BioLite

A lightweight bioinformatics framework with automated tracking of diagnostics and provenance

Mark Howison[1,2], Nicholas A. Sinnott-Armstrong[2], Casey W. Dunn[2]

[1] Center for Computation and Visualization

[2] Department of Ecology and Evolutionary Biology

Brown University

Presented at USENIX TaPP '12 | http://www.dunnlab.org/biolite

BROWN

The Problem

Next-Gen Sequencing technologies produce big data:

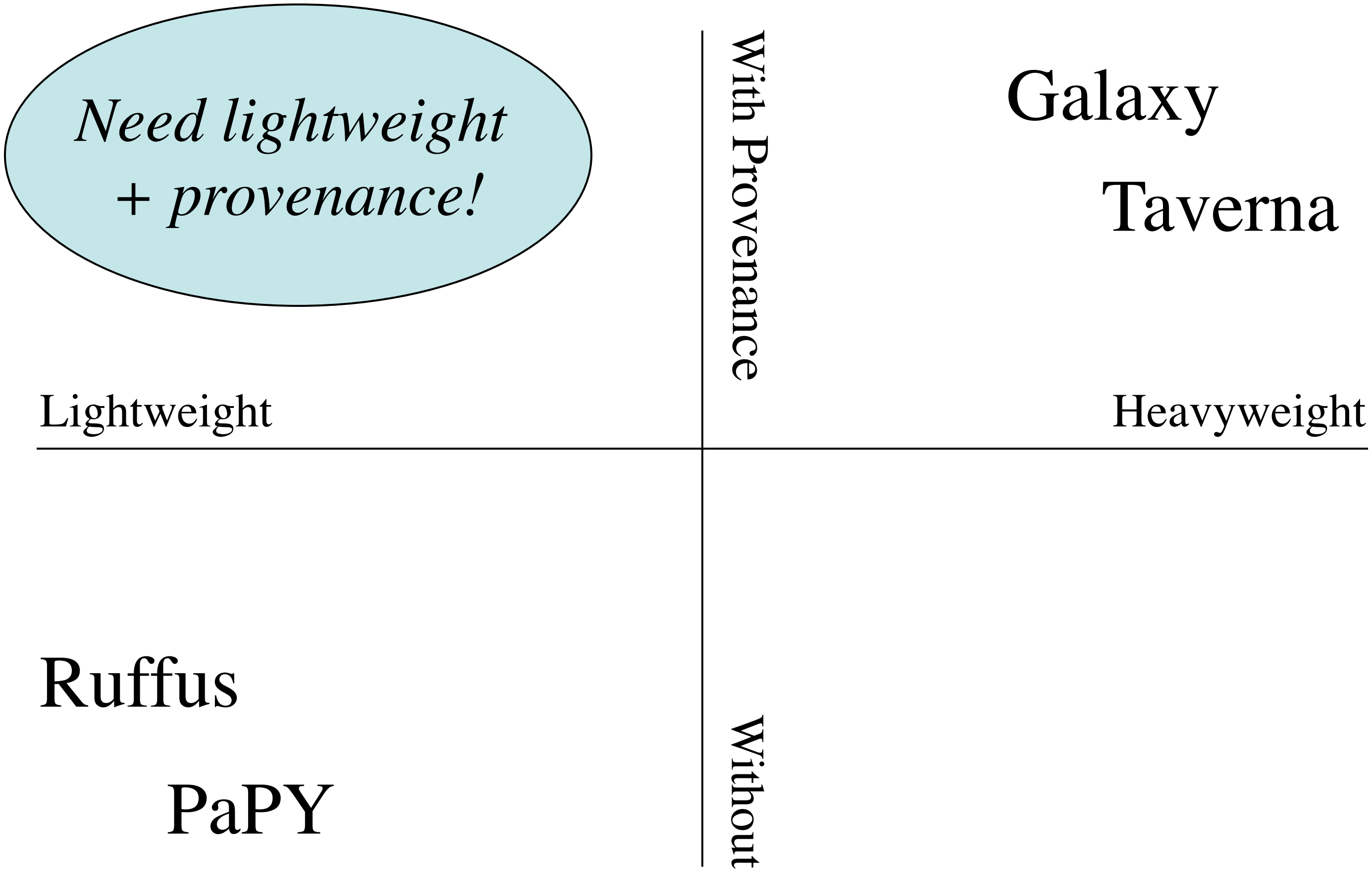
- ~250GB per run for an Illumina HiSeq 2000

The data require complex analyses:

- Quality control and filtering of the raw ‘reads’
- Assembly of short reads into contiguous sequences
- Alignment and comparison to known sequences

Need a better solution than ad hoc analyses and one-time scripts!

Other ‘Workflow’ Solutions



Need lightweight + provenance!

With Provenance

Galaxy

Taverna

Lightweight

Heavyweight

Ruffus

PaPY

Without

Lightweight Design Goals

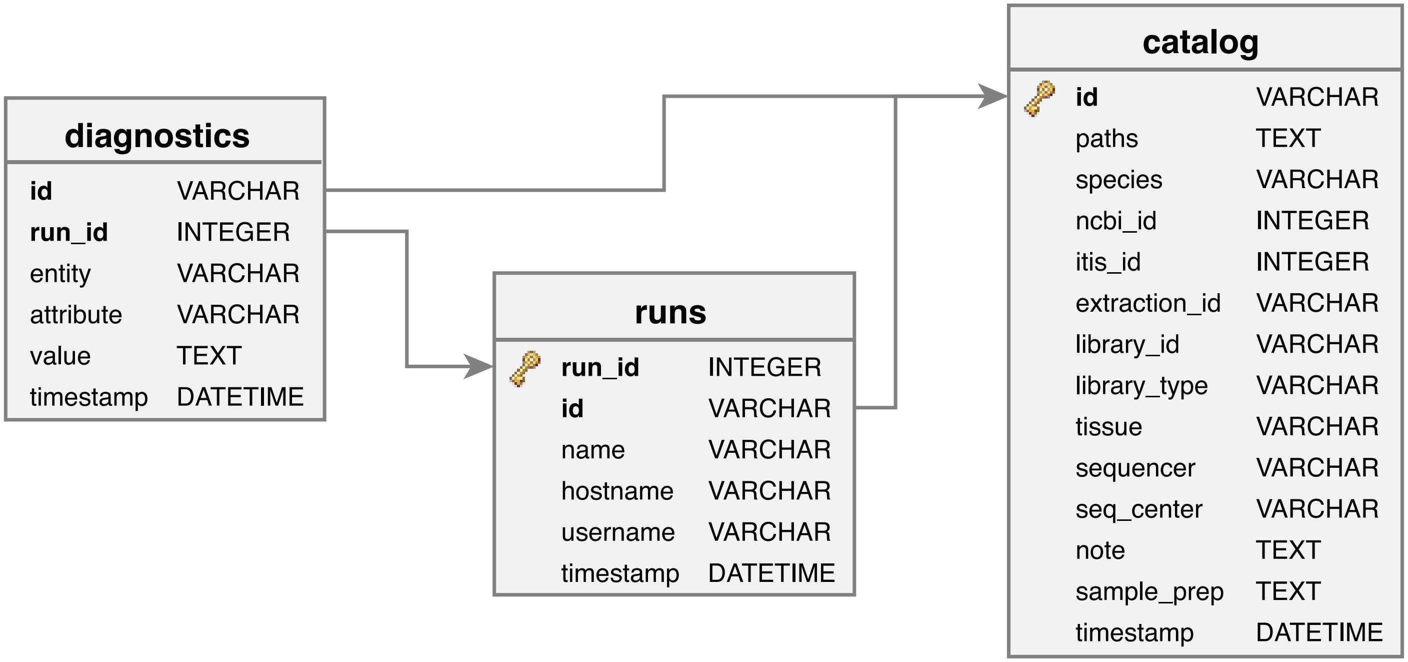
- Command-line usage
- Easily extendable through scripting and programming
- Minimal administrative overhead and dependencies
- Portability and performance

BioLite

A Python framework and set of C++ tools for:

- building out customized analysis **pipelines**
- fault-tolerance, through built-in **checkpointing**
- automating the collection/reporting of **diagnostics**
- tracking the **provenance** of analyses:
  - resource usage
  - paths and parameters
  - program versioning
  - statistics

BioLite's Database



diagnostics	
id	VARCHAR
run_id	INTEGER
entity	VARCHAR
attribute	VARCHAR
value	TEXT
timestamp	DATETIME

runs	
run_id	INTEGER
id	VARCHAR
name	VARCHAR
hostname	VARCHAR
username	VARCHAR
timestamp	DATETIME

catalog	
id	VARCHAR
paths	TEXT
species	VARCHAR
ncbi_id	INTEGER
itis_id	INTEGER
extraction_id	VARCHAR
library_id	VARCHAR
library_type	VARCHAR
tissue	VARCHAR
sequencer	VARCHAR
seq_center	VARCHAR
note	TEXT
sample_prep	TEXT
timestamp	DATETIME

The *diagnostics* table has a complete non-executable history of the analysis:

diagnostics +
pipeline script = *reproducibility*

Table 1: Storage requirements for 168 runs

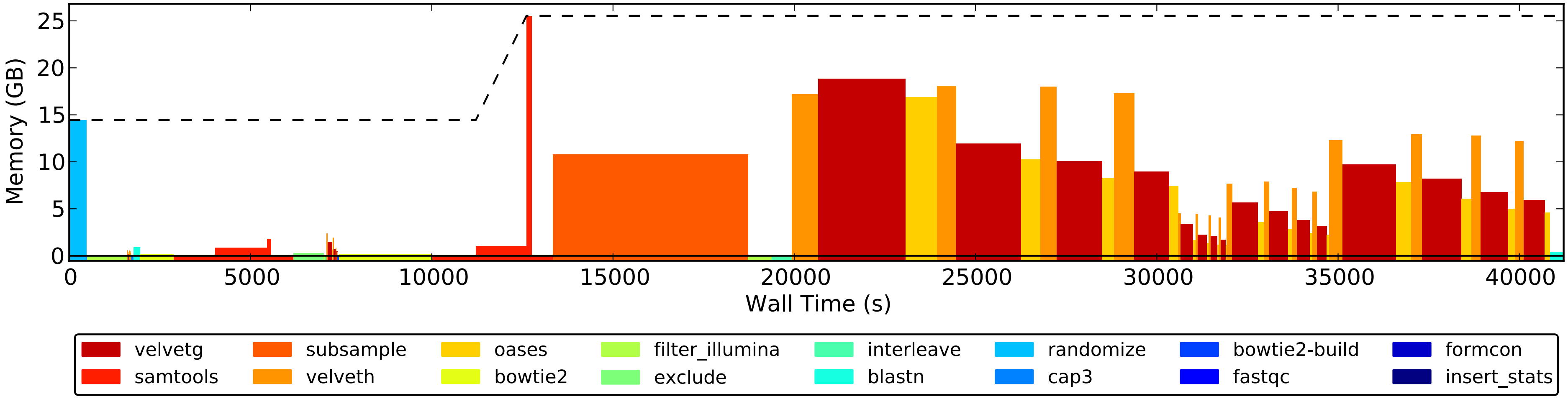
Data	**GB**
raw data sets	192.4
intermediate results (permanent)	1,241.6
intermediate results (scratch)	1,057.1
diagnostics: SQLite and text files	0.073

Reports

- API for accessing raw diagnostics, generating custom reports
- Reporting code is integrated with analysis scripts
- Tabular reports show comparisons across data sets

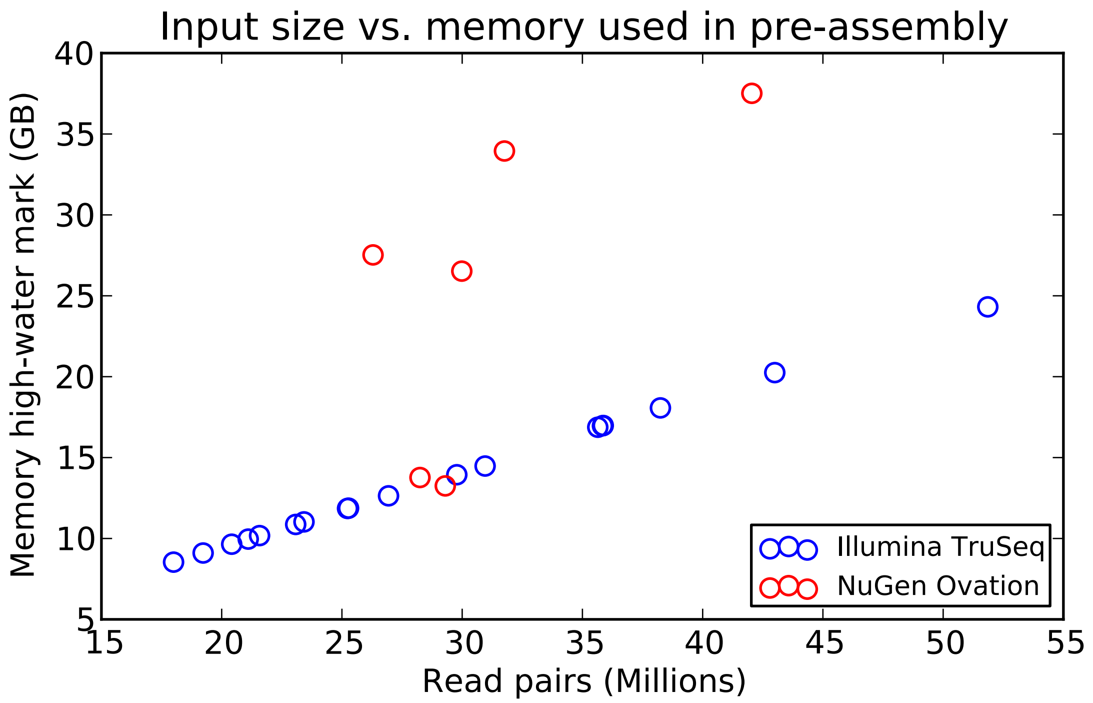
Resource Profiling

For understanding how resources are used by different stages of a pipeline run



Resource Profiling

For comparing computational requirements across analyses, forecast future requirements

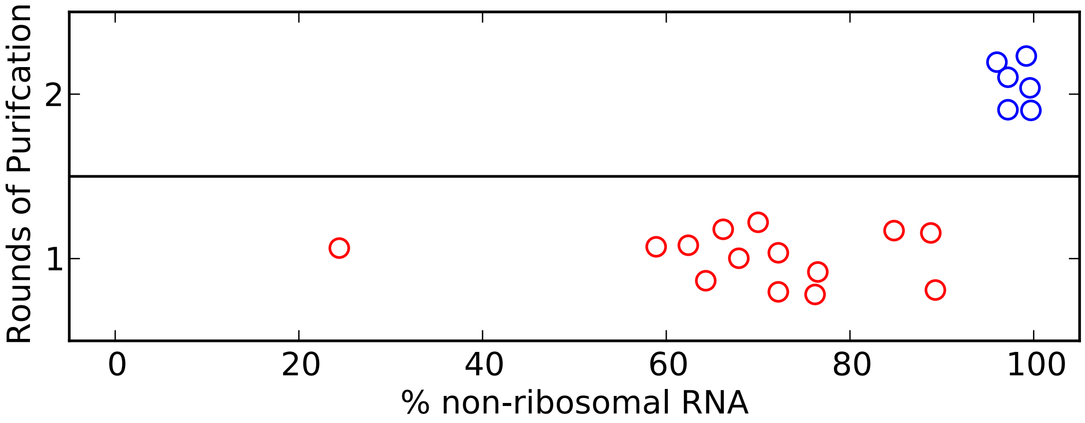


BROWN

Diagnostics

For answering questions about up-stream data collection, comparisons across analyses, etc.

Example application: how does purification method affect usable RNA content?



Applications of BioLite

- Development is driven primarily by Agalma, a *de novo* transcriptome pipeline
- Chlorox, a chloroplast genome assembly tool
- Other tools in progress at Brown…

Questions?