Workflow tools

Are they useful?

What is a workflow tool?

- support executing a network of operations with interdependencies

Problems with current offerings

- many are too specialized. Not easily reused for other applications

Why are there so many tools?

- sometimes it may seem easier to start from scratch than read other people's undocumented code!
- existing workflow tools are sometimes not flexible enough

Usage statistics

bpipe: 1; snakemake: 1; gib: 1; bcbio: 2

scripting: 9

make: 3; nextflow: 1

cuneiform: 2; luigi: 1

GATK queue/piper: 2

galaxy: 1; Cloudgene: 2

Ideal system: brainstorming

- 2 part problem:

  specifying what you want to have done

  implementing the “engine” that can do it

- “query interface” to workflows
  - define what you want your result to work like
  - “magic” compiler figures out how to do it

Ideal system: brainstorming

- how often does a workflow change?
- what do you use workflows for?
- (at least) 2 uses
  - automatic system is one use
  - significant percentage of work is more exploratory
  - based on intermediate results, change tools, parameters; try many variations of a workflow and compare results

Essential features

- complete reproducibility
- API: important for automation
- running on a cluster should be easy to do
- share standardized workflows with others using the same workflow tool. In its definition, it should include program names, versions, parameters and, if necessary, standard datasets/libraries/references

Essential features

- should coexist with other frameworks/programs/services running on the premises

Nice to have features

- ability to re-use intermediate results
- interact with other systems in infrastructure
  - trigger actions, be triggered…
- ability to prompt user is “odd” cases
- would be nice if it can help you parallelize operations easily/automatically

Are they useful: the consensus is “yes”, to some extent

-
- one approach for “long-term” reproducibility: serialize workflows to shell scripts, with software versions