Multiple Sequence Alignment

with

PASTA

Michael Nute

Austin, TX

June 17, 2016

Agenda

- Installation recap & Dendropy version issues
- Quick recap of PASTA Algorithm
- Run the GUI
- Explore GUI options and what they do in terms of PASTA
- Run a test alignment
- Explore PASTA outputs and diagnostics
- Run a *different* test alignment
- Compare the PASTA fill-in-the-blank defaults for the two test alignments

PASTA: Installation

We hope everybody has been able to install PASTA based on instructions from our email. If not:

See detailed installation instructions at:

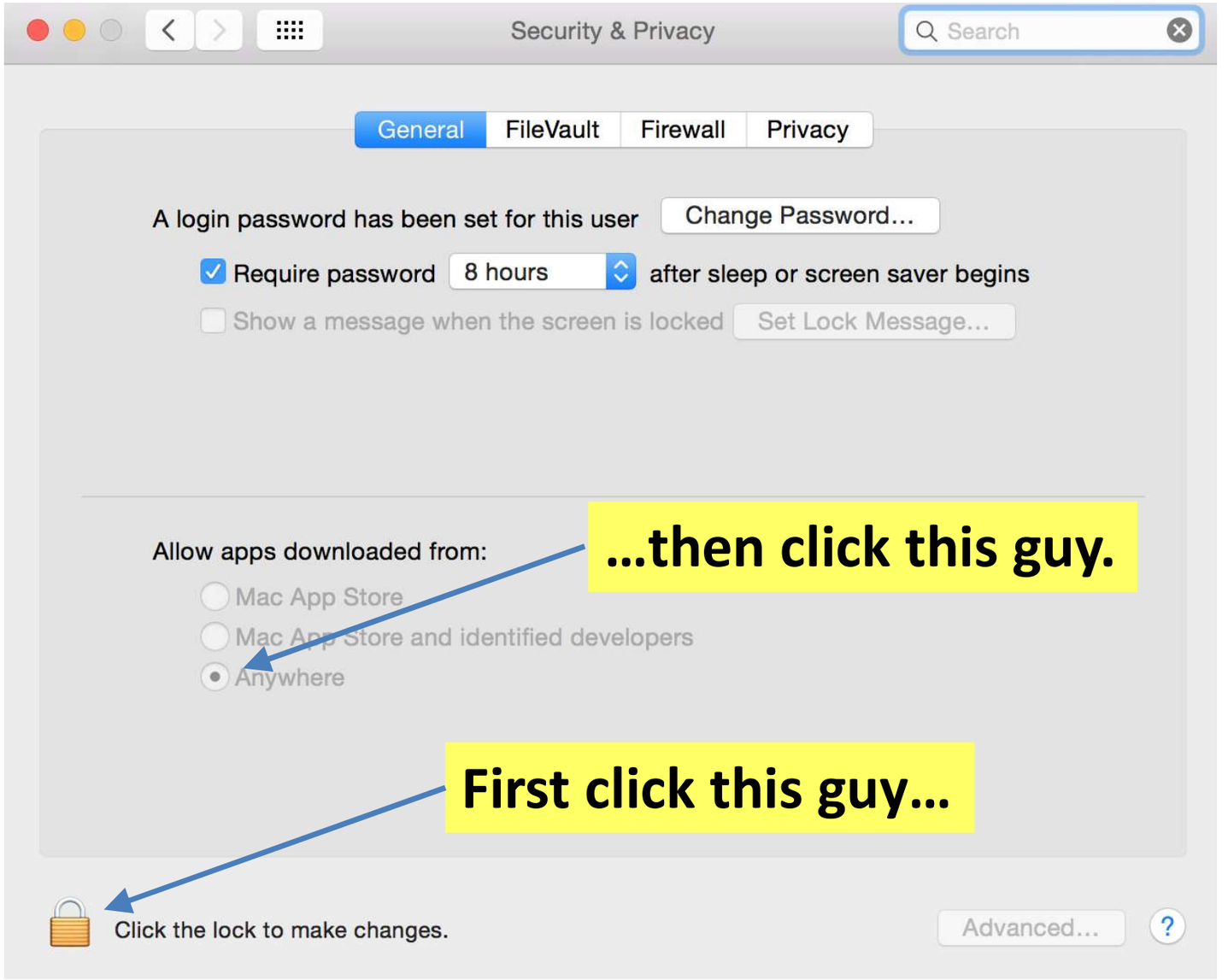
https://github.com/smirarab/pasta

Three Options:

1) MAC
   - DMG file available at the link above
2) Linux
   - Detailed instructions available at the link above
   - Requires JAVA, wxPython,
3) Virtual Machine (Recommended: VirtualBox)
   - Virtual appliance available at link above
   - This is the only option for Windows users

PASTA: Installing with a Mac

- If you have a mac, download the .dmg file (link is on the github page).
- Open the dmg and copy the app to a folder you want to use.
- You may need to update settings to allow you to run software downloaded from the internet. Go to **System Preferences → Security & Privacy** and you will see the screen on the right:



PASTA: Dealing with DendroPy version issues

- PASTA currently depends on an older version of DendroPy (3.12) and will not run if the newest version (4.0+) is the primary installed version
- To fix this, we need to find a suitable location, download the old code and point Python to that location before running PASTA.
  - Easiest way:
    - Go to your pasta-code folder (the one containing the “pasta” install folder)
    - Download the dendropy 3.12 source & extract it
    - Link that folder to the root pasta folder.
  - Sample commands. See: http://publish.illinois.edu/michaelnute/files/2014/10/commands_to_fix_dendropy.txt

```
1. wget https://github.com/jeetsukumaran/DendroPy/archive/v3.12.1.zip
2. unzip v3.12.1.zip
3. ls –l DendroPy-3.12.1 <-- VERIFY THAT “dendropy” SUBFOLDER EXISTS
4. dpydir=$(pwd)/DendroPy-3.12.1/dendropy
```

PASTA: Dealing with DendroPy version issues

At this point you have two options:

1. Establish a link to the DendroPy 3.12 in the pasta install folder
   - This is the easiest and is a permanent solution, but it might cause this to be your default DendroPy installation, which would break anything using DendroPy 4.*
   - **Command**:

```
ln -s $dpydir ./pasta/dendropy
```

2. Manually add this to your python path
   - This won't break anything, but you'll have to do this every time you want to use PASTA
   - **Command**:

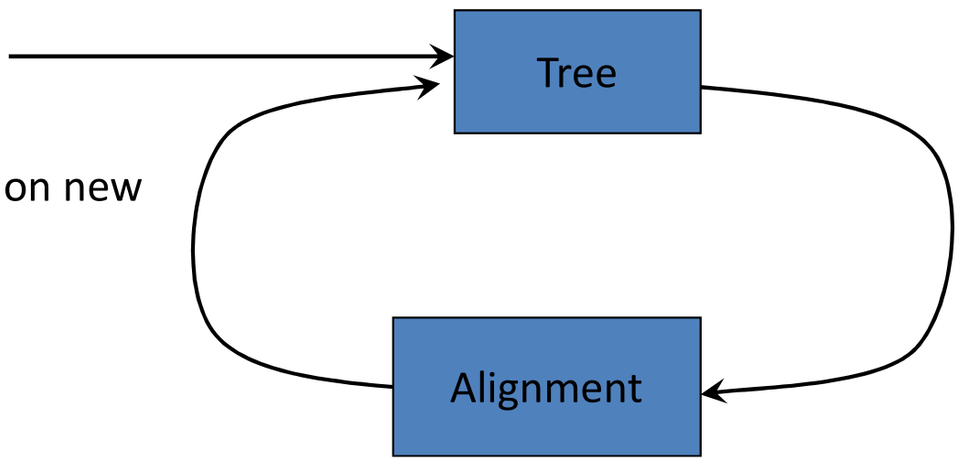
```
export PYTHONPATH=$dpydir:$PYTHONPATH
```

Test your installation: `python ./pasta/run_pasta.py -h`

I promise that a permanent, easy fix for this is coming soon.

SATé and PASTA Algorithms

Obtain initial alignment and estimated ML tree



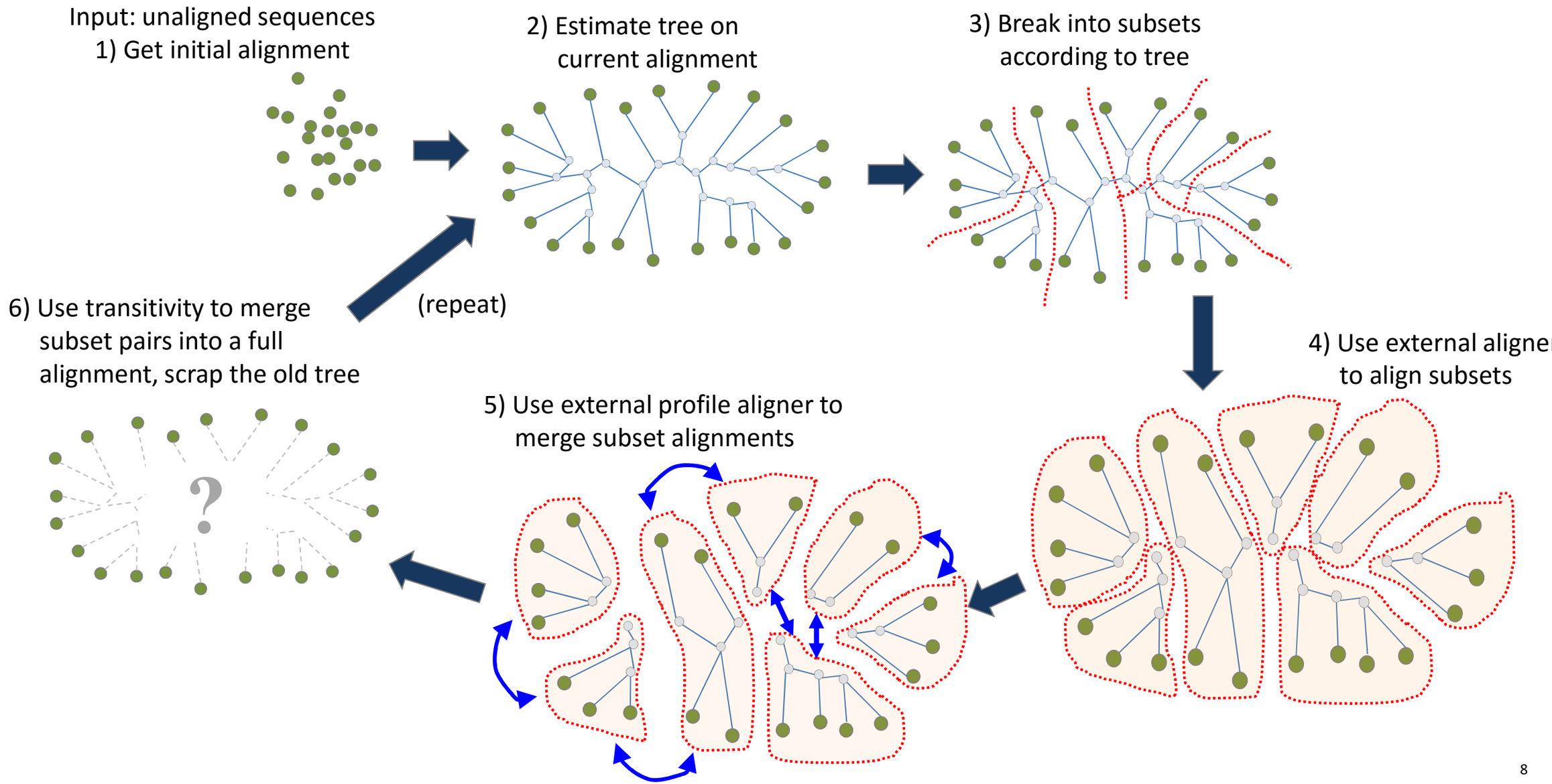
Estimate ML tree on new alignment

Use tree to compute new alignment

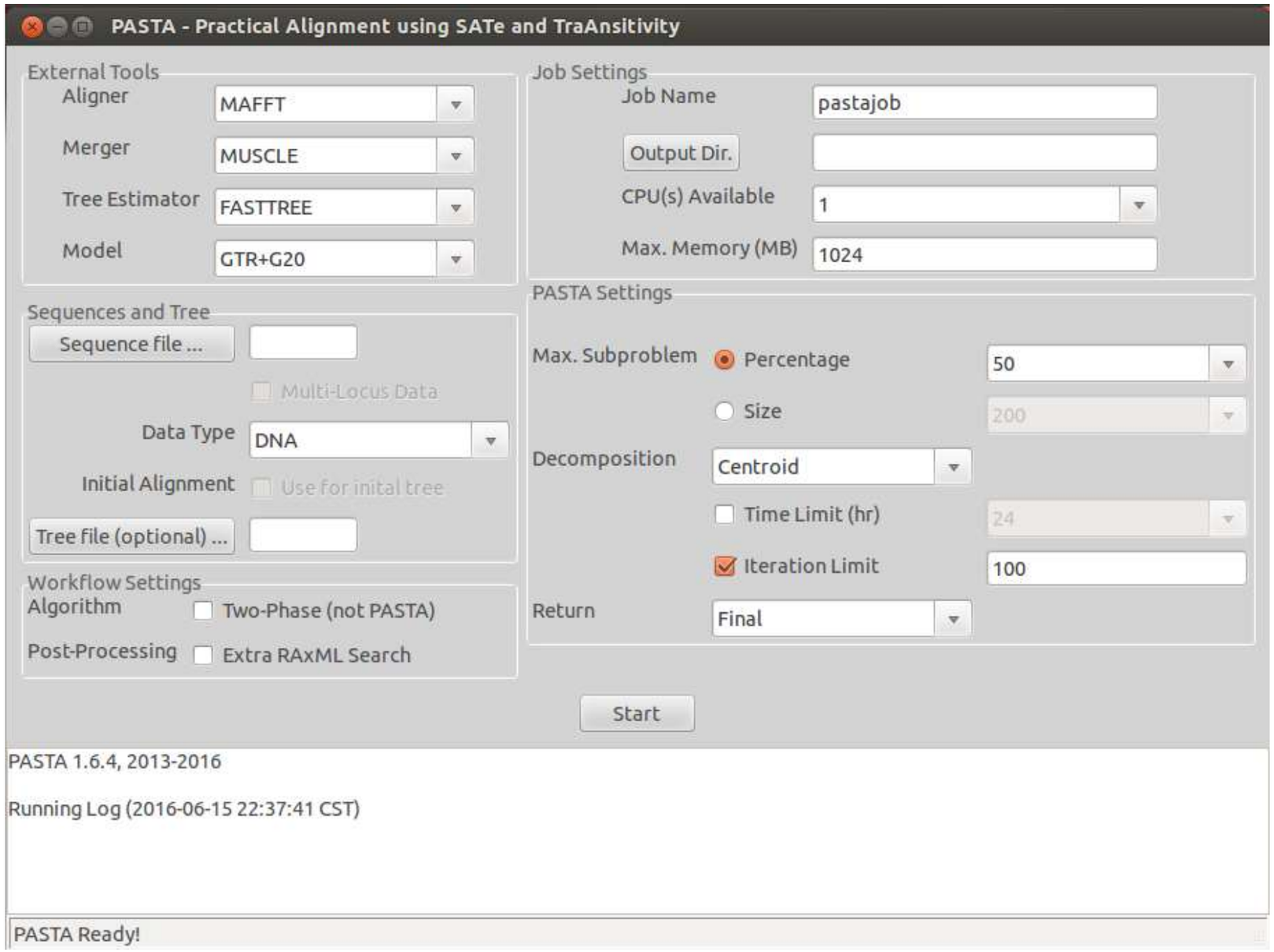
Repeat until termination condition, and

return the alignment/tree pair with the best ML score

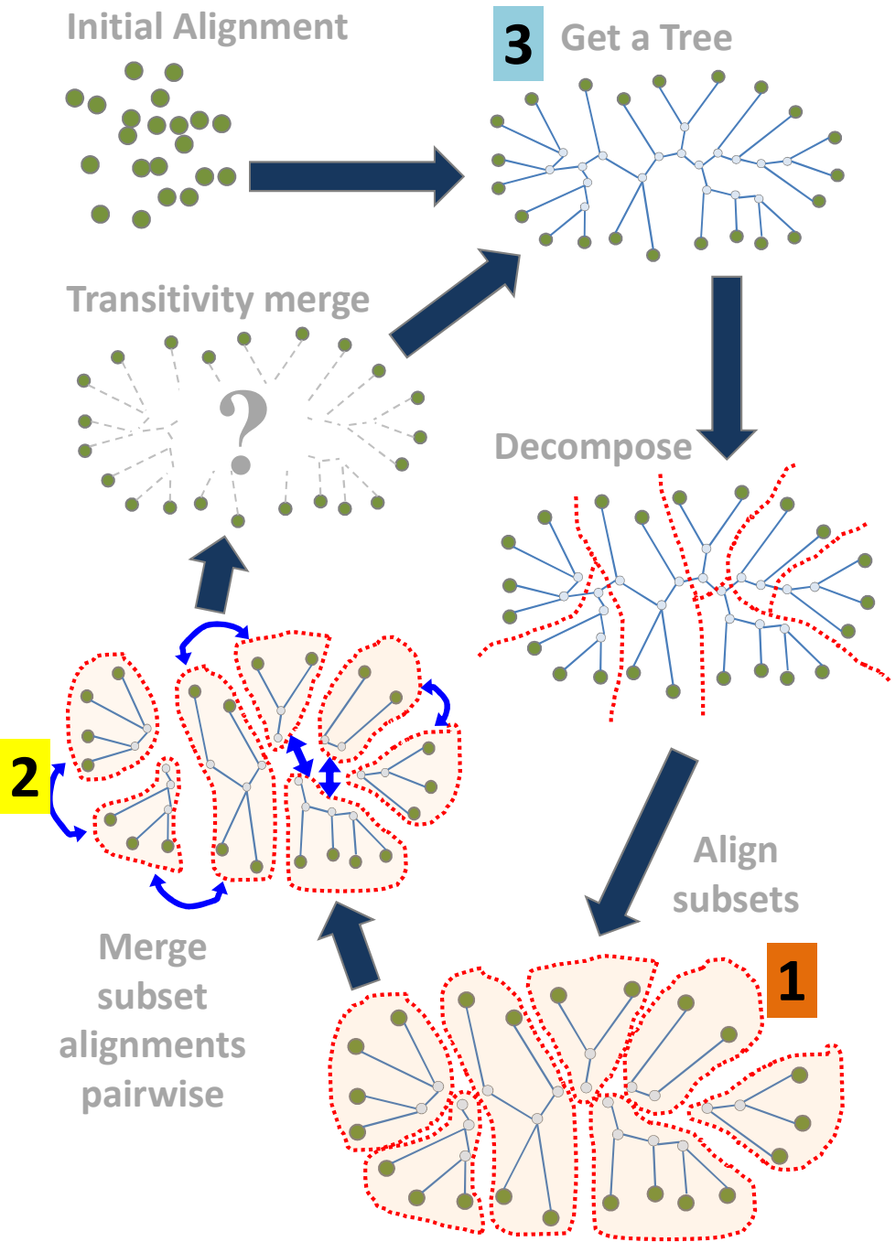
PASTA Algorithm



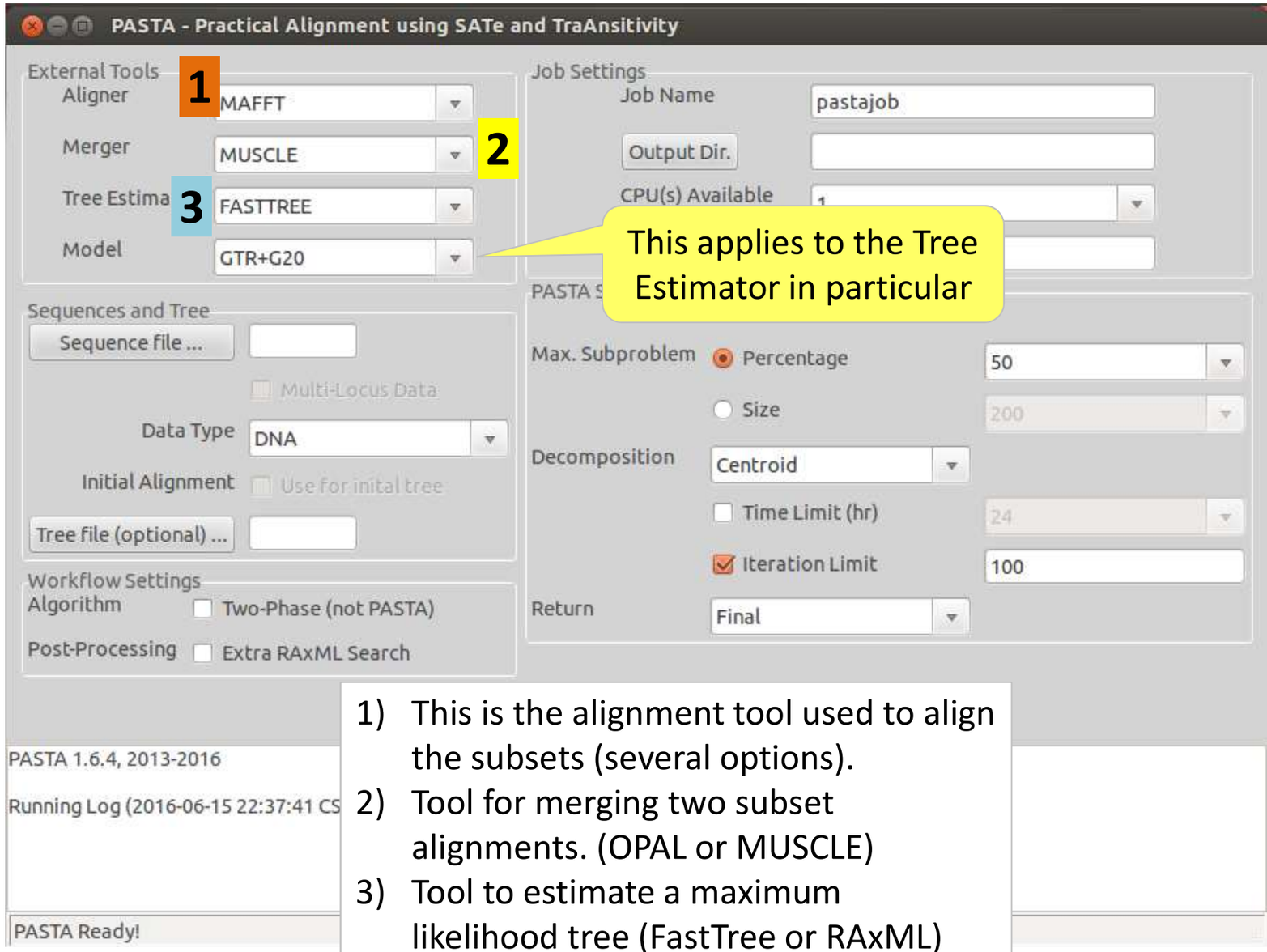
PASTA GUI (Linux version, but Mac looks the same)



PASTA Algorithm

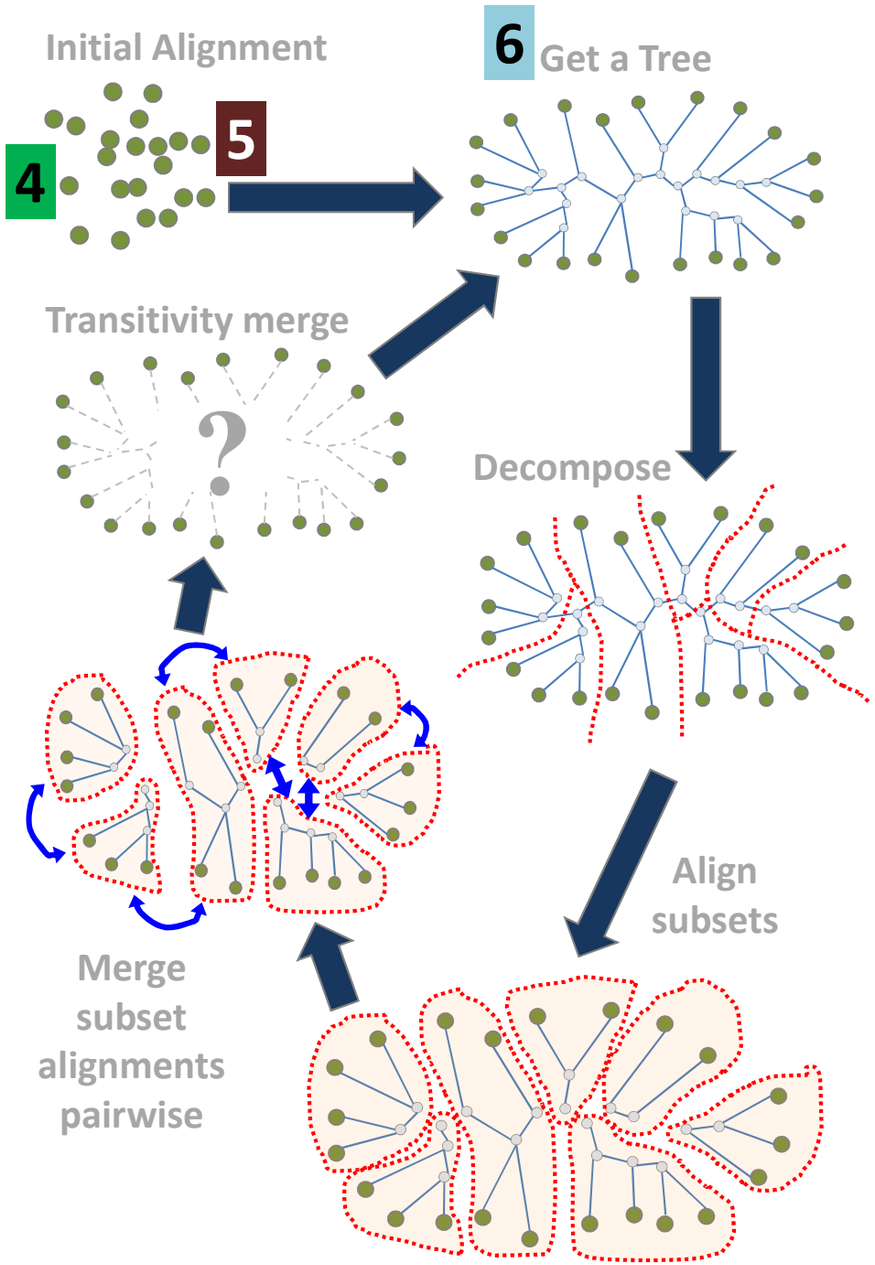


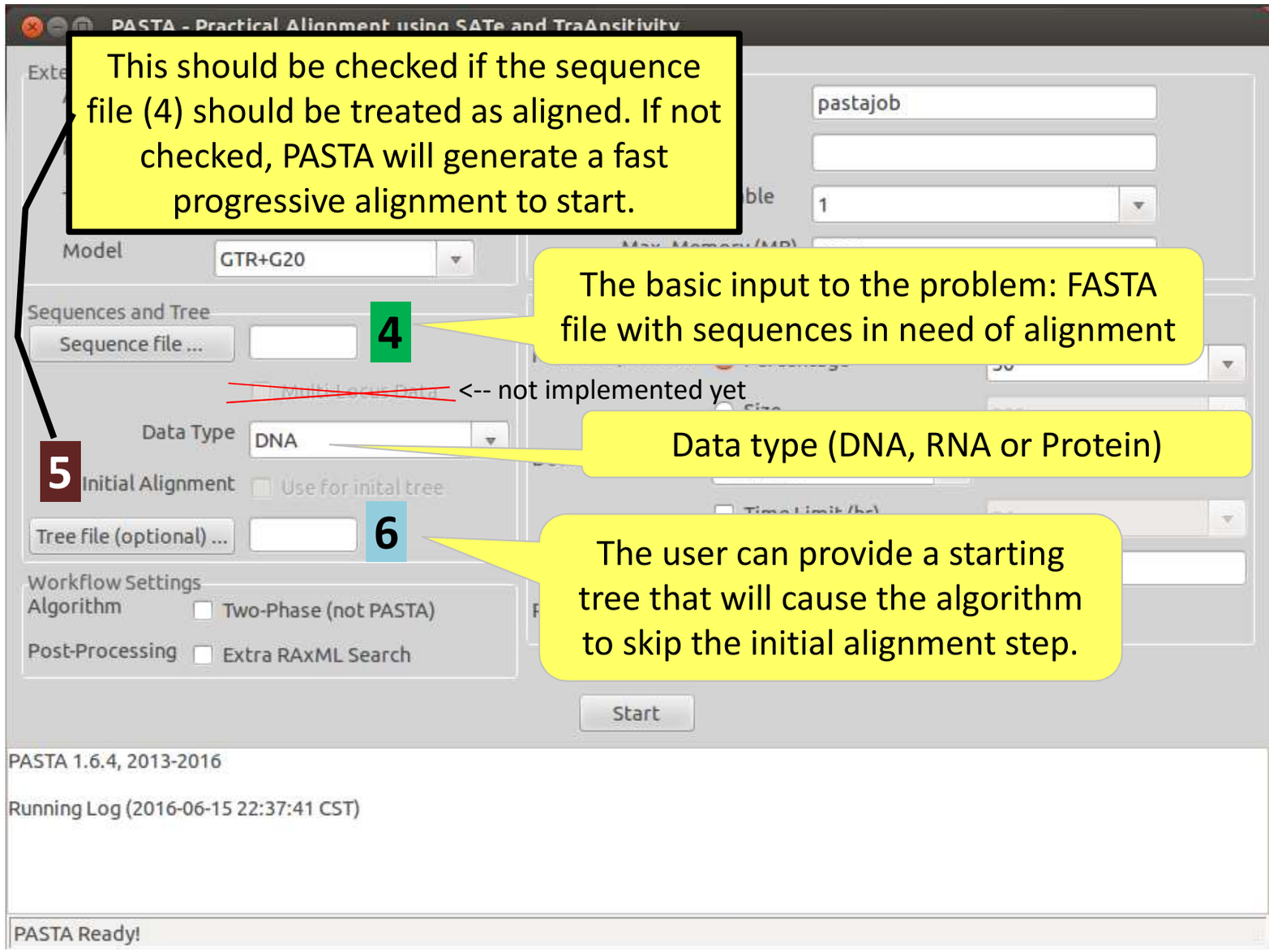
PASTA GUI



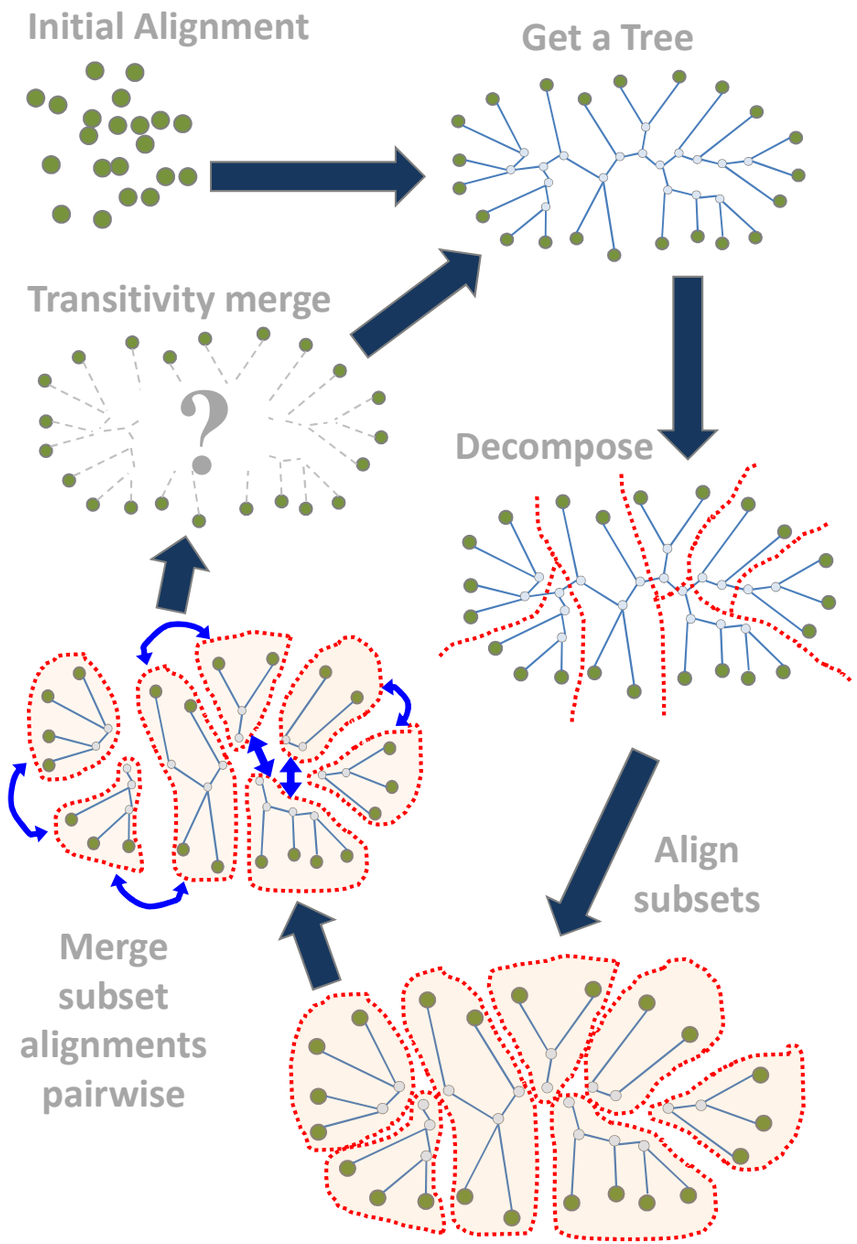
1) This is the alignment tool used to align the subsets (several options).
2) Tool for merging two subset alignments. (OPAL or MUSCLE)
3) Tool to estimate a maximum likelihood tree (FastTree or RAxML)

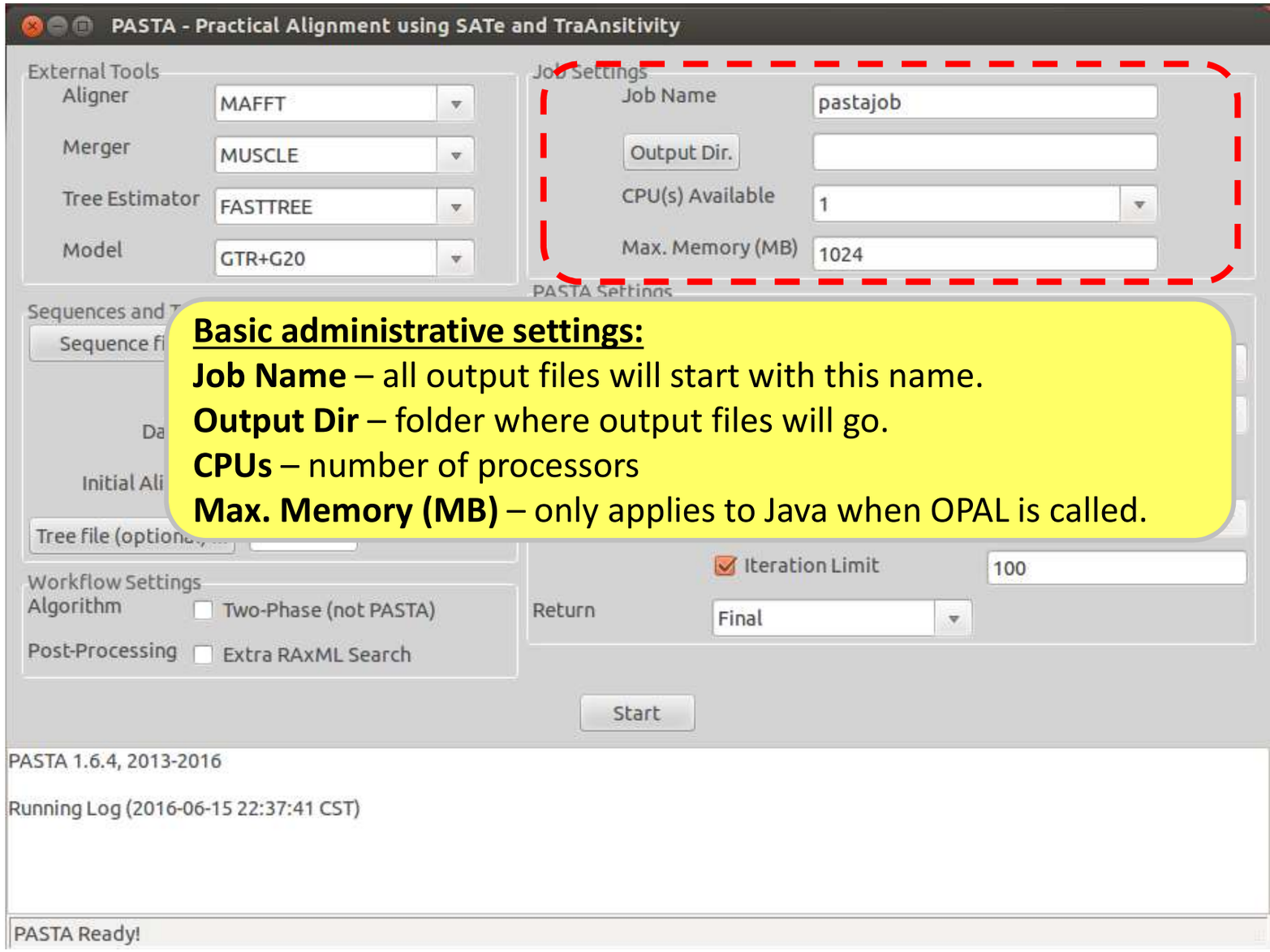
PASTA Algorithm





PASTA Algorithm





Basic administrative settings:

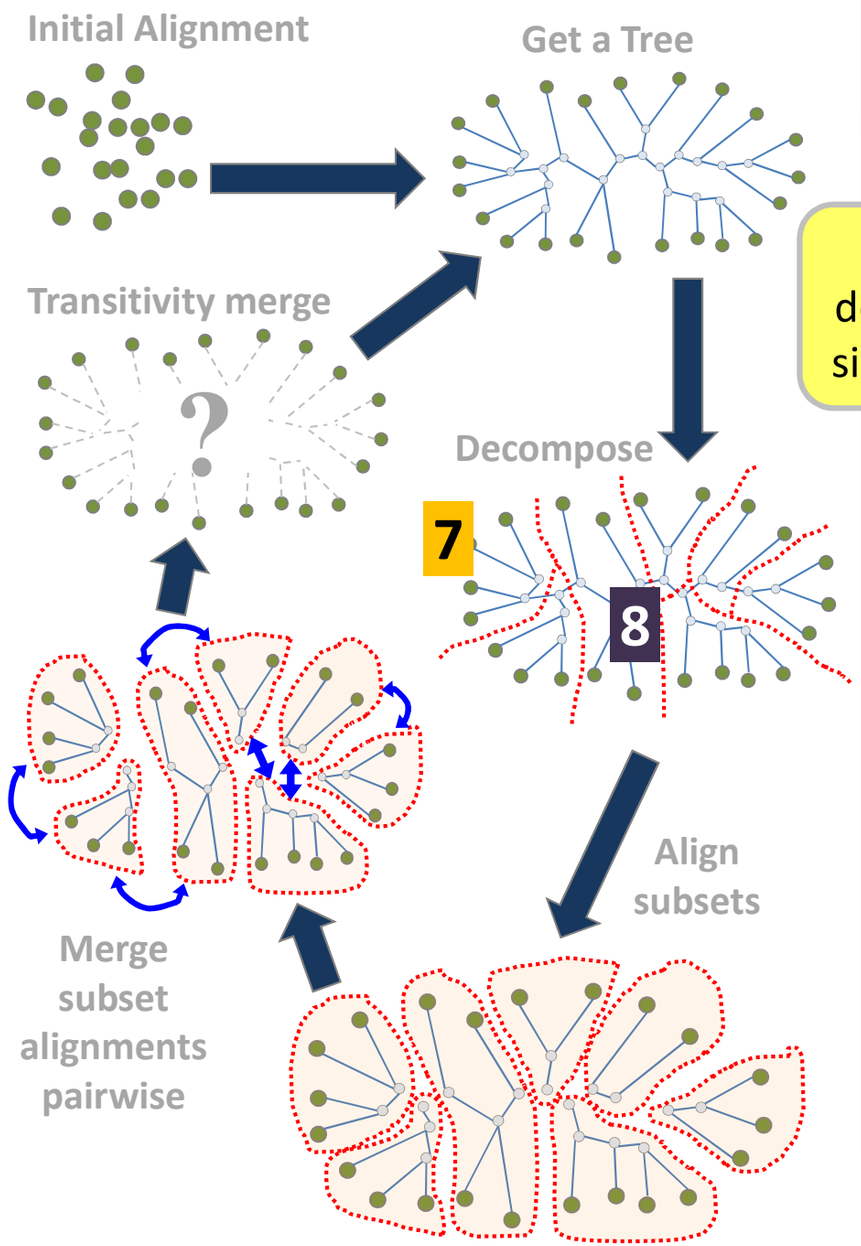
Job Name – all output files will start with this name.

Output Dir – folder where output files will go.

CPUs – number of processors

Max. Memory (MB) – only applies to Java when OPAL is called.

PASTA Algorithm

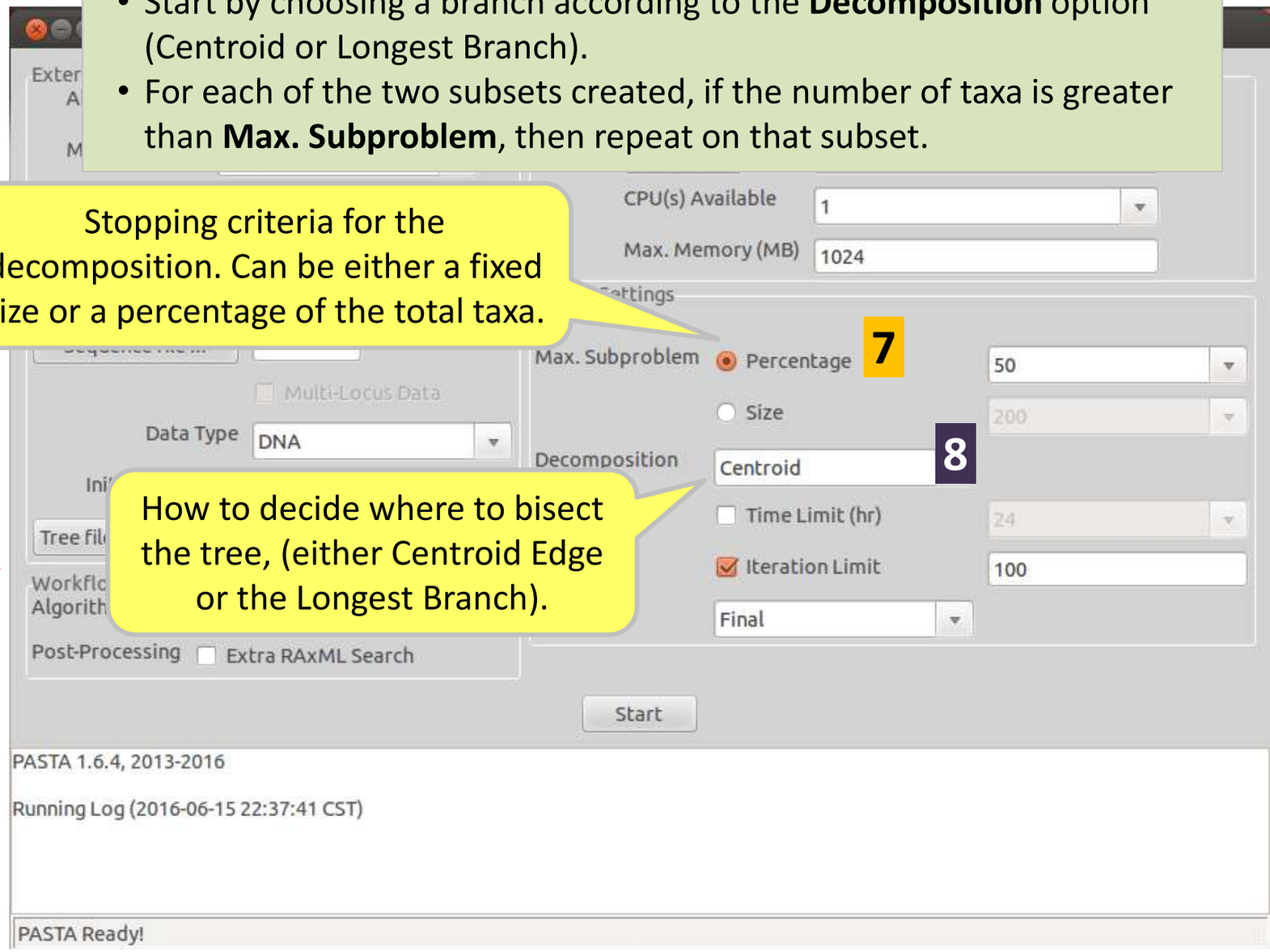


Decomposition Steps:

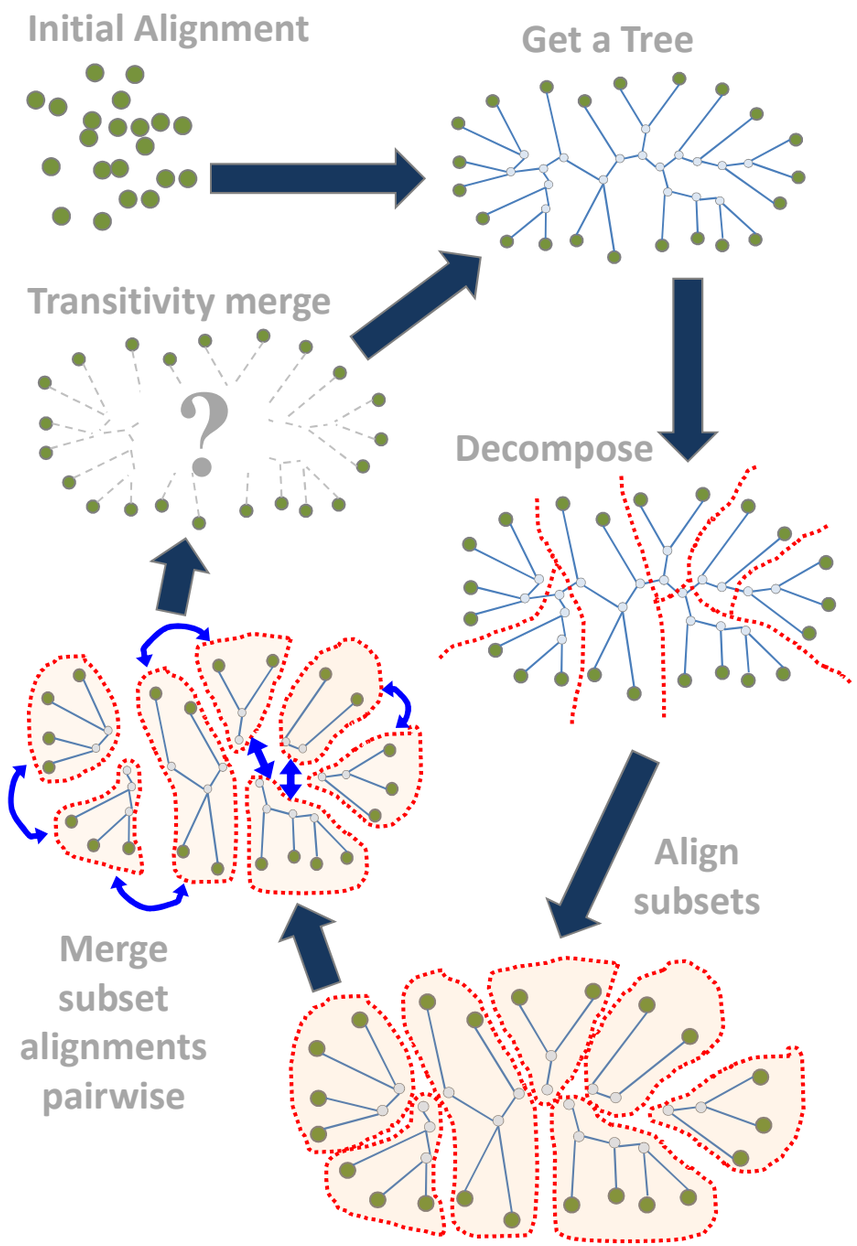
- Start by choosing a branch according to the **Decomposition** option (Centroid or Longest Branch).
- For each of the two subsets created, if the number of taxa is greater than **Max. Subproblem**, then repeat on that subset.

Stopping criteria for the decomposition. Can be either a fixed size or a percentage of the total taxa.

How to decide where to bisect the tree, (either Centroid Edge or the Longest Branch).



PASTA Algorithm



PASTA - Practical Alignment using SATe and TraAnsitivity

External Tools
Aligner: MAFFT
Merger: MUSCLE
Tree Estimator: FASTTREE
Model: GTR+G20

Job Settings
Job Name: pastajob
Output Dir.
CPU(s) Available: 1
Max. Memory (MB): 1024

Sequences and Tree
Sequence file ...
Multi-Locus Data
Data Type: DNA
Initial Alignment: Use for inital tree
Tree file (optional) ...

PASTA Settings
Max. Subproblem: Percentage 50 / Size 200
Decomposition: Centroid
Time Limit (hr) 24
Iteration Limit 100
Return: Final

Workflow Settings
Algorithm: Two-Phase (not PASTA)
Post-Processing: Extra RAxML Search

Start

PASTA 1.6.4, 2013-2016
Running Log (2016-06-15 22:37:41 CST)
PASTA Ready!

When to Stop Running?

(see below)

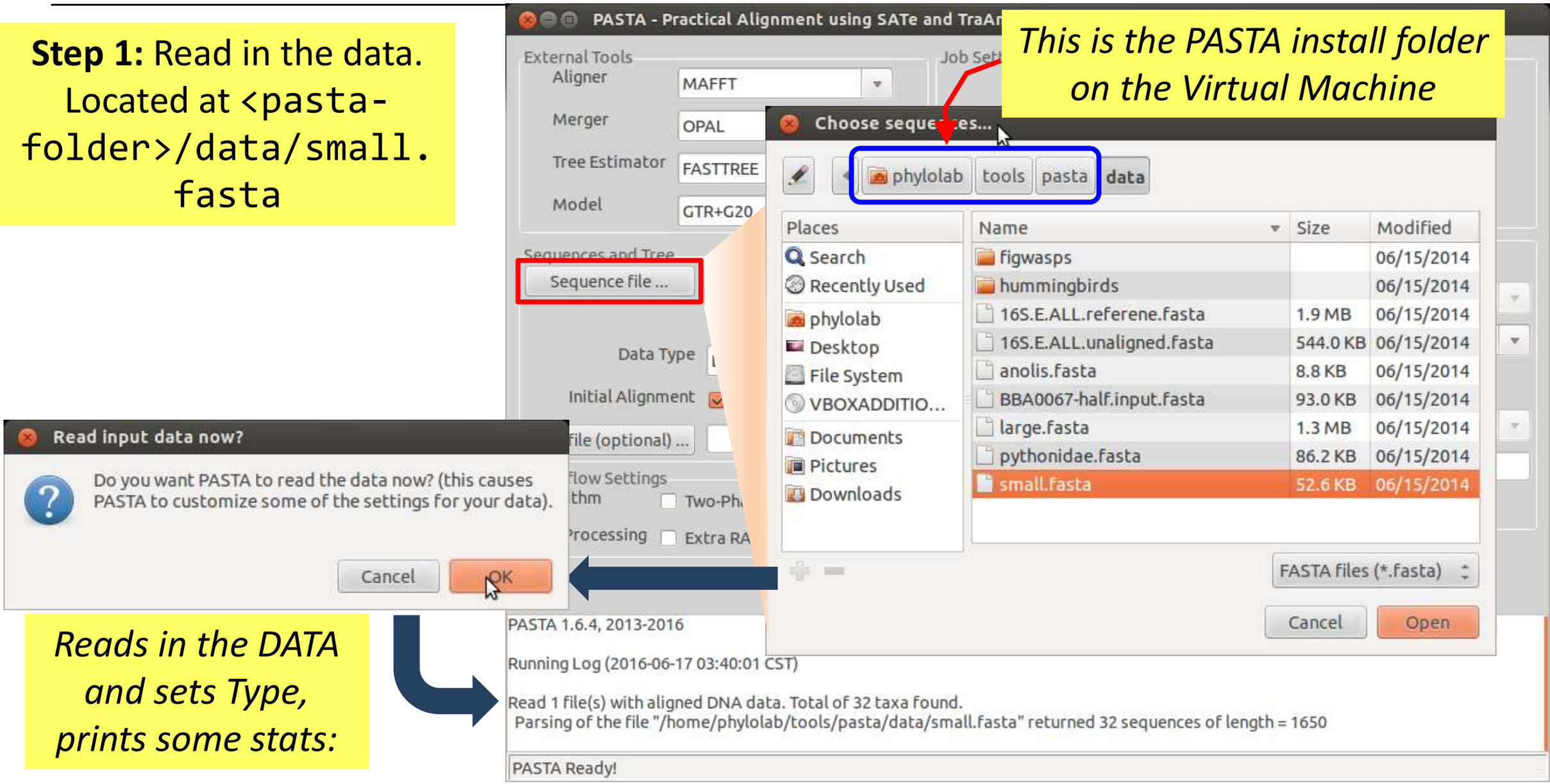
Which iteration to return? (Final or Highest Likelihood)

Should final tree be RAxML?

Two-Phase search is simply 1) run an alignment, 2) get a tree from it. This is completely different than PASTA and *if this is checked, PASTA (formally) will not be run*.

Example 1: small.fasta

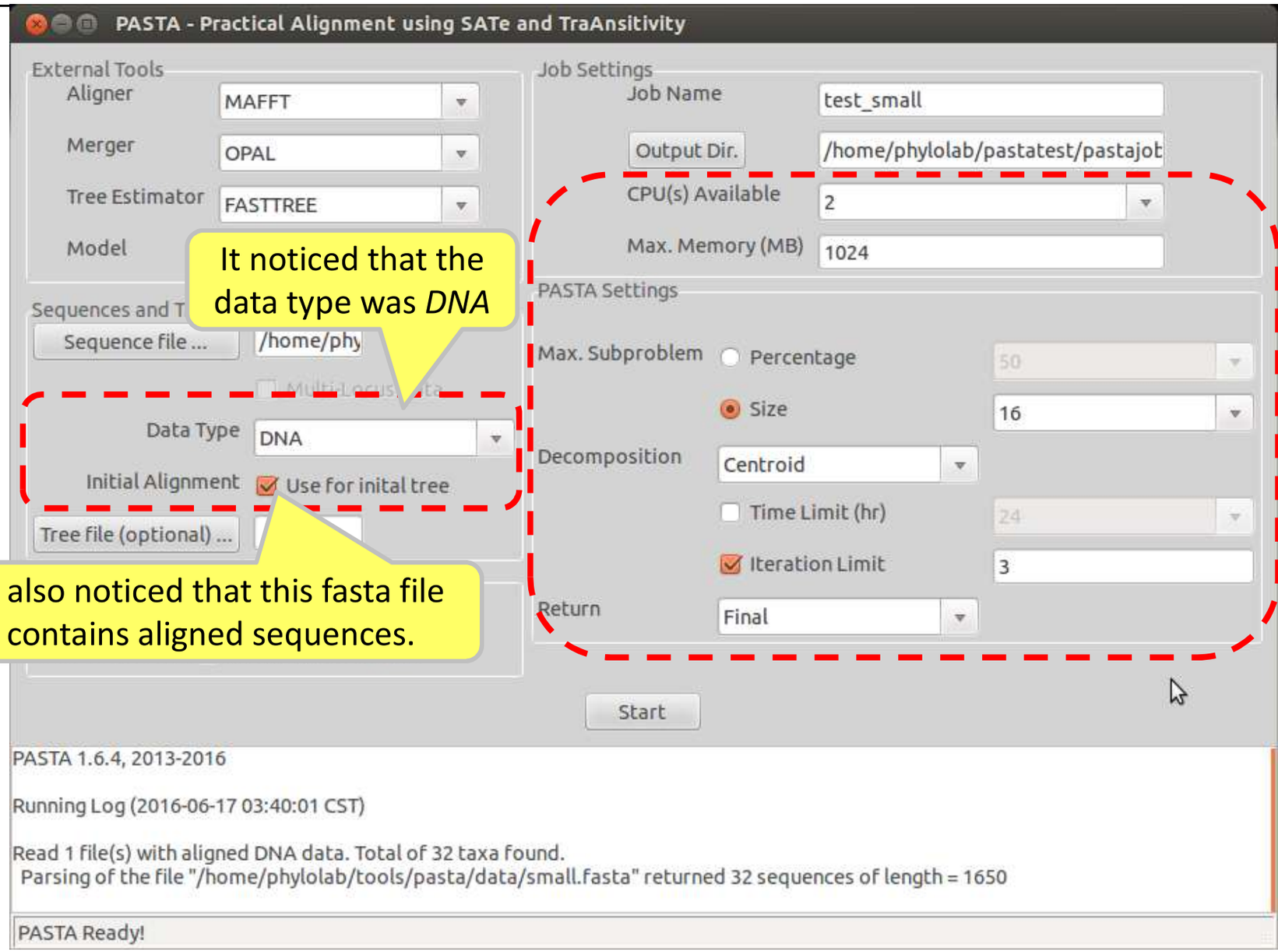
Step 1: Read in the data. Located at `<pasta-folder>/data/small.fasta`



Reads in the DATA and sets Type, prints some stats:

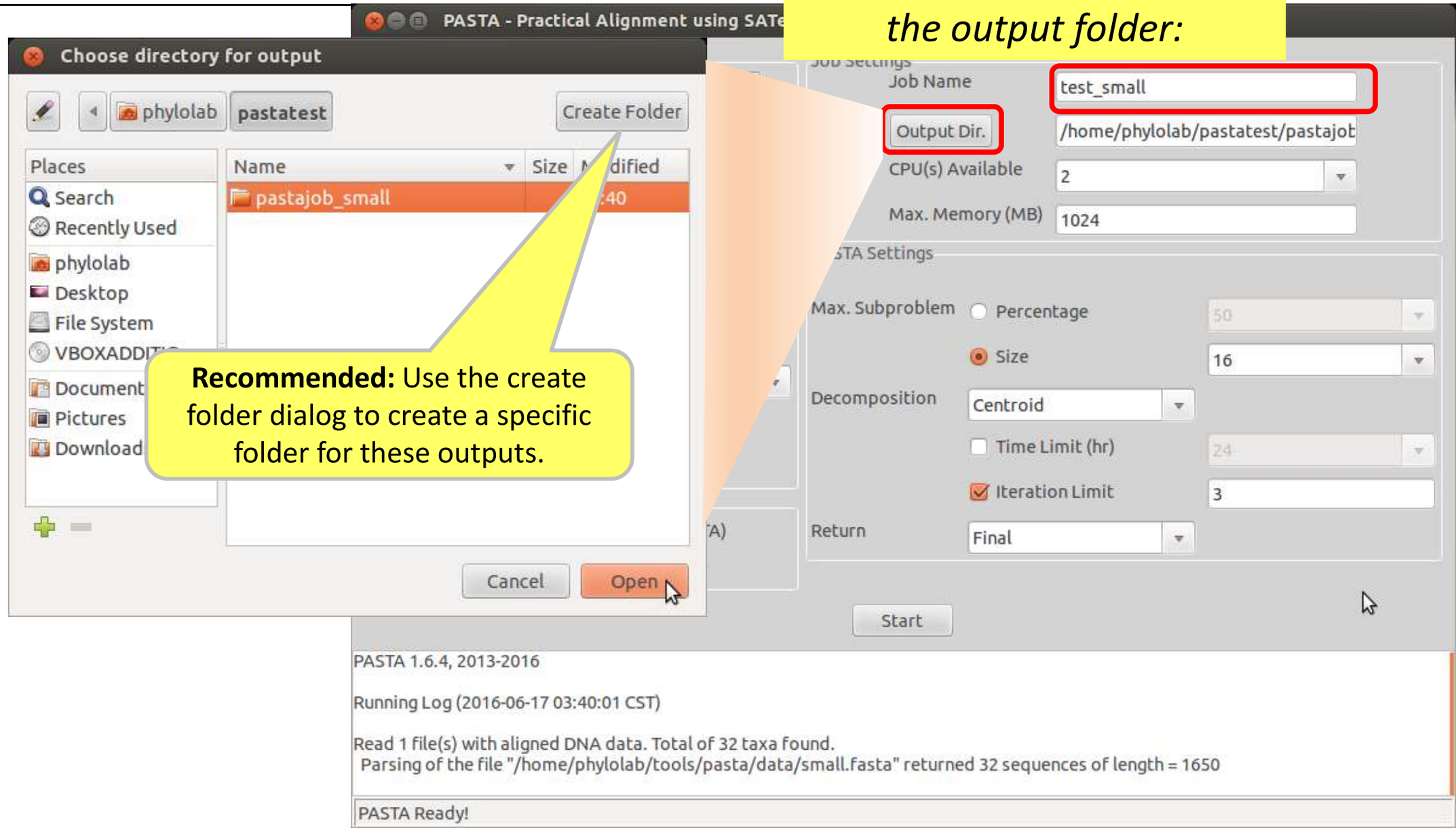
Example 1: small.fasta

Importing the data caused the GUI to automatically set several settings based on the size, data type, etc…



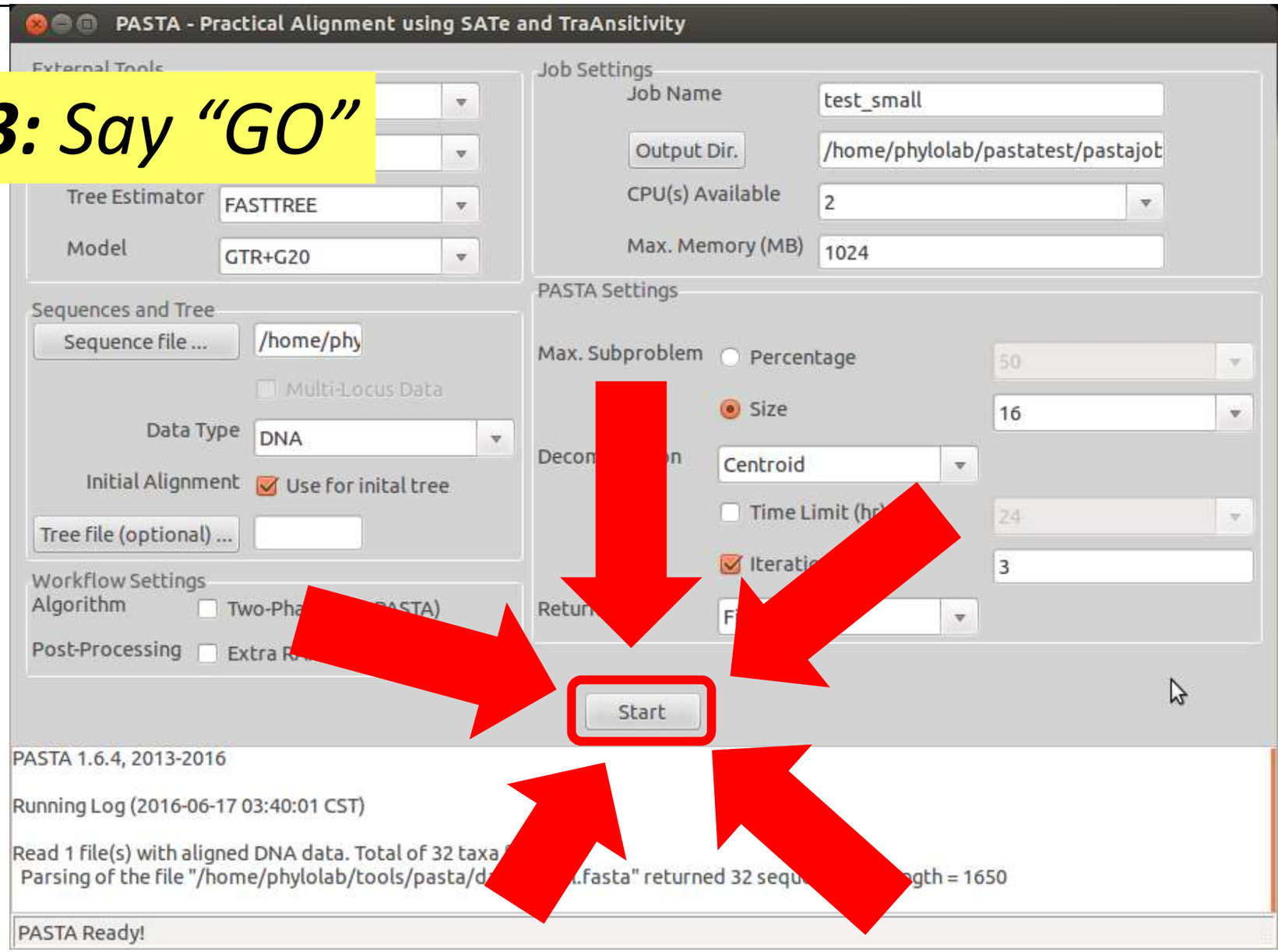
Example 1: small.fasta

***Step 2:** name the job & set the output folder:*



Example 1: small.fasta

Step 3: *Say "GO"*



Example 1: Examining the Output Folder

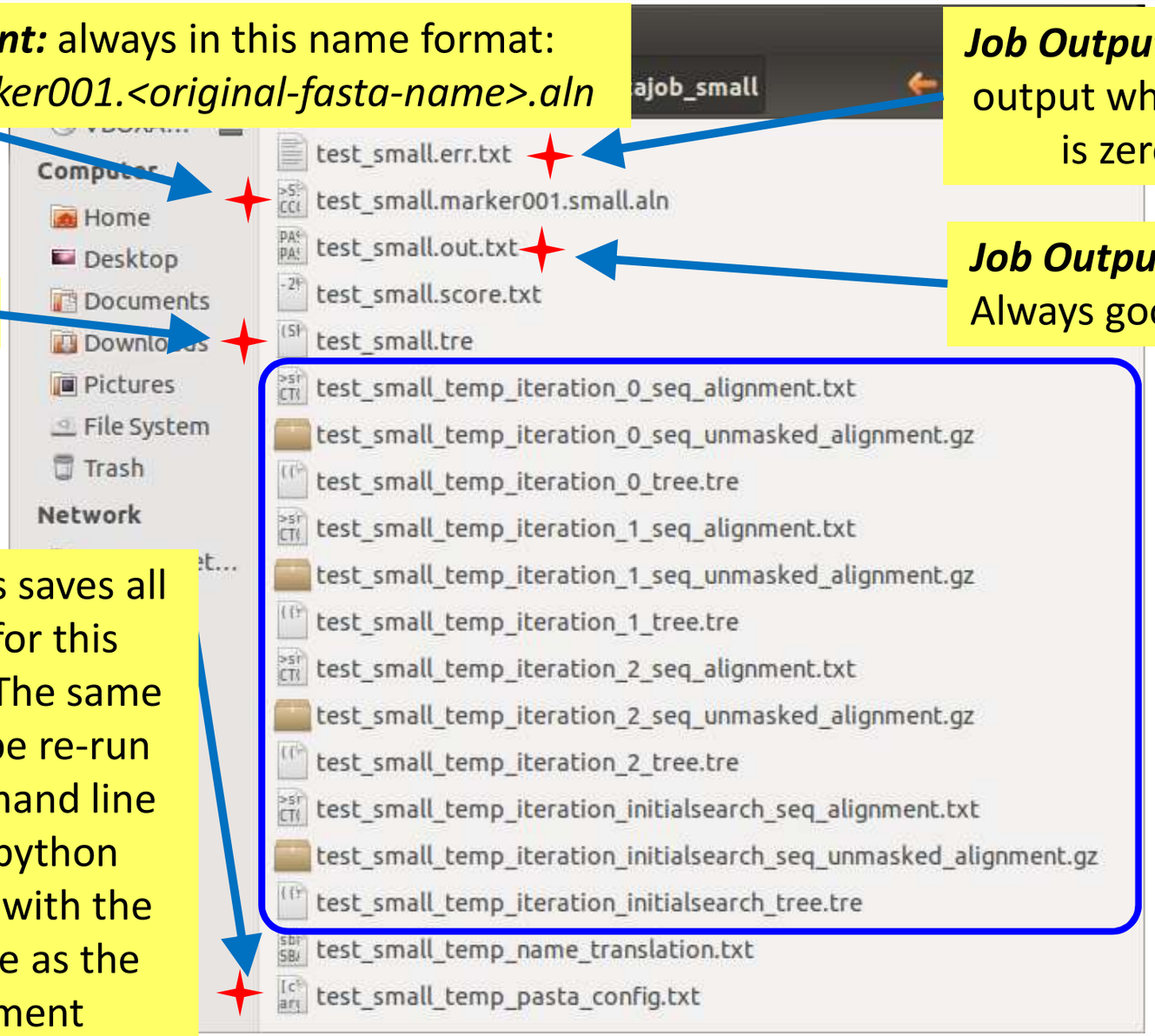
✦ = Important File

Final Alignment: always in this name format: *<jobname>.marker001.<original-fasta-name>.aln*

Job Output (Errors): contains PASTA console output when errors are reported. If this file is zero bytes, that is a good thing.

Job Output: contains PASTA console output. Always good to examine this file after a run.

Final Tree



Config File: This saves all the settings for this particular job. The same exact job can be re-run from the command line by running "python run_pasta.py" with the path to this file as the ONLY argument

Intermediate alignments and trees after the initial search and after each iteration. Useful mainly for diagnostics and debugging

Example 2: BBA0067 (time permitting)

- (protein data)

Final Tips & Best Practices

- After running an alignment, it is always a good idea to look at the console outputs generated to verify that PASTA did what it was expected to do. If the error file is non-zero size, read that too.
- The PASTA default settings are appropriate and well-chosen for most applications. Unless you have a good reason to use something else, this is a good starting point.
- PASTA scales with the number of cores available, so giving it as many processors as possible is a good idea.
- There are more settings available than what is in the GUI. Check the config file output for any pasta job to see the full list. Also can type “python run_pasta.py –h” from the pasta folder to see a thorough help menu
- Approximate running time benchmarks (length=1500 base pairs):
  - 100 Sequences: <10 minutes on a laptop
  - 1000 Sequences: About 1-3 hours on a 16-core server
  - 10000 Sequences: About 8-15 hours on a 16-core server
  - (Should scale about linearly after this, but will depend on settings…)

Resources

- PASTA User Group:

  https://groups.google.com/forum/#!forum/pasta-users

- Link to these slides:

  http://publish.illinois.edu/michaelnute/useful-files/

- Github Repository (which has more documentation, including full install instructions):

  http://github.com/smirarab/pasta

My Email: nute2@Illinois.edu