WDL 书写规范（草稿）

华大基因大数据中心

2018 年 11 月 29 日

目录

1 目的

本文档适用于华大基因大数据中心的自动化平台，用于规范前后端的流程编辑，使流程更加简洁、稳定，便于维护和复用。

2 说明

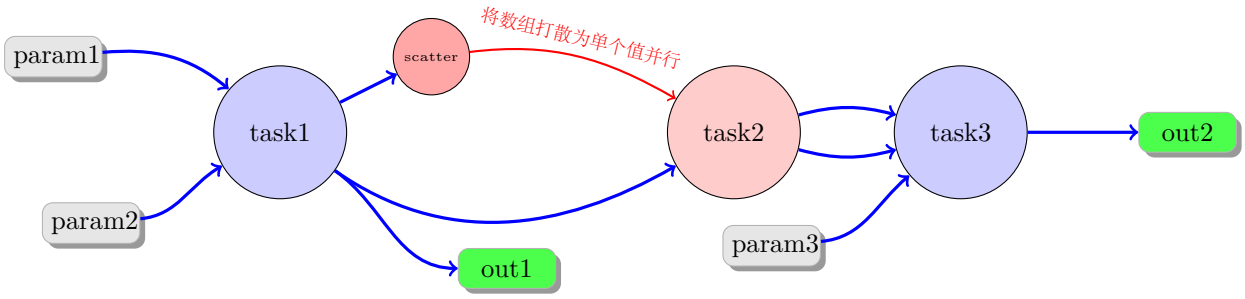
2.1 自动化平台

自动化平台使用 Cromwell + WDL 作为自动化流程框架。

WDL(Workflow Definition Language) 是一种流程描述语言，包括 workflow 和 task 两部分，如下图。

Cromwell 是 WDL 流程的执行引擎（Execution Service）。

2.2 流程示意图



3 原则

- WDL 只关注计算过程
- 任务的调度（包括重跑、投递队列等）全部交给 cromwell 及自动化平台
- 便于测试和维护
- 可复用

4 基本约定

1. 使用 WDL 1.0
   - 点击查看官方语法说明
   - WDL 文件头部需要添加 version 1.0
2. 使用空格缩进，禁止使用 tab
3. workflow 和 task 需分文件编写

5 workflow 书写规范和相关约定

完整 workflow 示例见附录。

5.1 定义 input 语块

5.1.1 样本信息格式约定

用户以约定格式在前端提交数据信息。约定如下：

- 提交**TSV 格式**文件
- 文件**第 1 列**必须为 SampleID，其余各列可根据 workflow 需求自定义。

待讨论：用户提交的 TSV 格式约定...

- 考虑项目名的传递方式
- 目前平台会解析 fq 文件名，将 tsv 中信息转换为 list + map，出现解析错误时需要由用户手动更改数据信息和 fq 文件名
- 建议平台处理样本信息不再依赖文件名
- 平台可检查 TSV 格式是否合法，数据信息全部传给 workflow 处理

示例

```
#SAMPLE_ID FLOWCELL LANE LIB READ1 READ2
SAMPLE_A H814YADXX 1 X /path/to/A_L001_R1.fastq.gz /path/to/A_L001_R2.fastq.gz
SAMPLE_A H814YADXX 2 X /path/to/A_L002_R1.fastq.gz /path/to/A_L002_R2.fastq.gz
SAMPLE_B H814YADXX 1 Y /path/to/B_L001_R1.fastq.gz /path/to/B_L001_R2.fastq.gz
SAMPLE_B H814YADXX 2 Y /path/to/B_L002_R1.fastq.gz /path/to/B_L002_R2.fastq.gz
```

5.1.2 input 语块

workflow 中必须定义 input 语块，用于接受来自系统的样本信息。其中需要声明两个约定参数，格式如下：

```
workflow foo {
  input {
```

```
    Array[Array[String]] DataInfo
    String SampleID
  }
}
```

DataInfo 数据信息

- 若用户提交的 TSV 文件包含多个样本数据，将被按样本拆分。
- 单个样本的数据信息将在 workflow 中被映射成二维数组，即参数 DataInfo。

SampleID 样本名 or 数据 ID

- 自动化平台从 TSV 文件第 1 列提取 SampleID。

5.2 定义中间变量

- 在 input 语块和 call 语块之间，定义中间变量。
- 该类变量可以使用 input 内参数进行赋值。
- 必须赋值。

workflow 除数据信息外，不再接收其他 input。程序及 reference 路径等配置信息，在此处定义。复用 workflow 时，若无 task 级别的修改，只需更改此处变量。（前端的流程配置页面应与流程编辑页面合并。）

5.3 定义 output 语块

workflow 中必须定义 output 语块，用于指定流程产生的结果文件。不在 output 中定义的结果，不会被交付，将被作为临时中间数据处理。

5.4 定义 parameter_meta 语块

应支持在前端查看此信息，方便非流程创建者使用某流程投递任务。

```
workflow foo {
  parameter_meta {
    SampleID : { description: "sample name" },
    DataInfo : {
      description: "data info: SAMPLE_ID\tFLOWCELL\tLANE\tLIB\tREAD1\tREAD2"
    }
  }
}
```

5.5 定义 meta 语块

流程失败需要通知流程维护者。

```
meta {
  maintainer: "Joe Somebody"
  email: "joe@company.org"
  description : "WGS分析流程，reference版本：hg38_decoy"
}
```

5.6 使用 import 导入 task

禁止在 workflow 文件中定义 task。

为便于维护和测试，task 必须是单独的 WDL 文件。然后在 workflow 中 import task，如下图：

```
version 1.0

import "task1.wdl"

workflow foo {}
```

6 task 书写规范和相关约定

6.1 定义 input 语块

6.1.1 参数类型约定

input 语块中的 File 类型的文件，cromwell 会将其软连接至 task 的执行目录。基于这一特性，我们做以下约定。

1. 声明计算数据文件，必须用 File 类型声明
   - 来自 DataInfo 的输入文件（如 fq.gz）
   - 来自其他 task 的中间计算数据(某些文件的索引也需同时声明，如 bam、bai 和 vcf.gz、vcf.gz.tbi 等)

     ```
     input {
       File input_bam
       File? input_bam_index
     }
     ```

   原因

   可帮助检查输入文件是否可访问

   用于 cromwell 判断此数据是否已跑过

2. 声明非计算数据文件和程序时，可用 String 类型替代 File 类型 **(只限 task input 语块)**

   原因

   像 reference.fa、dbsnp.vcf.gz 等有索引的**资源文件**，若将其索引也同样用 File 声明，较为繁琐。[1]

[1] 目前只有 Google 的 Pipelines API (GCE) backends 优化了该问题

6.1.2 参数名约定（可选）

为提高 task 的可复用性，对于支持多线程的程序，可将其线程数和内存额提取为参数。示例：

```
input {
  Int cpu = 5
  Int mem = 10
}
```

mem 单位为 GB 的内存值，用于定义该 task 需申请的内存额

cpu 线程（thread）个数

6.2 command 语块规范

1. command 中的输入输出，禁止使用绝对路径。

   - 结果应保存在当前目录
   - 若程序要求必须输入绝对路径，应使用 $PWD

   ```
   command {
     ${cnvnator} -root $PWD/out.root \
       -chrom ${sep=' ' chr_name} \
       -tree ${input_bam} \
       -unique
   }
   ```

2. 若执行失败，必须返回非 0 值。

   - 若使用了管道，需添加 `set -o pipefail` ，返回最后一个非 0 值
     示例：

     ```
     command {
       set -o pipefail
       ${bwa} mem ...| ${samtools} view -Sb -o ${SM}.${ID}.bam -
     }
     ```

   - 保持只有一条命令。若包含多条命令，需添加 `set -e` ，遇到错误则退出。
     示例：

     ```
     command {
       set -e
       ${statQualDistFq} ${fq1} >${fq1name}.fqStat.txt
       ${statQualDistFq} ${fq2} >${fq2name}.fqStat.txt
     }
     ```

     例外：有些命令执行失败属正常情况，则不能添加 `set -e` 。

     ```
     command {
       hdfs dfs -rm -r -skipTrash /path...
       other commands...
     ```

```
}
```

- 若程序执行状态无法通过 0 和非 0 判断，则需用额外的脚本判断程序 log 或结果，然后返回 0 或非 0。

```
command {
  program... | tee log
  check.py log
}
```

6.3 定义 runtime 语块

- 必须设置 backend
- 必须指定 task 需要申请的资源额度
- 若声明了 cpu 和 mem，应使用变量定义

```
runtime {
  backend: "SGE"
  cpu:cpu
  memory:"${mem} GB"
}
```

- 若未声明此类参数，可使用常量

```
runtime {
  backend: "SGE"
  cpu:1
  memory:"100 MB"
}
```

6.4 定义 output 语块

6.4.1 task 输出约定

- task 的输出必须在 output 中定义

？ 在 task 执行目录下，创建相对目录，适配入仓目录结构

示例 在 command 语块中添加 `mkdir -p result_alignment`
call-task-path/execution/output.bam ->
call-task-path/execution/result_alignment/output.bam

待讨论：入仓目录结构问题...

- 对于可重用 task，在不同流程或同一流程不同场景下重用时，难以动态创建适当的相对输出目录
- 可考虑在 workflow 的 parameter_meta 中重定义 output 的相对目录结构

```
workflow foo {
  parameter_meta {
    SampleID : { description: "sample name" },
    DataInfo : {
      description: "data info: SAMPLE_ID\tFLOWCELL\tLANE\tLIB\tREAD1\tREAD2"
    }
    output_bam : {
      ArchiveDirectory:"bam/",
      WareHousing: true
    },
    output_vcf : {
      ArchiveDirectory:"vcf/",
      WareHousing: true
    }
  }
}
```

6.5 以 program 为单位创建 task

为便于维护和更新，提高 task 的可复用性，task 的创建应以 program 为单位。

例如，对 anno_db.pl 程序可创建 AnnoDB.task，而不需要分别创建 SNPAnno.task、INDELAnno.task、CNVAnno.task、SVAnno.task 等。

另外，文件名应与 TaskName 相同。

6.6 添加 workflow main 语句

在 task{} 外部，定义 workflow main 语句。作为 task 测试入口。

```
version 1.0

task TaskName { ... }

workflow main { call TaskName }
```

7 测试

7.1 语法检查

功能测试前，可用 womtools.jar 检查 wdl 语法。

用法：`> java -jar womtool.jar validate task.wdl [-i input.json]`

7.2 测试方法及测试用例

- 准备测试小数据
- 准备测试用 input.json (task_test.json)
  格式示例

```
{
  "main.TaskName.program": "String",
  "main.TaskName.input": "String"
}
```

- 在自动化平台的所有队列进行测试
- 所有测试成功后，合并至 gitlab 相关分支
  - 考虑使用持续集成

8 WDL 管理

8.1 版本控制

使用 gitlab 管理 WDL，详见gitlab WDL 库。workflow 和 task 用不同分支管理。

master 用于管理 workflow 和相关文档

wgs-task 用于管理 WGS 相关 task

使用以下命令导出 zip

git archive –format zip -1 –remote ssh://git@gitlab.genomics.cn:2200/flexlab/acs/WDL.git wgs-task –output ”./wgs-task.zip”

meta-task 用于管理 meta 相关 task

8.2 上线流程管理

1. 正式上线的流程存放至约定路径下

   约定路径 /hwfssz1/BIGDATA_COMPUTING/bigdata_autoanalysis/auto/WDL

2. 约定路径子目录 workflow 和 task 分别存放 workflow.wdl 和 task.zip
3. 约定路径进行严格权限控制
4. 约定路径下的内容**必须从 gitlab 获取**，严禁修改后直接上线
5. 维护 changeLog 文件，记录每次更新的 gitlab tag 或版本号