Protocols for assembly

A Problem

- NGS workshop attendees could run everything we did in the workshop w/o much trouble: canned data, canned analyses, cloud computers.

- Then they'd go home and try to use it on their own data.

- FAIL.

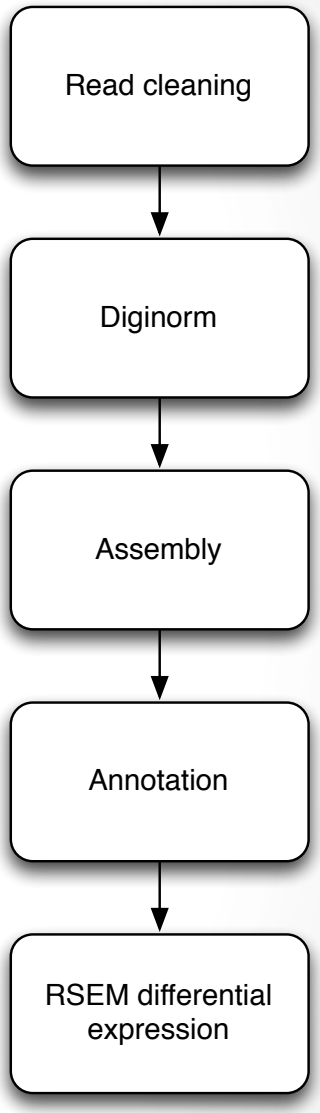
A solution?

- Update tutorials to work with *real* data sets.
- Write full pipelines!

- Pros: lots more people could do their work!
- Cons: an immense amount of effort, even if it all goes right!

khmer-protocols:

- Effort to provide standard “cheap” assembly protocols for the cloud.
- Entirely copy/paste; ~2-6 days from raw reads to assembly, annotations, and differential expression analysis. ~$150 per data set (on Amazon rental computers)
- Open, versioned, forkable, citable.



Features

- Open – licensed so you can do whatever you want with them, including modify & redist.
- Forkable – you can make your own copy quite easily (c.f. github, next week).
- Versioned – you can refer to fixed version numbers.
- Citable – we're building DOIs for them so you can cite the version you used in a paper.

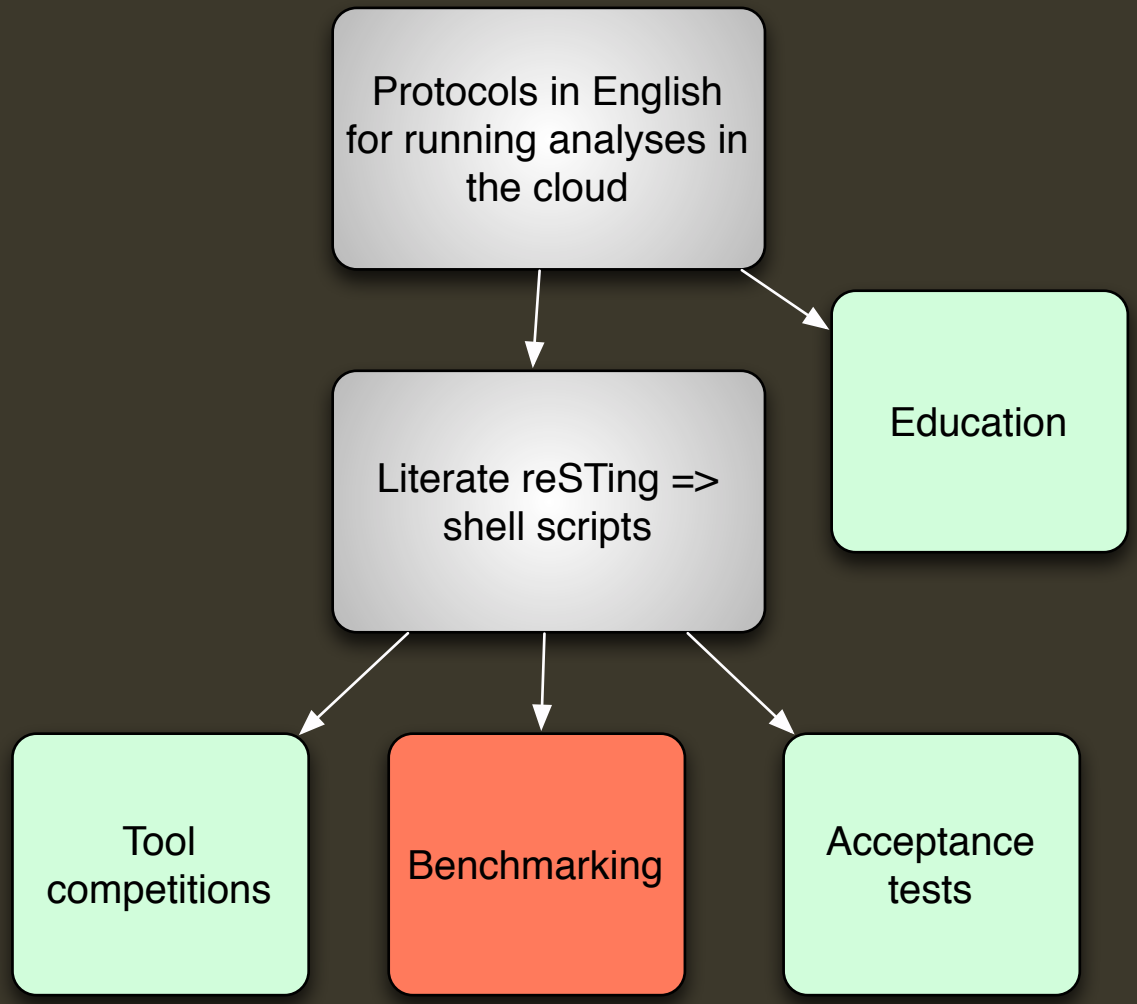
A few thoughts on our approach…

- Explicitly a “protocol” – explicit steps, copy-paste, customizable.

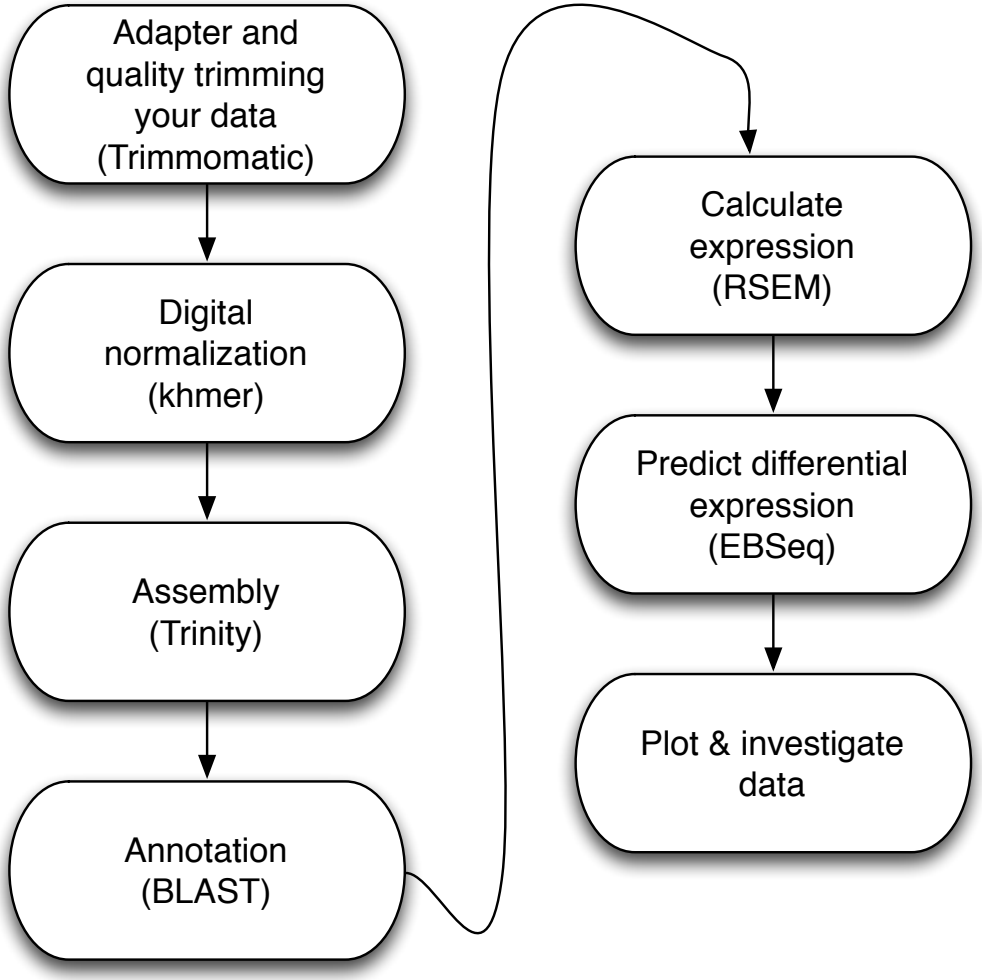
- No requirement for computational expertise or significant computational hardware.
  - ~1-5 days to teach a bench biologist to use.
  - $100-150 of rental compute (“cloud computing”)…
  - …for $1000 data set.

- Adding in quality control and internal validation steps.

Multi-use!


Protocols in English for running analyses in the cloud
Literate reSTing => shell scripts
Education
Tool competitions
Benchmarking
Acceptance tests

The mRNAseq protocol



Today (for a subset)