WORKFLOW AND DATA MANAGEMENT FOR SYSTEMS BIOLOGY RESEARCH

Overview

- Problems in Research Data Analysis
  - *Data Management*
  - *Workflow Management*
- Systems Biology Workflow Management Systems
- Omics Dashboard

Problems: Data Management

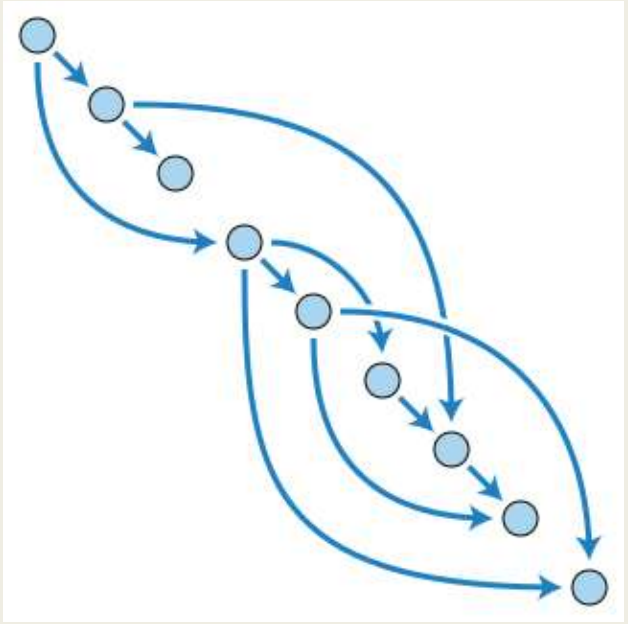
- Instruments produce a large number of file in differing formats
- Laboratory Information Management Systems (LIMS) are designed to manage scientific data and metadata. LIMS packages handle all aspects of the scientific process, from ordering to certification of results.
- LIMS packages designed for clinical and forensics labs are not necessarily the best tools for research environments.
- Researchers develop a number of ways (many of them ad hoc) to manage their data, including databases and datastores.

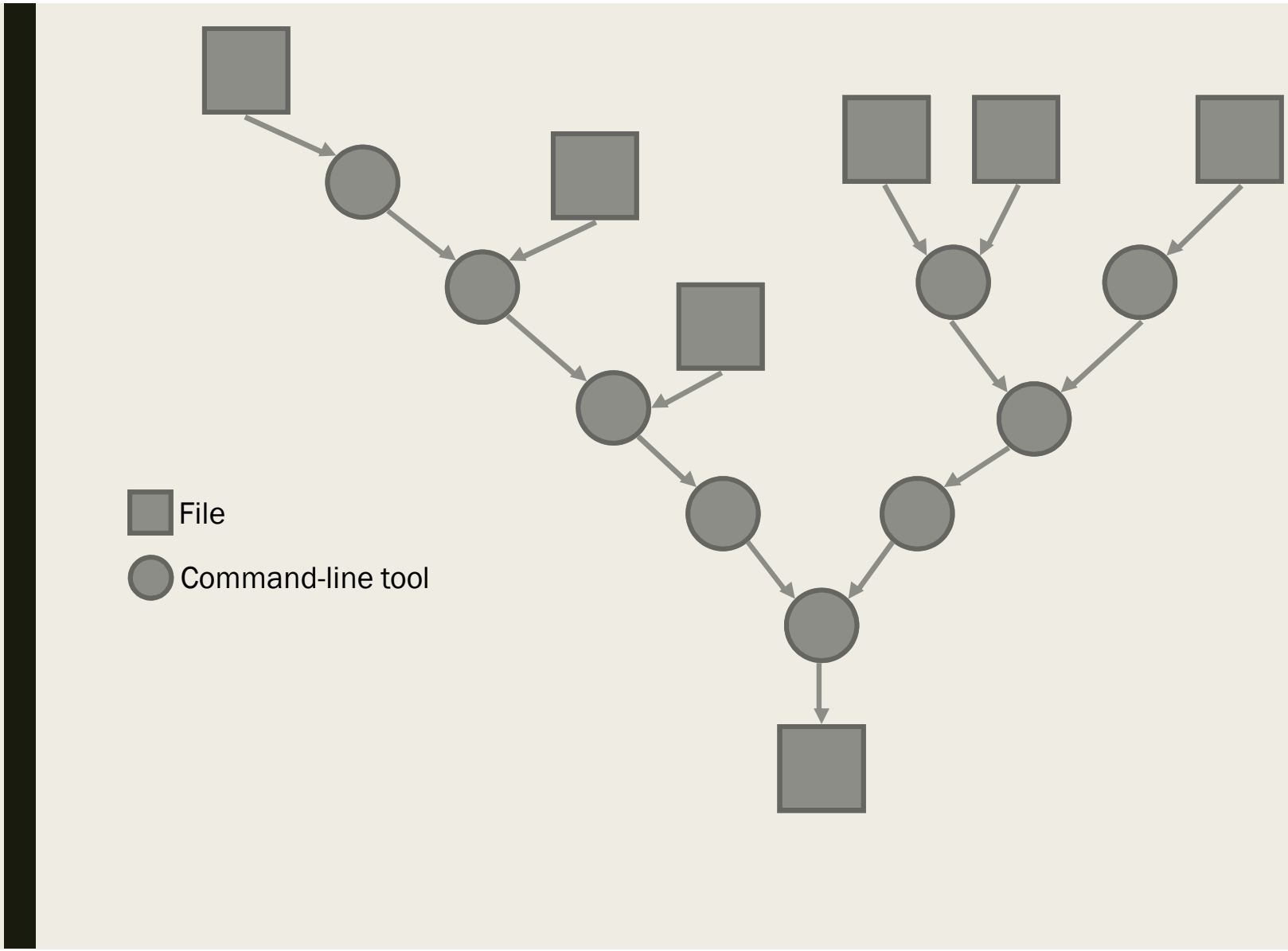
Problems: Workflow Management

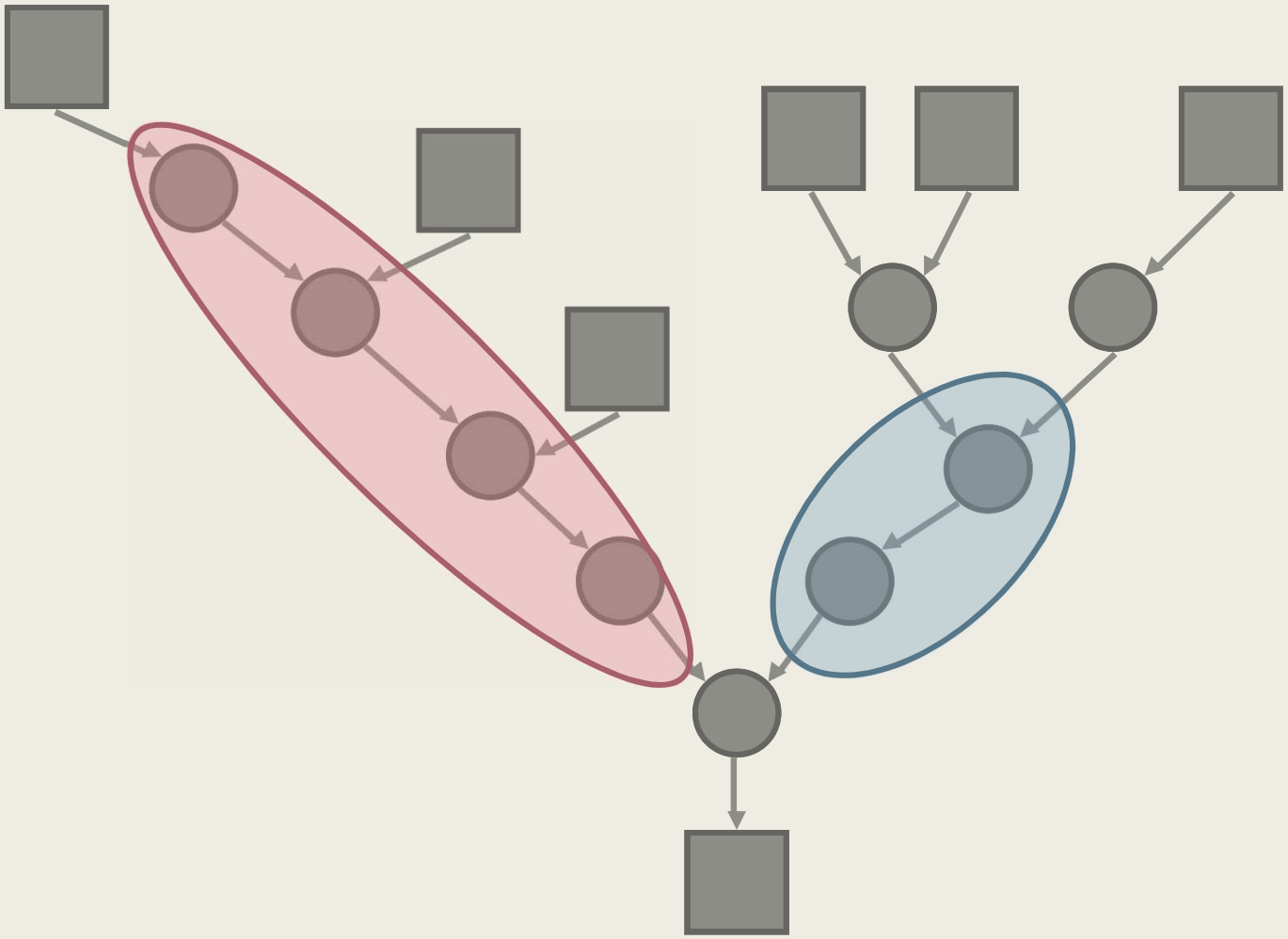
- **Workflow**: a series of repeatable steps applied to input data to obtain desired output data.
- Scripts and makefiles are the traditional ways of representing workflows.
  - *Scripts: define workflow as a sequence of commands (can only run sequentially).*
  - *Makefiles: Steps can be defined as inputs to other steps, allowing parallel execution of independent steps.*

Workflow Management

- Workflows are directed acyclic graphs (DAGs).
- Every directed acyclic graph has a topological ordering (an ordering of nodes such that if there is an edge from *u* to *v*, *u* appears before *v*).
- All nodes with indegree 1 can be collapsed into single nodes (their parents of indegree 1 or indegree 0) representing steps to execute sequentially. These nodes can be executed in parallel.
- Make files and most workflow engines use this method.




File
Command-line tool



Workflow Description Languages

- Makefile syntax is esoteric to non programmers.
- JSON and YAML are flexible languages that are easy for both humans and machines to understand.
- The Common Workflow Language (CWL) and Workflow Definition Language (WDL) are two projects which aim to create standard representations of workflows.

WDL vs. CWL

CWL

- Command line tools are described in their own YAML files, which specify which optional and required parameters a tool will take as an input and which artifacts are produced as outputs.
- A workflow consists of a definition of required inputs, expected outputs, and a list of steps consisting of aliases to command line tool wrappers.
- Workflows can be represented in either JSON or YAML format (YAML is a superset of JSON).

WDL

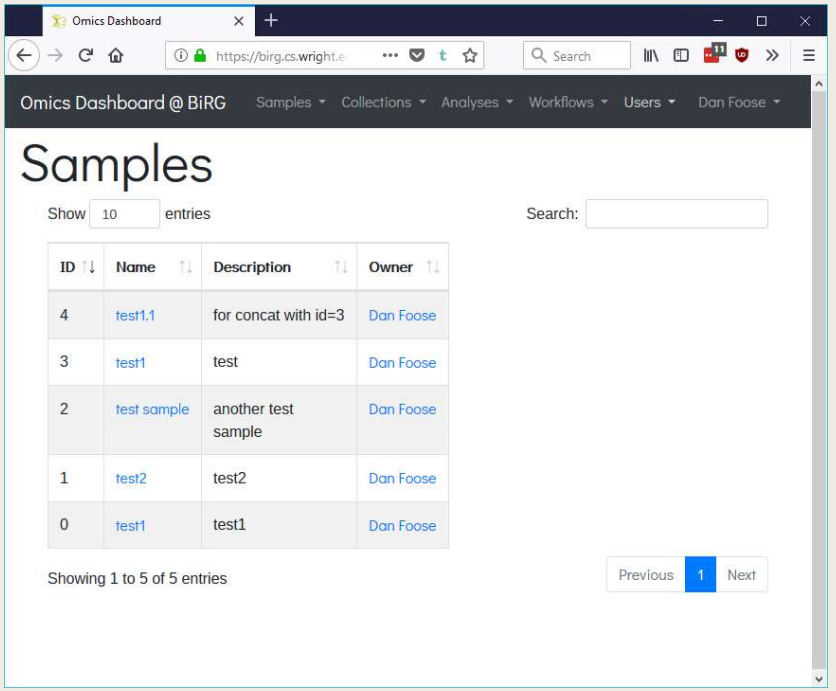
- The commands, inputs and outputs for tools are defined in each task in the workflow.
- Workflow inputs and outputs are defined in the workflow definition, as in CWL.
- Workflows are represented in a domain-specific language (WDL)

Workflow Management Applications

- CWL
  - *Galaxy: a time-sharing service provided by academic supercomputing centers*
  - *Arvados: focused on clusters and cloud infrastructure*
  - *Toil: scale from single machine to large cluster*
  - *Rabix: local machine focus*
  - *Apache Taverna: big data/cloud focus*
- Others (with their own workflow languages)
  - *KNIME*
  - *Orange*

Omics Dashboard

- Replacement for legacy Rails app developed c. 2010.
- Provides basic permissions management for users and user groups.
- Uses HDF5 file format to group related data generated from the same subjects.
- Uses CWL and Docker for reproducible workflows.



Omics Dashboard: Focus

- Designed to be deployed on single machines, usually on bare metal, to handle datasets that easily fit on commodity storage devices,
  - *The first step in the installation instructions is not "sign up for AWS".*
  - *Support for using Kubernets clusters for job execution will eventually become available.*
- Provides an easily-extensible system to handle any kind of data.
- Designed to be easy to administrate.

Omics Dashboard: Implementation

- Consists of two applications implemented in Python 3.6 using the Flask framework, which are served via a proxy server using the uWSGI protocol (in our deployment, we use the NGINX server on birg.cs.wright.edu to mount the apps on /omics and /omics_jobserver).
- The omics application generates and serves webpages, accepts requests via a RESTful API and manages data.
- The omics jobserver application is used to execute arbitrary CWL workflows and manage jobs.
- The two services consume each other's RESTful APIs.
- Both services are deployed in Docker containers.

Omics Dashboard: Data Management

- Sample: a base dataset uploaded directly from a text file.
- Collection: a concatenation of samples, possibly from samples containing different kinds of data
- Analysis: a grouping of collections on which workflows can be executed.
- Workflow: a workflow which may be executed on any set of collections. Inputs may be collections or scalars including paths within collections, strings, and numbers.
- When attached to an analysis, the workflow will produce collections that are attached to the analysis.

Omics Dashboard: Data Management

- Samples and collections are represented by sequentially-named HDF5 files which have flexible schema (only a few attributes are required) and contain all attributes and data objects for the sample or collection.
- Users, user groups and analyses are represented by tables in a SQLite database.
- Workflows are represented as YAML files following the CWL workflow schema.
- Workflow module definitions are represented as YAML files following the CWL command line tool schema.

Omics Dashboard: Workflow Execution

- The omics jobserver service receives a POST request from the omics service containing the workflow and workflow parameters.
- The jobserver then creates a job, adds the jobs to the jobs list, then starts the job.
- When the job is complete, the jobserver sends a POST request to the omics service, which adds the results of the workflow as collections and cleans up the temporary files generated by the jobserver.

Demo