Getting started with WDL & Cromwell

Bioinformatics workflows at any scale

Ruchi Munshi
Data Sciences Platform
Broad Institute
The backdrop: data generation set to explode

Quarterly output (in TBases) of the Genomics Platform

Story begins here
Meet Cromwell & WDL

• Execution engine that can
  – Run on any platform (HPC and on Cloud)
  – Seamlessly scale based on workflow needs
  – Provide maximal flexibility for all use cases
  – https://github.com/broadinstitute/cromwell

• Workflow language that humans can read/write
  – Methods developers and biomedical scientists at large
  – https://github.com/openwdl/wdl/
Use containers for portability & reproducibility

A container encapsulates all the software dependencies associated with running a program.

Takes the guesswork out of running workflows on different platforms!
Use a workflow execution engine that runs anywhere

Cromwell

Local  HPC  AWS  Google  Alicloud  TES

Funnel

https://github.com/broadinstitute/cromwell
Run using HPC **and** Cloud resources!

- HPC
- Local
- Data buckets
- Persistent Cromwell server
- Compute environment
Two main ways to run Cromwell

**Command Line**
- Simple, self-contained command
- Appropriate for independent analysts
- Call Caching

**Server**
- API endpoints
- More scalable, appropriate for production environments
- Call caching

```
java -jar cromwell.jar \
  run hello.wdl \
  hello_inputs.json
```
Managing data

- Root directory
- Data handling strategies
- Support for object stores

Managing resources

- CPU
- Memory
- Custom attributes

Run command
- Built-in variables
- Full flexibility

```ruby
backend.providers.SGE.config {
  check-alive = "qstat -j ${job_id}"
}

backend.providers.SGE.config {
  kill = "qdel ${job_id}";
}
```

```
backend.providers.SGE.config {  
  submit = ""
  qsub \n  -terse \n  -V \n  -b y \n  -N ${job_name} \n  -wd ${cwd} \n  -o ${out}.qsub \n  -e ${err}.qsub \n  -pe smp ${cpu} \n  ${"l mem_free=" + memory_gb + "g"} \n  ${"-q " + sge_queue} \n  ${"-P " + sge_project} \n  /usr/bin/env bash ${script}  
}"
```

Plenty of workflow solutions to go around

![XKCD Comic](https://xkcd.com/927/)

So of course we decided to create a new one.
Workflow description Language (WDL)

```wdl
workflow myWorkflowName {

  File my_ref
  File my_input
  String name

  call task_A {
    input: ref=my_ref, in=my_input, id=name
  }

  call task_B {
    input: ref=my_ref, in=task_A.out
  }
}

task task_A {

}

task task_B {

}

task task_A {

}

task task_B {

}

task task_A {

}

task task_B {

}

task task_A {

}

task task_B {

}

File ref
File in
String id

command {
  do_stuff -R ${ref} -I ${in} -O ${id}.ext
}

runtime {
  docker: "my_project/do_stuff:1.2.0"
}

output {
  File out="${id}.ext"
}
```
WDL runtime parameters

task echoHelloWorld {
    command {
        echo 'Hello, World!' 
    }
    runtime {
        docker: "phusion/baseimage"
        disks: "local-disk 10 HDD"
        memory: "1 GB"
        preemptible: 3
    }
}

workflow printHelloAndGoodbye {
    call echoHelloWorld
}

containers
resourcing
cost savings!
Basic WDL plumbing options

**LINEAR CHAINING**

```
call stepA
-Requested stepA.
call stepB { input: in=stepA.out }
call stepC { input: in=stepB.out }
```

**MULTI-IN/OUT**

```
call stepC { input: in1=stepB.out1, in2=stepB.out2 }
```

**SCATTER-GATHER**

```
Array[File] inputFiles
 scatter(oneFile in inputFiles) {
    call stepA { input: in=oneFile }
 }
call stepB { input: files=stepA.out }
```
But what about CWL?

Thanks to our **Workflow Object Model (WOM)**, Cromwell now supports multiple versions of WDL as well as CWL 1.0!

Randall Munroe, XKCD
https://www.xkcd.com/1739/
Cromwell has been busy

Cromwell in production at Broad:

Processed **47.5 million jobs**
over the last two years

And this is just the tip of the iceberg!
Want to discuss further?

My Email:
rmunshi@broadinstitute.org

More Information:
Docs: http://cromwell.readthedocs.io/en/develop/
Github: https://www.github.com/broadinstitute/cromwell
WDL: http://www.openwdl.org
Example Pipelines: https://github.com/gatk-workflows