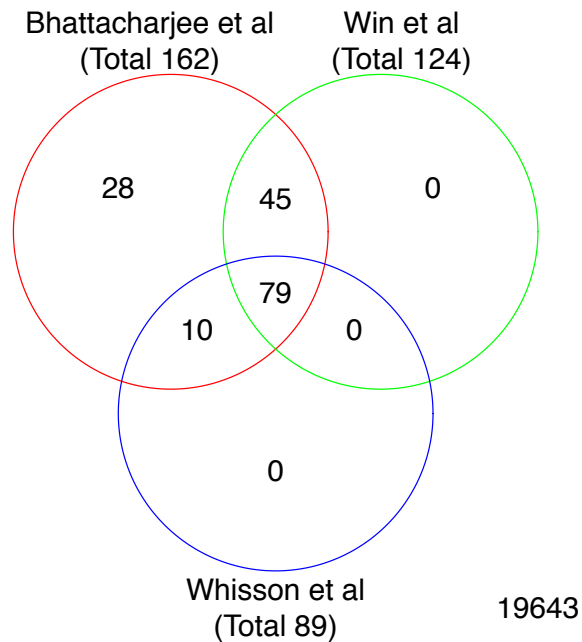
Image in pdf format

7: Bhattacharjee matches
6: Bhattacharjee et al. (2006) RXLR
5: Win matches
4: Win et al. (2007) RXLR
3: Whisson matches
2: Whisson et al. (2007) RXLR-EER with HMM
1:

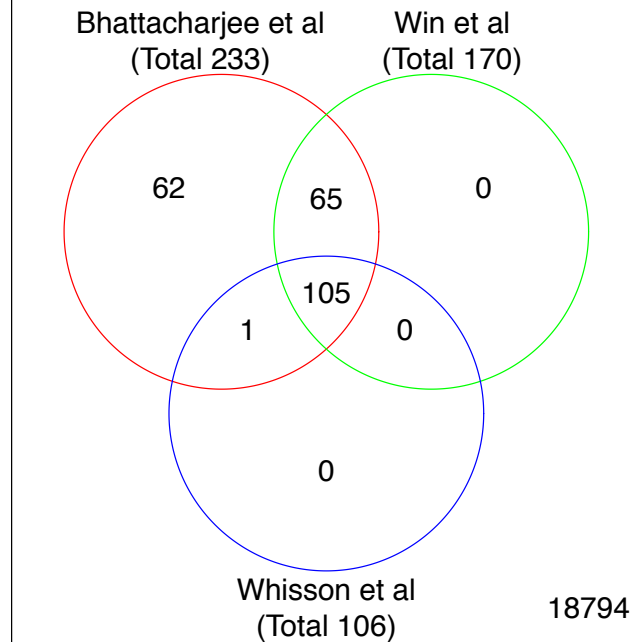
Repeating analysis as a Workflow



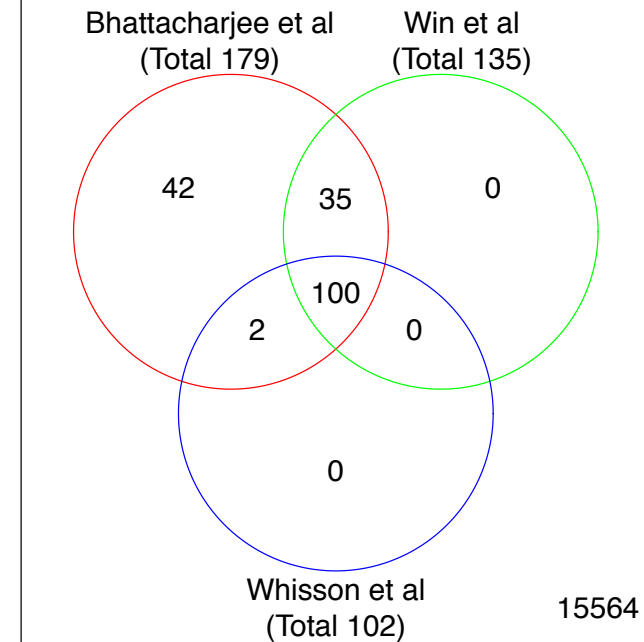
RXLRs in *Phytophthora* draft genomes



P. capsici
(19,805 proteins)



P. sojae
(19,027 proteins)

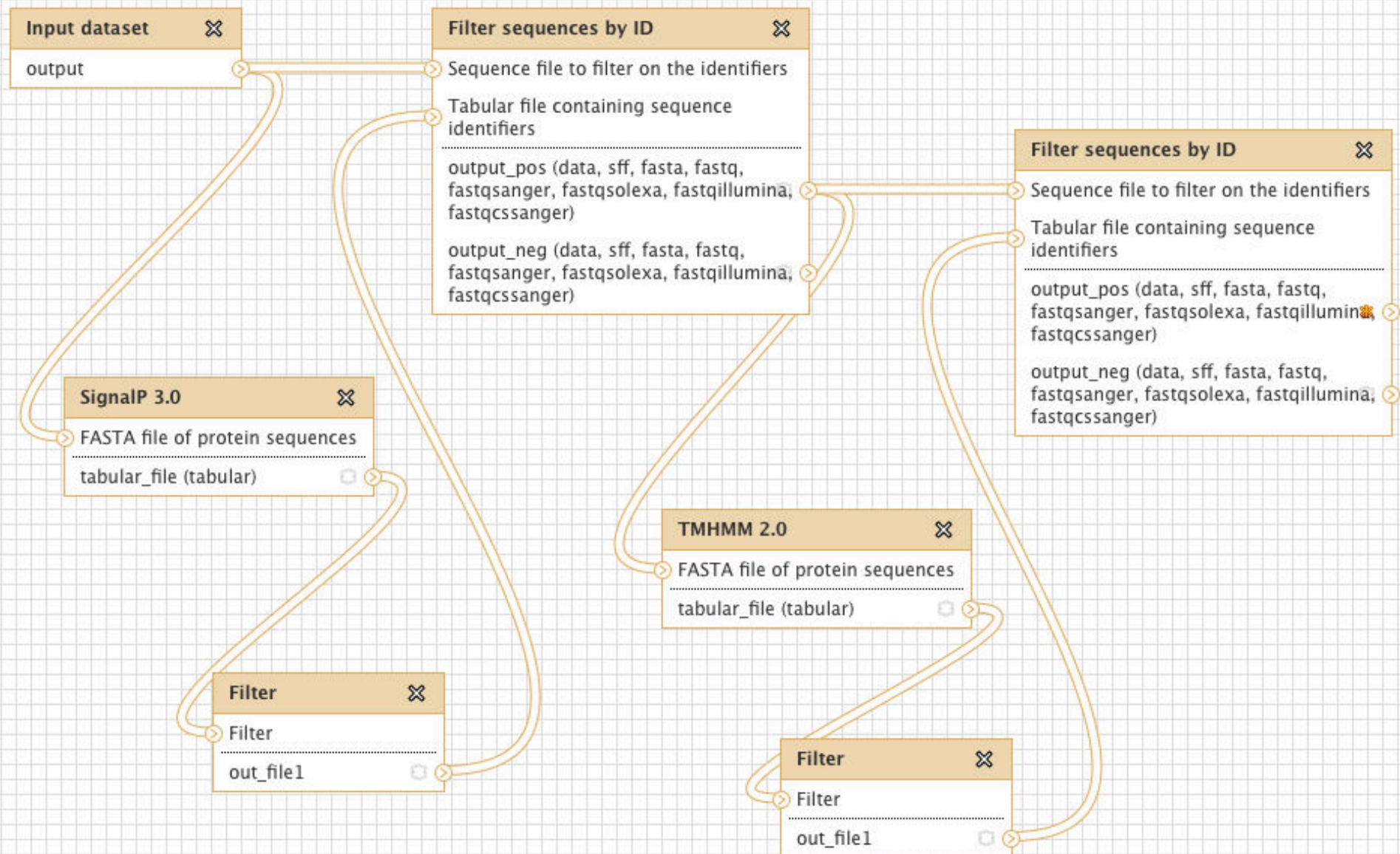


P. ramorum
(15,743 proteins)

Example – Finding effector proteins

- Start with a FASTA file of proteins
- Run signal peptide prediction
- Select proteins with signal peptide
- Run transmembrane prediction
- Select proteins without transmembrane (TM) domains
- Get FASTA file of proteins with signal peptide but not TM

Workflow Editor – Effector finding



Acknowledgements

- Helpful tool authors:

- Alex Nguyen (NLStradamus)

- Laszlo Kajan (PredictNLS)

- Peter Troshin, Michelle Scott (NoD)

- JHI Testers:

- John Jones, Remco Stam, Julietta Jupe

- The Galaxy Developers & mailing list community