Toil

Robust Pipeline Architecture for Genomic Workflows

John Vivian, Arjun Rao, Frank Nothaft, CJ Ketchum, Jake Narkizian, Jacob Pfeil,
Hannes Schmidt, David Haussler, Benedict Paten

UNIVERSITY OF CALIFORNIA SANTA CRUZ Genomics Institute

TOIL
Massively Parallel Workflows
(and fire-breathing dragon slugs)

What is Toil?

Toil is a massively scalable pipeline management system that is fault-tolerant, portable, simple, and open-source. Workflows can be easily developed on a laptop, then deployed to both cloud environments and standard HPC clusters.

Toil workflows (and Toil's source code) are written entirely in Python, giving users the power of a Turing-complete language with no DSL to learn.

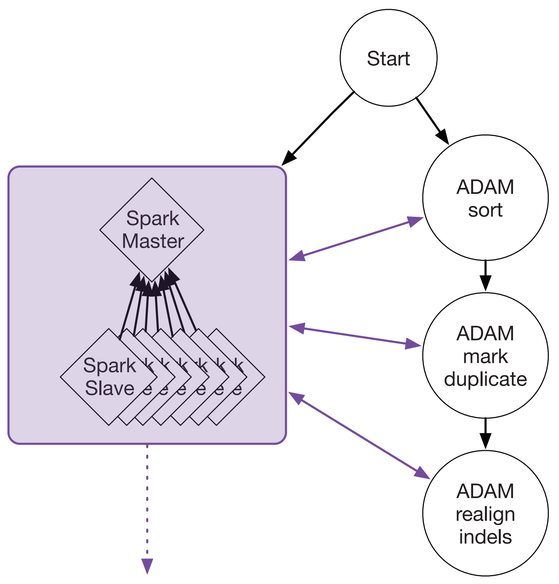
```
from toil.job import Job

def HelloWorld(message, cores=2, memory="2G", disk="3G")
      return "Hello world!, here's your message: %s" % message

j = Job.wrapFn(HelloWorld, "woot")

if __name__ == "__main__":
      parser = Job.Runner.getDefaultArgumentParser()
      options = parser.parse_args()
      print Job.Runner.startToil(j, options)
```

SPARK Support



Toil supports SPARK applications through the use of service jobs — which are long running jobs that interface with other jobs. ADAM, the Berkeley Genomics Engine, uses Toil to write SPARK-based genomic workflows.

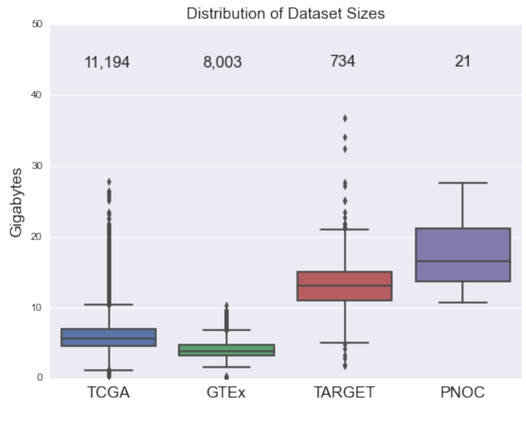
CWL Support

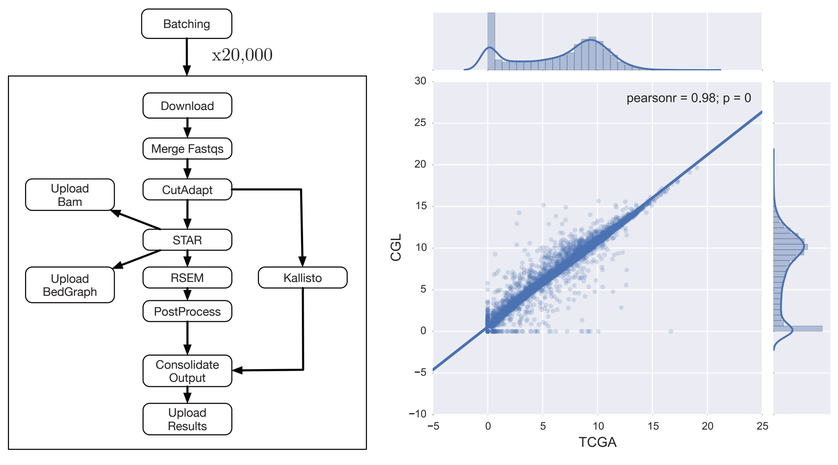
Workflows written in Common Workflow Language, a burgeoning standard for writing scientific workflows, can be run using Toil's engine with no change to the CWL source code or sample configuration file.

COMMON WORKFLOW LANGUAGE

GA4GH-BD2K Toil RNA-seq Recompute

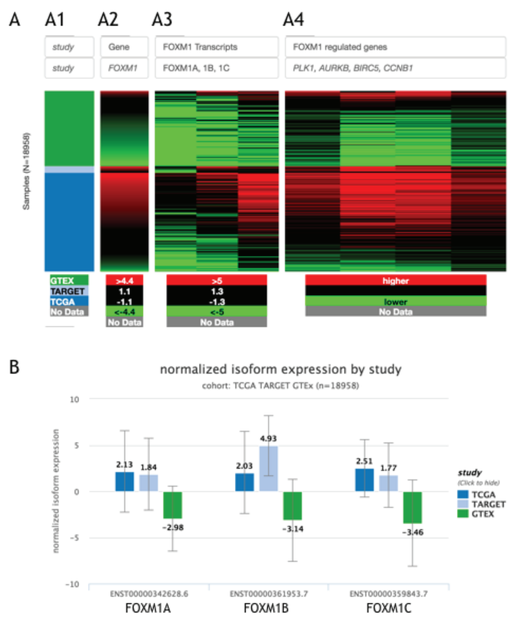
The goal of the Toil recompute was to process ~20,000 RNA-seq samples to create a consistent meta-analysis of four datasets free of computational batch effects. Using Toil, this recompute was run in a little over 3 days on a large AWS cluster for a cost of $1.30 per sample.





(Left) A dependency graph of the Computational Genomics Lab (CGL) RNA-seq pipeline. (Right) A scatter plot showing the Pearson correlation between the results of the TCGA best-practices pipeline and the CGL pipeline. 10,000 randomly selected sample/gene pairs were subset from the entire TCGA cohort and the normalized counts were plot against each other. The unit for counts is: log2(norm counts+1).

Xena Integration



An example of FOXM1 analyzed in the UCSC Xena Browser.

Check us out on Github! www.github.com/BD2KGenomics/Toil

Sign up for the Toil / Xena Webinar
Wednesday, July 20th at 11am
http://tinyurl.com/toilwebinar

Xena Public
Xena Private