Genome Informatics 2020

Lesson 2 - Portable and reproducible bioinformatic analysis

Lesson overview

1. Common Workflow Language (CWL). Building apps (tools and workflows
2. Docker
3. Constructing and running portable and reproducible bioinformatics analysis
4. Jupyter Notebook bioinformatic analysis on the cloud

Common Workflow Language

COMMON WORKFLOW LANGUAGE

- CWL is a way to describe command line tools execution
- Every tool has defined set of inputs and outputs
- Every tool is executed in its own environment (Docker)
- Execution on the cloud or local environment
- Enables portable and reproducible execution

1st-tool.cwl

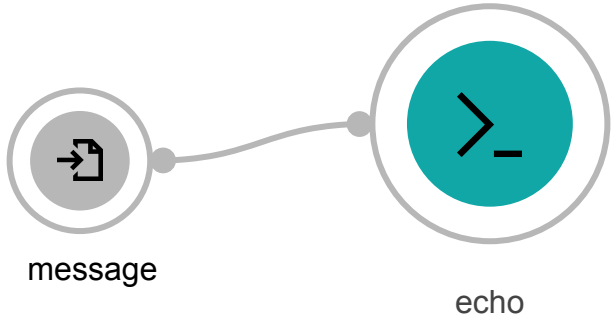
```
#!/usr/bin/env cwl-runner

cwlVersion: v1.0
class: CommandLineTool
baseCommand: echo
inputs:
  message:
    type: string
    inputBinding:
      position: 1
outputs: []
```

echo-job.yml

```
message: Hello world!
```

Used by CWL executor



CWL: Simple instructions for reproducible analyses

```yaml
class: CommandLineTool
cwlVersion: v1.0

id: bam_tools_index
label: Bam Tools Index

requirements:
  - class: DockerRequirement
    dockerPull: 'images.sbgenomics.com/markop/bamtools:2.4.0'
#  - class: InitialWorkDirRequirement
#    listing:
#      - $(inputs.input_bam)

baseCommand:
  - /opt/bamtools/bin/bamtools
  - index

inputs:
  - id: input_bam
    type: File
    inputBinding:
      position: 1
      prefix: '-in'

outputs:
  - id: indexed_bam
    type: File
    outputBinding:
        glob: '*.bam'
    secondaryFiles:
        - .bai
```

Text in YAML or JSON format.

Describes the tools and workflows.

Easier and faster to deploy tools

Wide adoption by 40+ institutes/research groups

Avoids lock-in to a given system

produces the command line

```
/opt/bamtools/bin/bamtools index -in input_bam.
```

How do I learn CWL?

You can learn the syntax: **CWL User Guide**

BUT you don't have to!

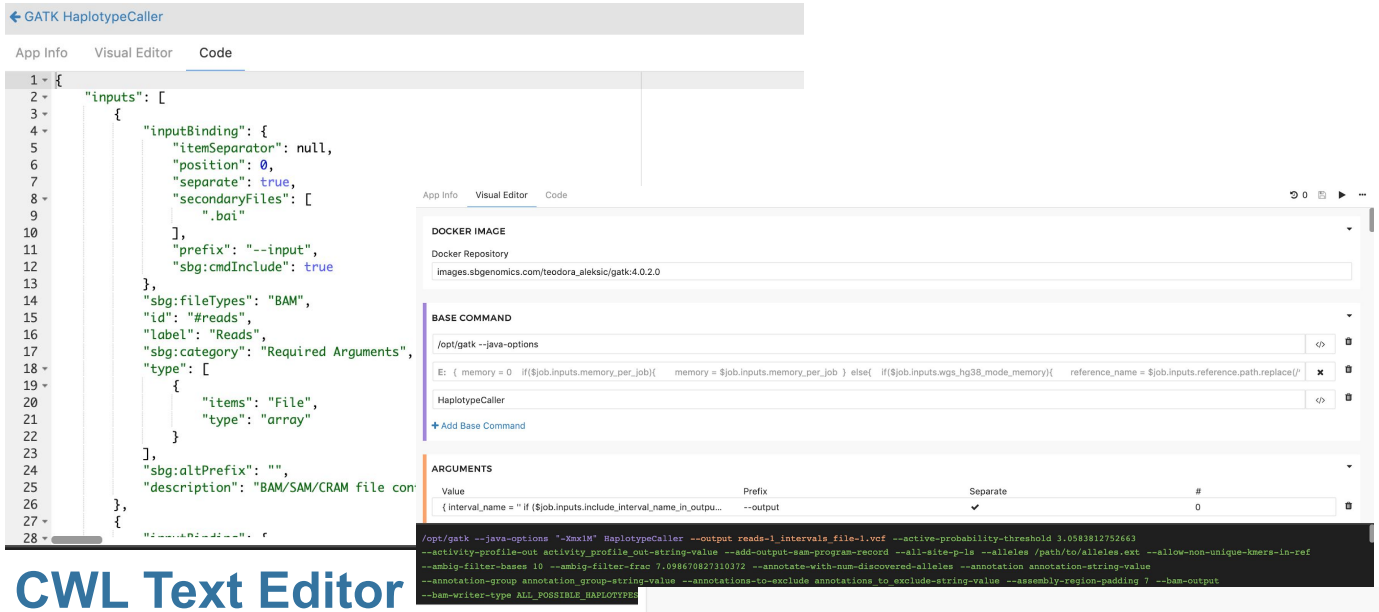
With the Seven Bridges **Software Development Kit** (Tools/Workflow Editor & Rabix Composer), you can easily create tools and chain them into workflows interactively and without any programming experience.

The Seven Bridges SDK will create the CWL code for you so you can get your tool up and running on the platform more quickly and easily.

Rabix Composer

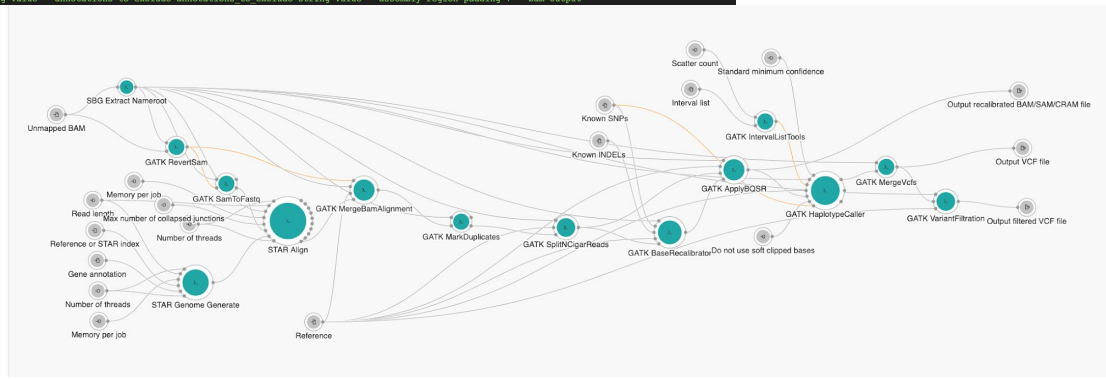
An Integrated Development Environment for CWL developers

- **Compatible** with different versions of CWL
- **Version history**
- **Graphical editors**
- **In-line documentation**
- Support for popular **scripting** languages
- **Desktop Version** - local testing
- **Web Composer**



CWL Text Editor

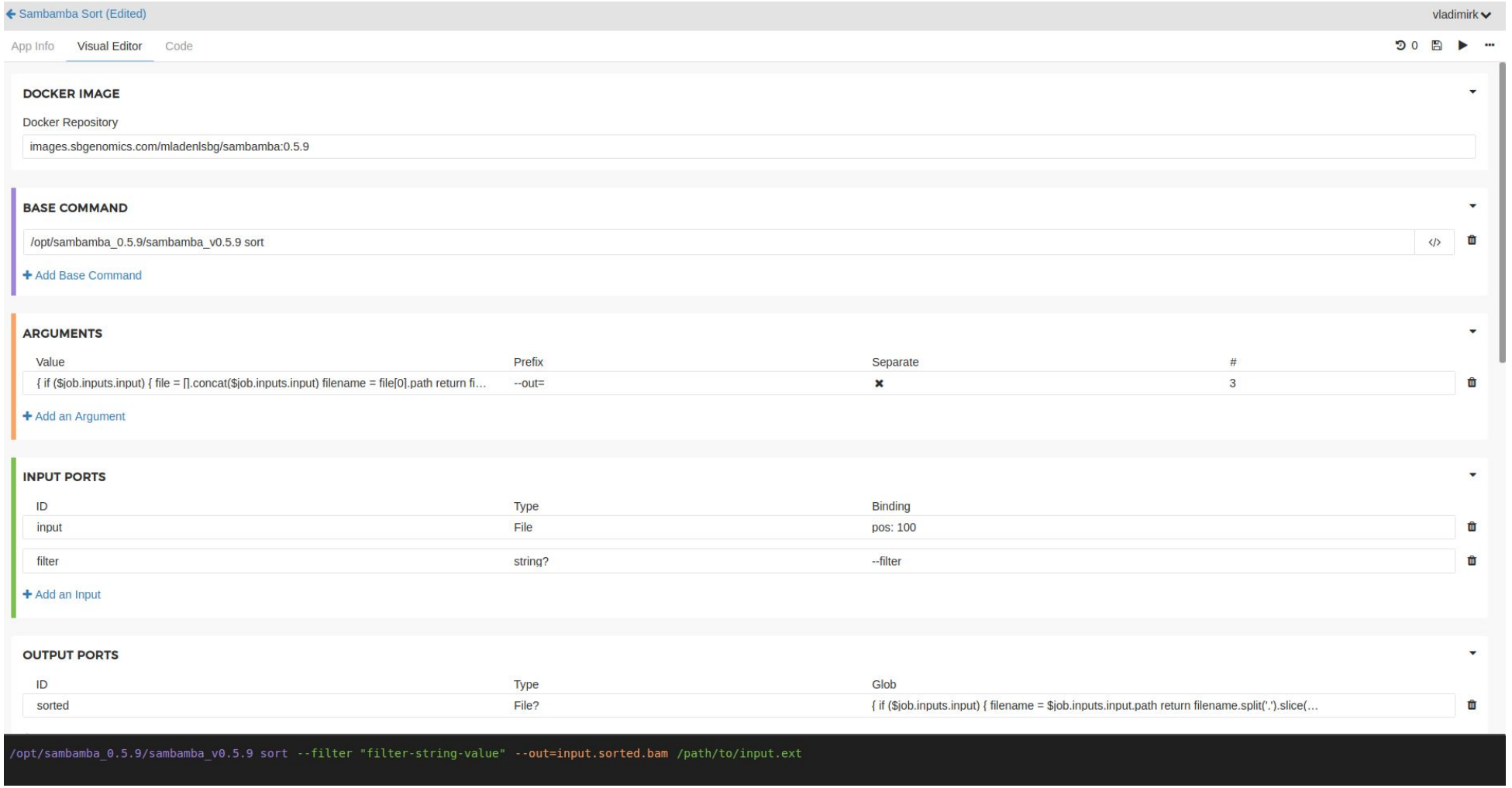
Tool Editor



Workflow Editor

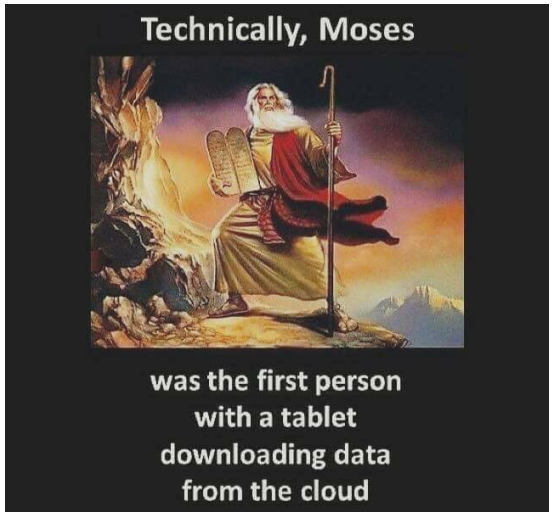
Rabix Composer

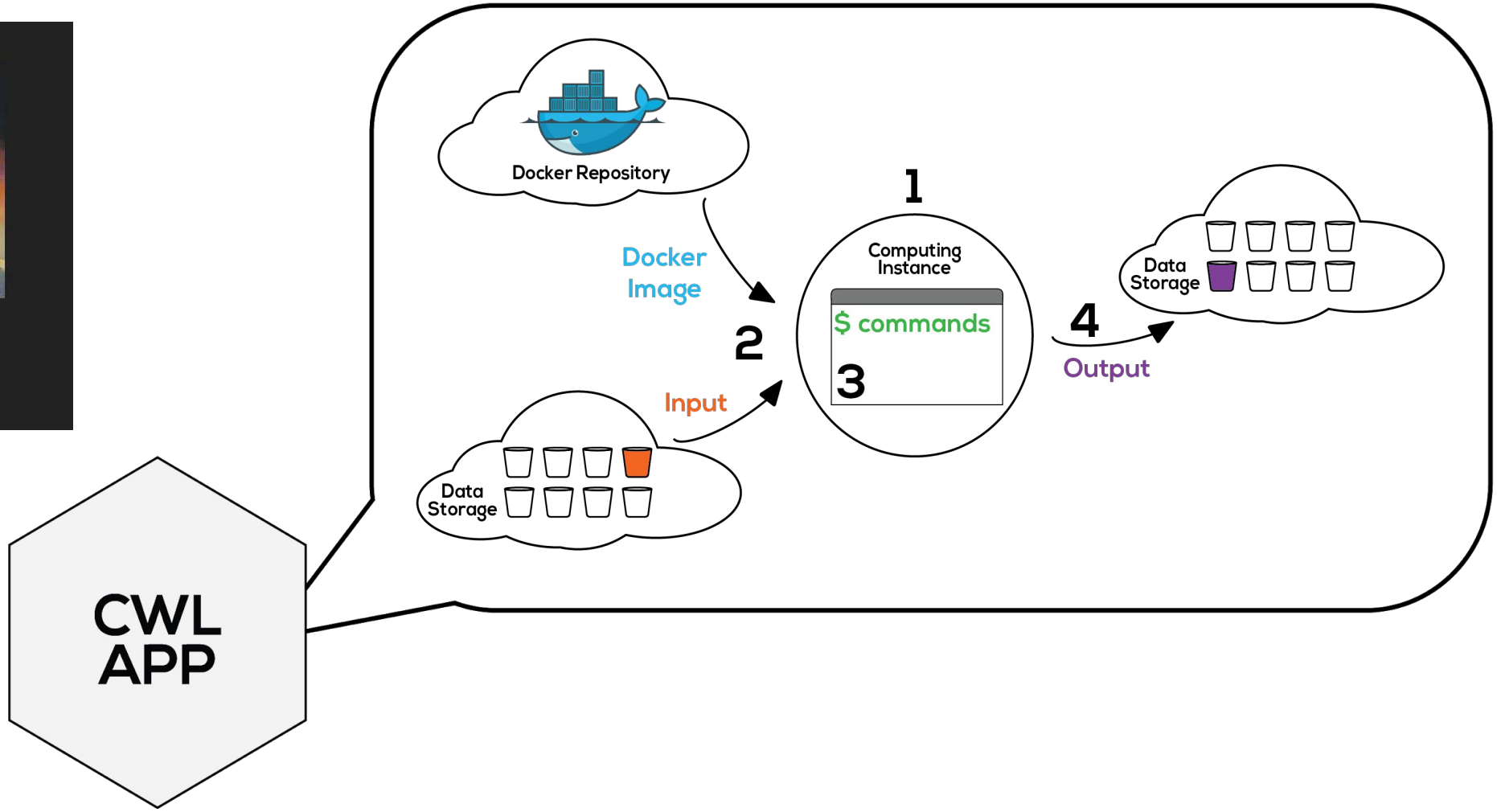
An Integrated Development Environment for CWL developers



CWL @ Cloud

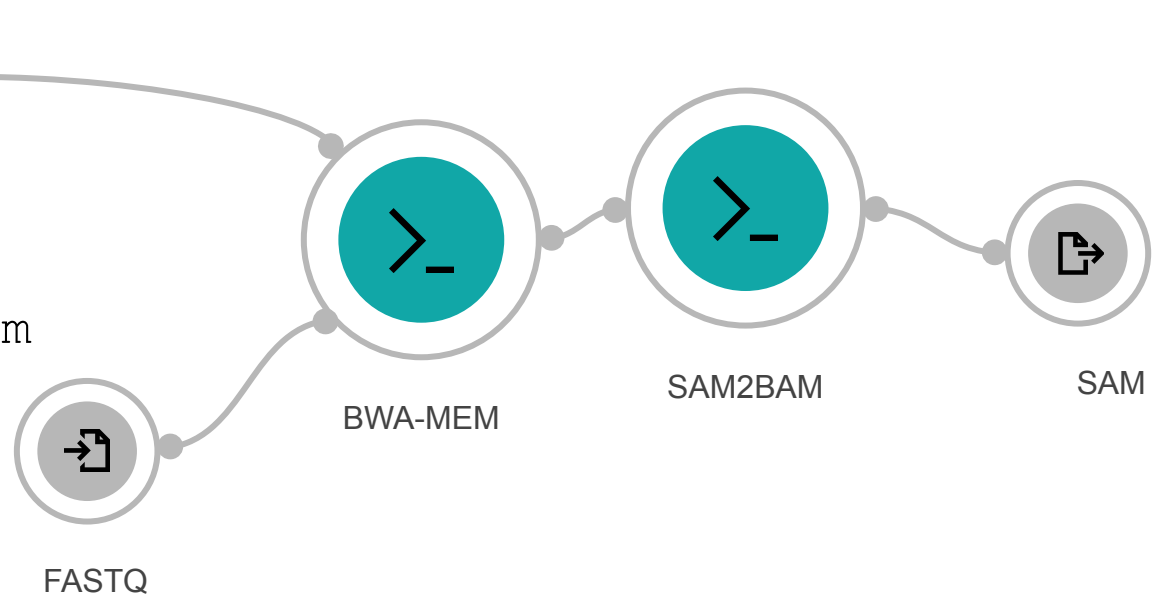
COMMON WORKFLOW LANGUAGE





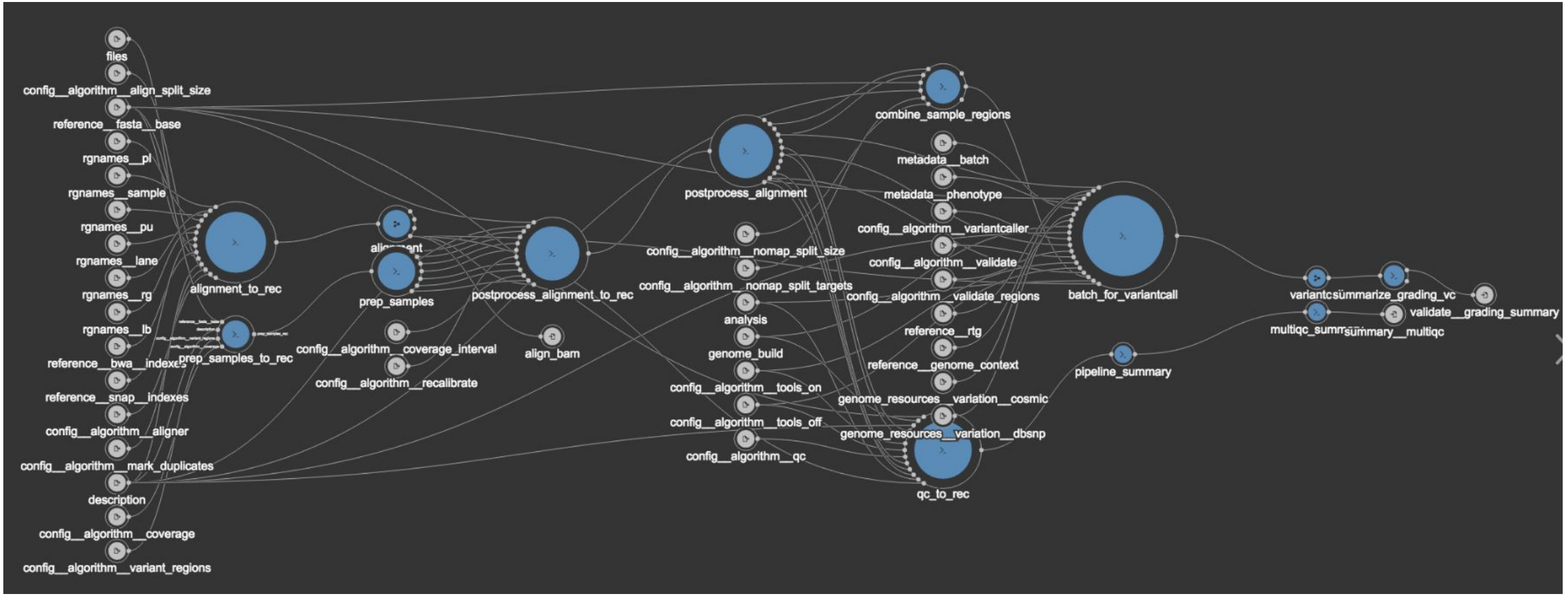
What is a CWL workflow?

- Acyclic graph of tools connected to perform some analysis
- Workflow's nodes are:
  - Inputs (file or parameter)
  - Tools
  - Outputs
  - Workflow

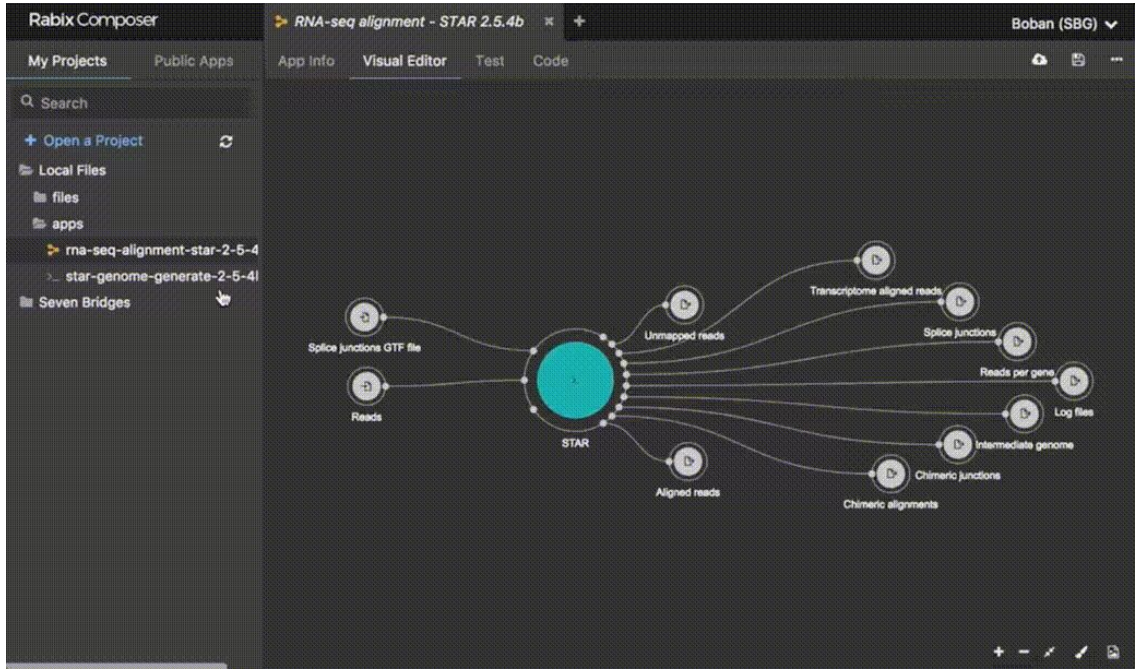


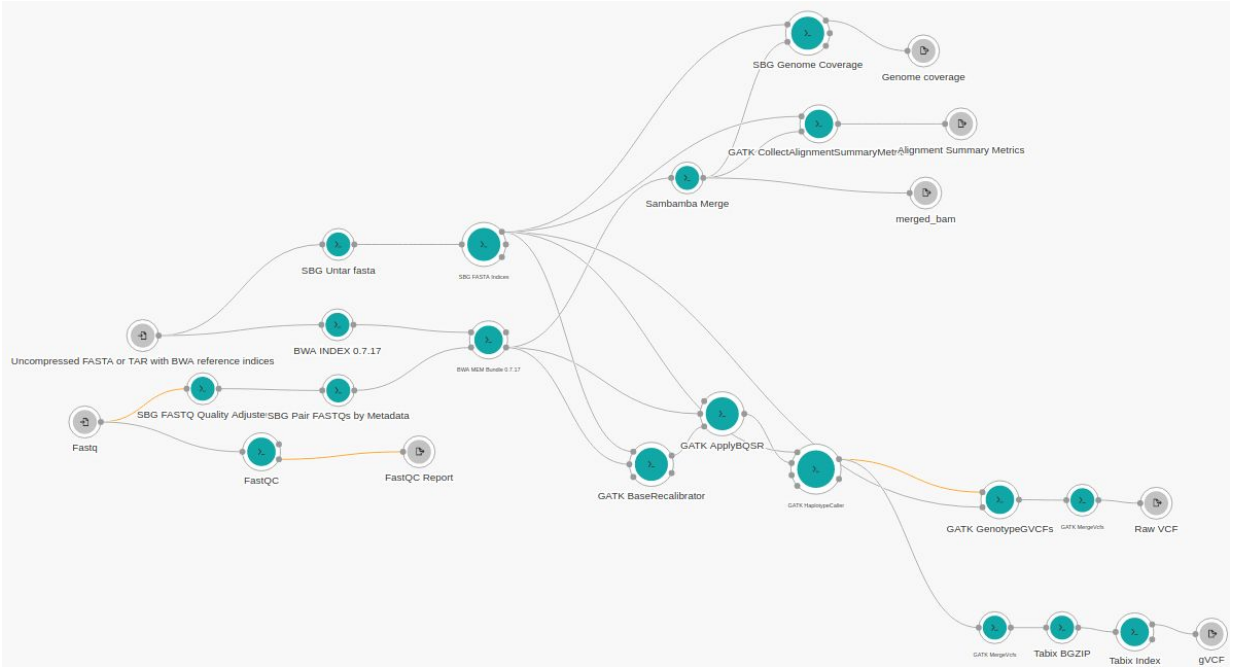

```
bwa mem ref.fa read1.fq read2.fq > aln.sam
sam2bam aln.sam > aln.bam
```

Why we need a workflow?



How to build a workflow?





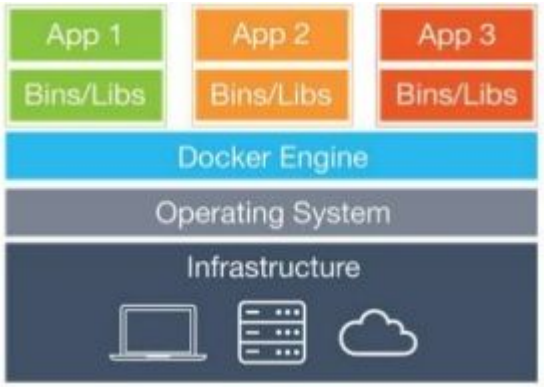
Desktop CWL composer

Web CWL composer

2. Docker

What is Docker?

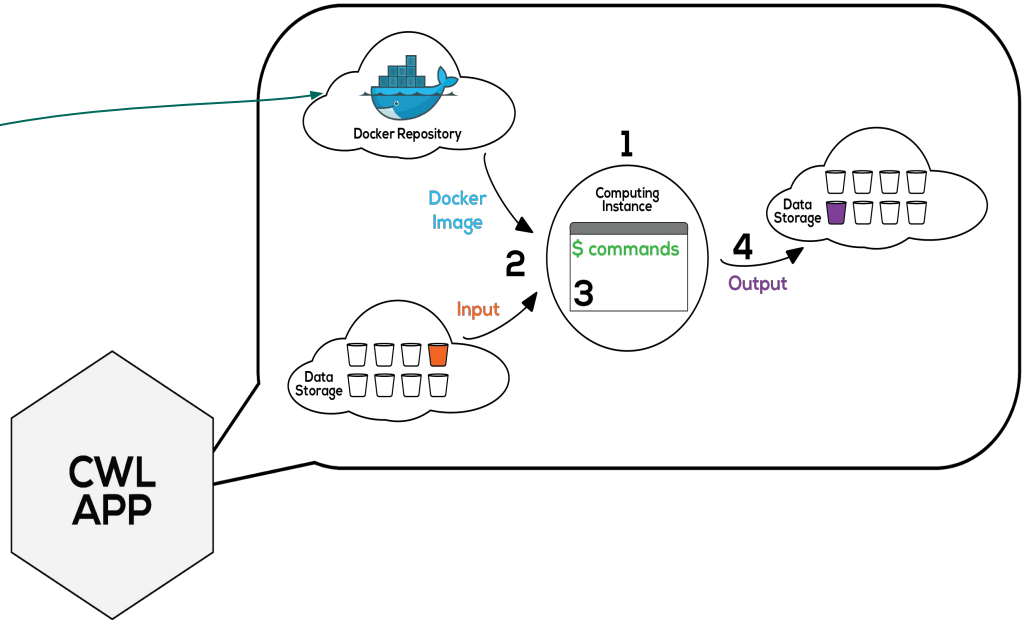
- Docker is a light-weight virtual environment
- Allows you to package the tool (e.g. Python script or some C program) with all of its dependencies into the standardized unit for software development
- Docker containers run on any computer, on any infrastructure
- Layered container structure
- Can directly access resources of host operating system



How to create Docker image?

```
FROM ubuntu:16.04
MAINTAINER vladimir.kovacevic@sbgenomics.com
RUN apt-get update && apt-get install -y wget \
make \
gcc \
zlib1g-dev
WORKDIR /opt
RUN wget
https://github.com/bwa/releases/bwa-0.7.15.tar.bz
2
RUN tar xfj bwa-0.7.15.tar.bz2
WORKDIR /opt/bwa-0.7.15
RUN make
COPY Dockerfile /opt/Dockerfile
```

Docker file




```
# Build image from Dockerfile and push to docker repo

docker build -t images.sbgenomics.com/vladimirk/bwa:0.7.15 .

docker push images.sbgenomics.com/vladimirk/bwa:0.7.15
```

Best practice: Using a Dockerfile

A Dockerfile is a text file that stores commands to create a Docker image

- Uses a domain-specific language to describe how to build an image
- The Docker tool automates the building of an image from a Dockerfile
- Docker reads commands and executes in succession

Benefits:

- Stores whole procedure of image creation
- Helps facilitate and automate the process of maintaining tools that are wrapped for the platform
  - Automate builds
  - Can be used as the source of documentation at failure points and can restart failed builds
  - Transparency
  - Easy to share on GitHub/DockerHub

A Dockerfile consists of **Instructions** followed by **arguments** and comments:

```
#Comment
INSTRUCTION arguments
```

Dockerfile Instructions

FROM	• Initializes new build stage and sets Base Image (“pulling”)
RUN	• Executes the command of argument during build process • Execution results are committed to current image and resulting image is used for next instruction • Chain multiple commands with && and \ for a line break
CMD	• Provides default command, which is executed inside container when it’s created based on image • Need to use argument [“/bin/bash”] , as that is how the container is invoked during task execution for SB Platform
ADD	• Used to copy files, directories, or remote file URLs from original location <source> to container destination path <destination> • You can only specify those source paths that are within context directory
COPY	• Used to copy files or directories to container at specified path • Unlike ADD, doesn’t take URL as <source> and will not unpack archived file as <source>
WORKDIR	• Used to set default working directory for container. Instructions will be executed in the defined working directory

Use a Dockerfile to build an image

Dockerfile

```
# Define base image

FROM ubuntu:latest

# Install required packages

RUN apt-get update && apt-get install -y \
        wget \
        python3-pip \
        libhdf5-dev

# Install python modules

RUN pip3 install numpy
RUN pip3 install h5py

#Install Kallisto

WORKDIR /opt
RUN wget https://github.com/pachterlab/kallisto/releases/download/
v0.43.1/kallisto_linux-v0.43.1.tar.gz
RUN tar -zxvf kallisto_linux-v0.43.1.tar.gz
RUN rm -rf kallisto_linux-v0.43.1.tar.gz

# Add to path

ENV PATH /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/opt/
kallisto_linux-v0.43.1

COPY Dockerfile /opt/
MAINTAINER Kristina Clemens, Seven Bridges, <kristina.clemens@sbgenomics.com>
```

3. Constructing and running portable and reproducible bioinformatics analysis

Cancer Genomics Cloud platform

CANCER GENOMICS CLOUD
SEVEN BRIDGES

HOME ABOUT POLICIES KNOWLEDGE CENTER LOGIN

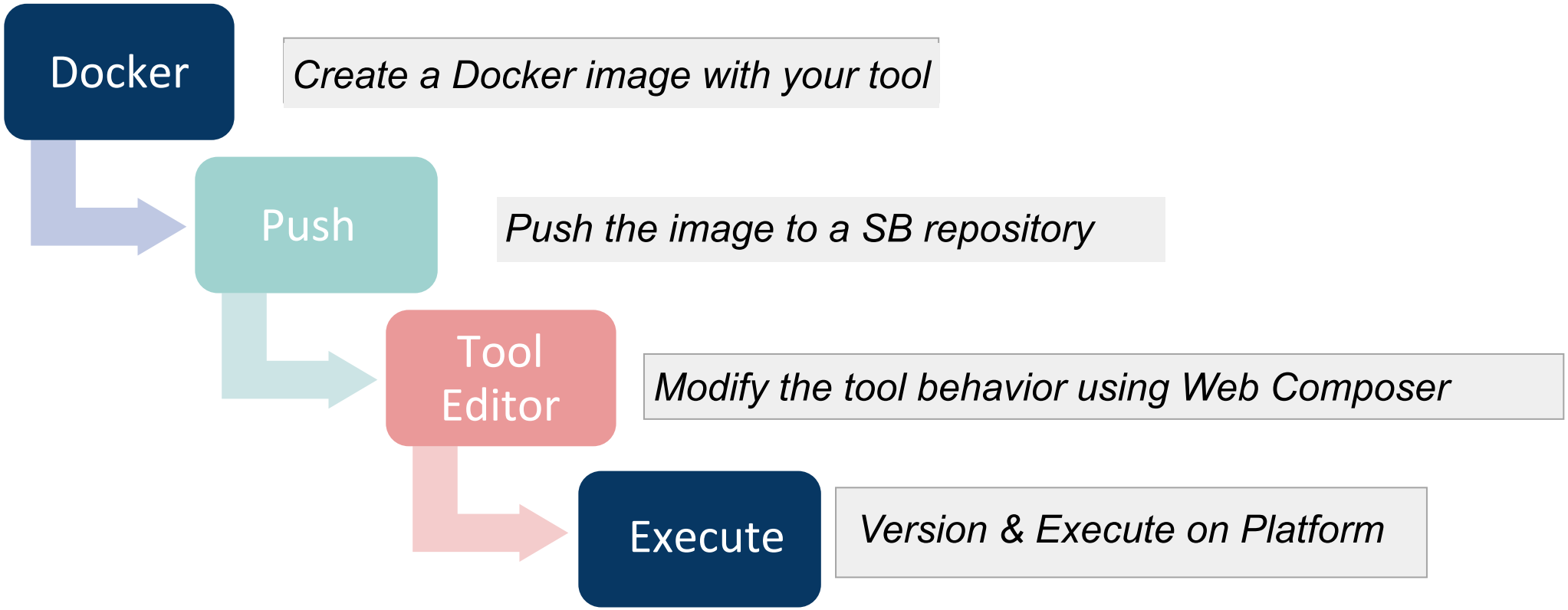
- Two petabytes of multi-dimensional genomics data available to ~3800 authorized researchers to analyse on the cloud
- The Cancer Genome Atlas (TCGA), a landmark cancer genomics program, molecularly characterized over 20,000 primary cancer and matched normal samples

Learn from cancer genomics data.

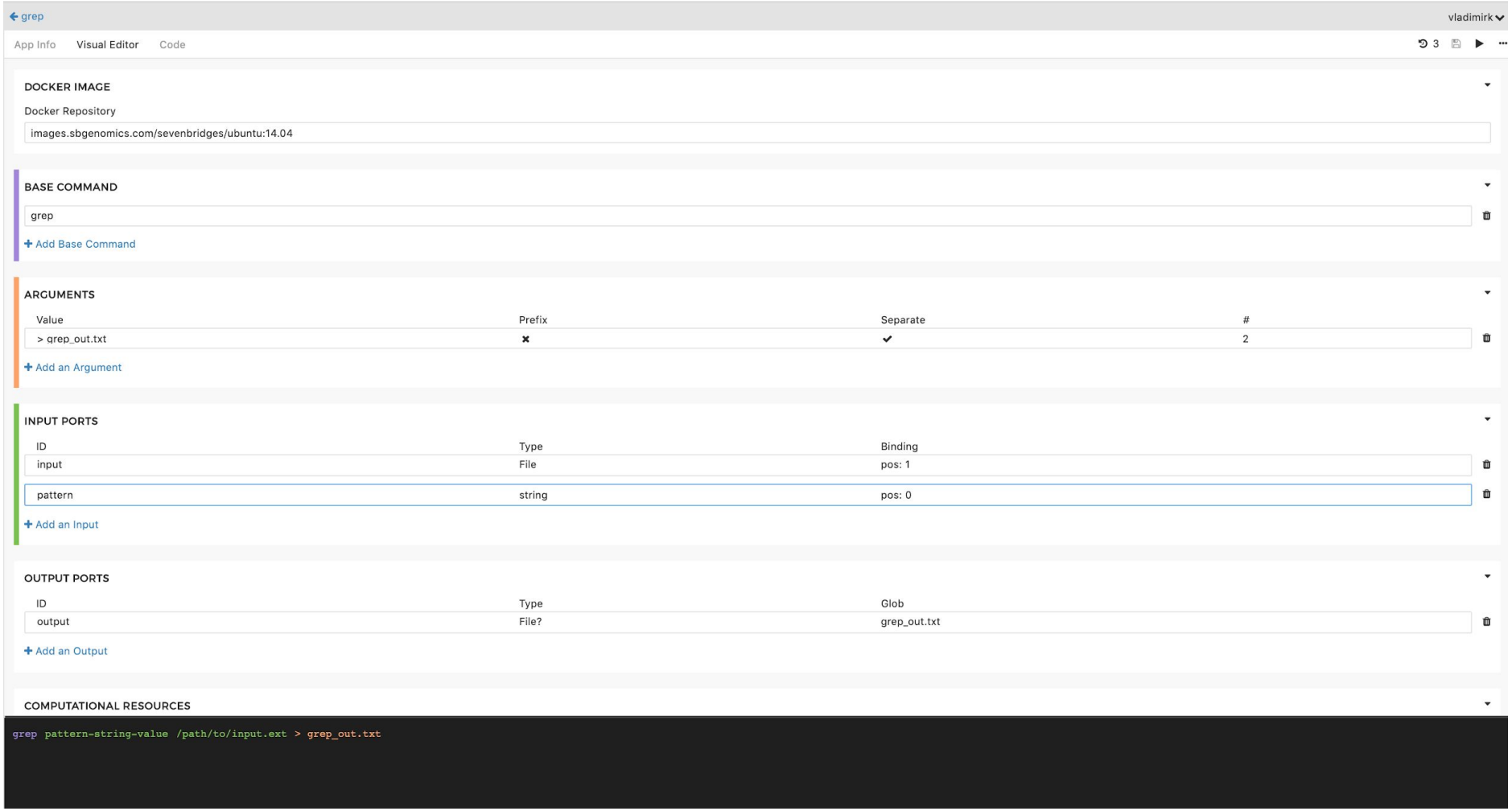
FASTER

- Free registration for academia with $300 credit!

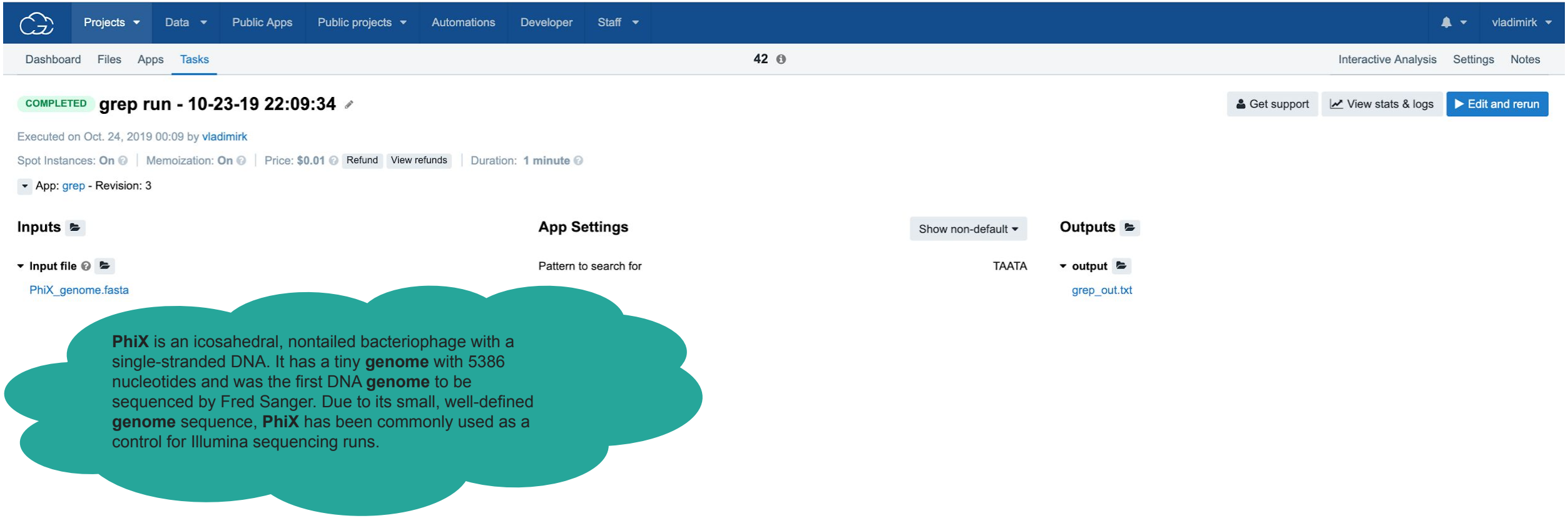
Bringing your own tools to the Platform



Let’s build some tool!



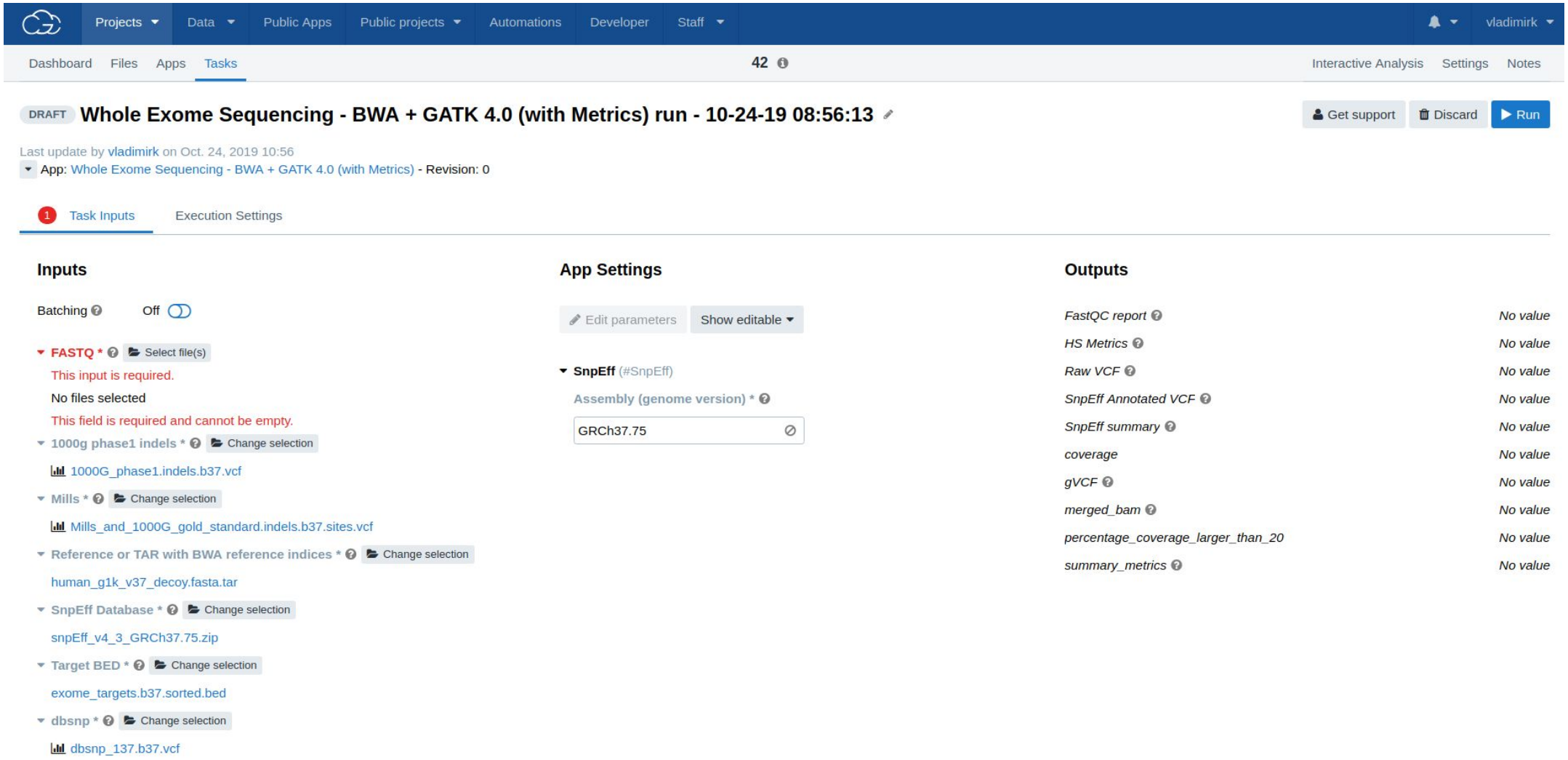
...and run it!


Projects
Data
Public Apps
Public projects
Automations
Developer
Staff
vladimirk
Dashboard
Files
Apps
Tasks
42
Interactive Analysis
Settings
Notes
COMPLETED
grep run - 10-23-19 22:09:34
Get support
View stats & logs
Edit and rerun
Executed on Oct. 24, 2019 00:09 by vladimirk
Spot Instances: On
Memoization: On
Price: $0.01
Refund
View refunds
Duration: 1 minute
App: grep - Revision: 3
Inputs
App Settings
Show non-default
Outputs
Input file
PhiX_genome.fasta
Pattern to search for
TAATA
output
grep_out.txt
PhiX is an icosahedral, nontailed bacteriophage with a single-stranded DNA. It has a tiny genome with 5386 nucleotides and was the first DNA genome to be sequenced by Fred Sanger. Due to its small, well-defined genome sequence, PhiX has been commonly used as a control for Illumina sequencing runs.

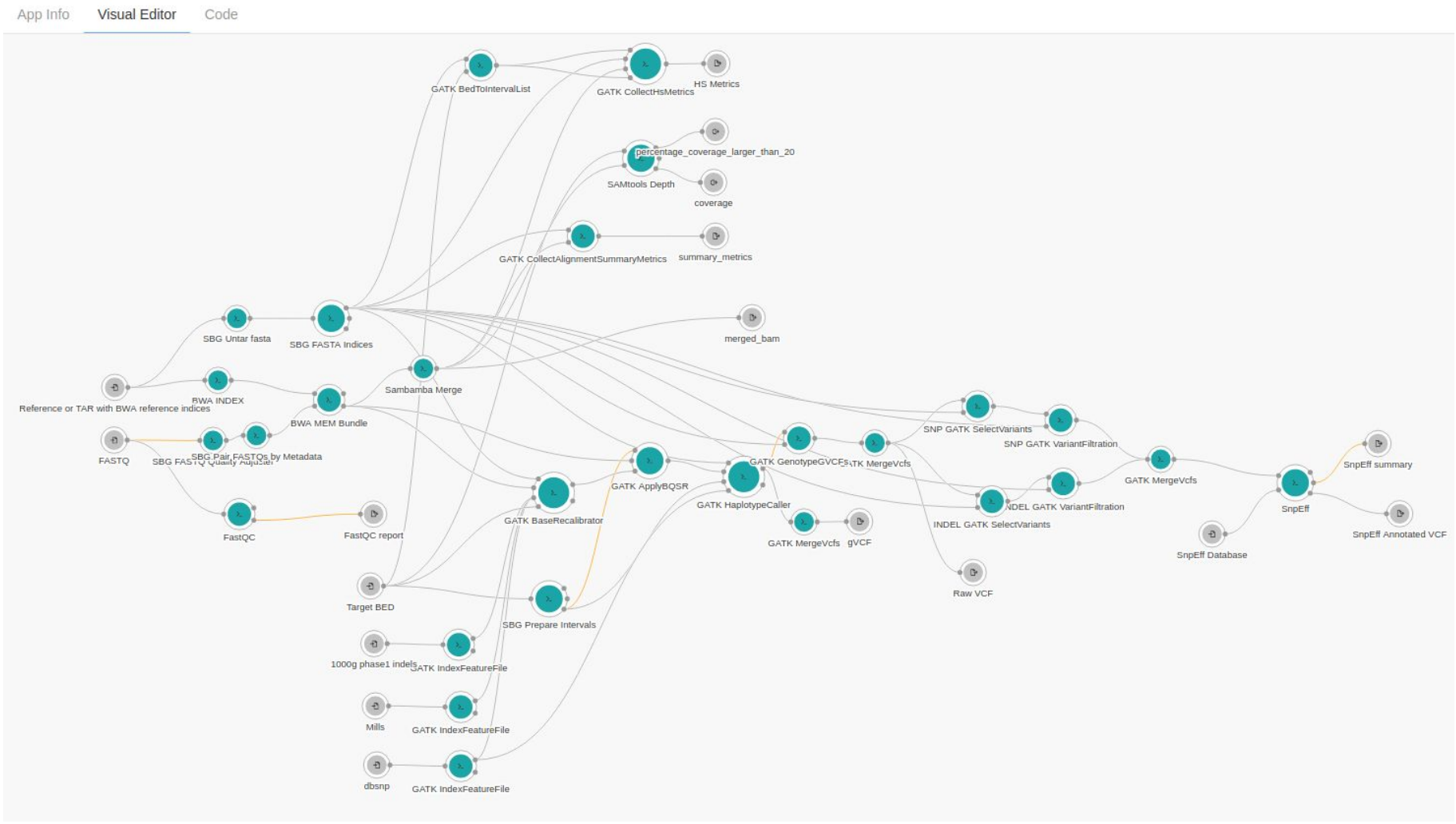
So, what just happened?

- Request for default (c4.2xlarge) instance sent to aws
- Initialize instance
- cwl.job.json created from task inputs and parameters
- Together with cwl.app.json sent to initialized aws instance
- Download input files to the aws instance
- Download of docker image(s) of the tool(s)
- Run the tool inside docker container
- Collect marked outputs and upload them to the cloud storage attached to our platform's project

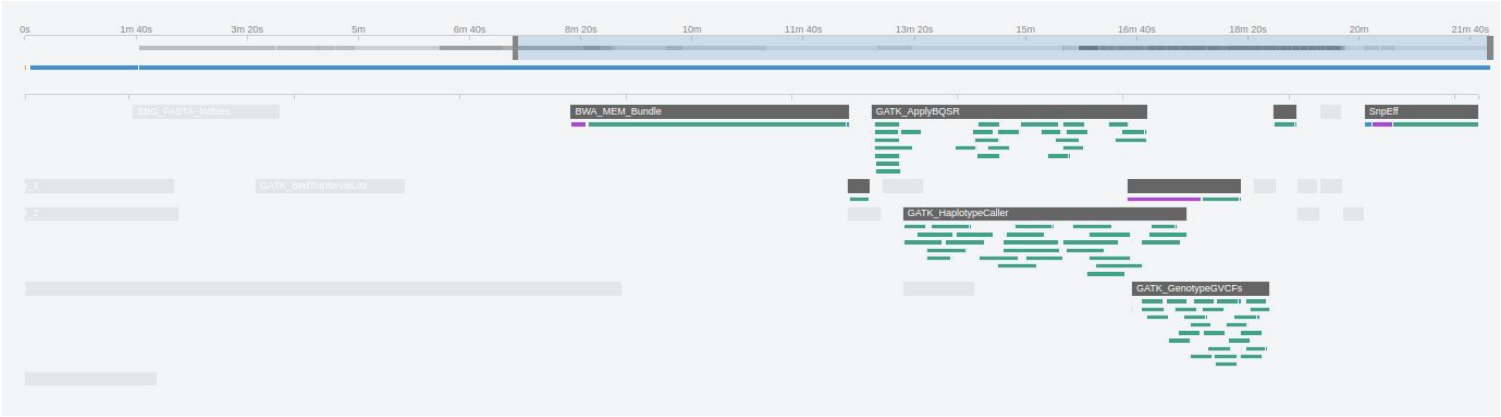
What about some real data?

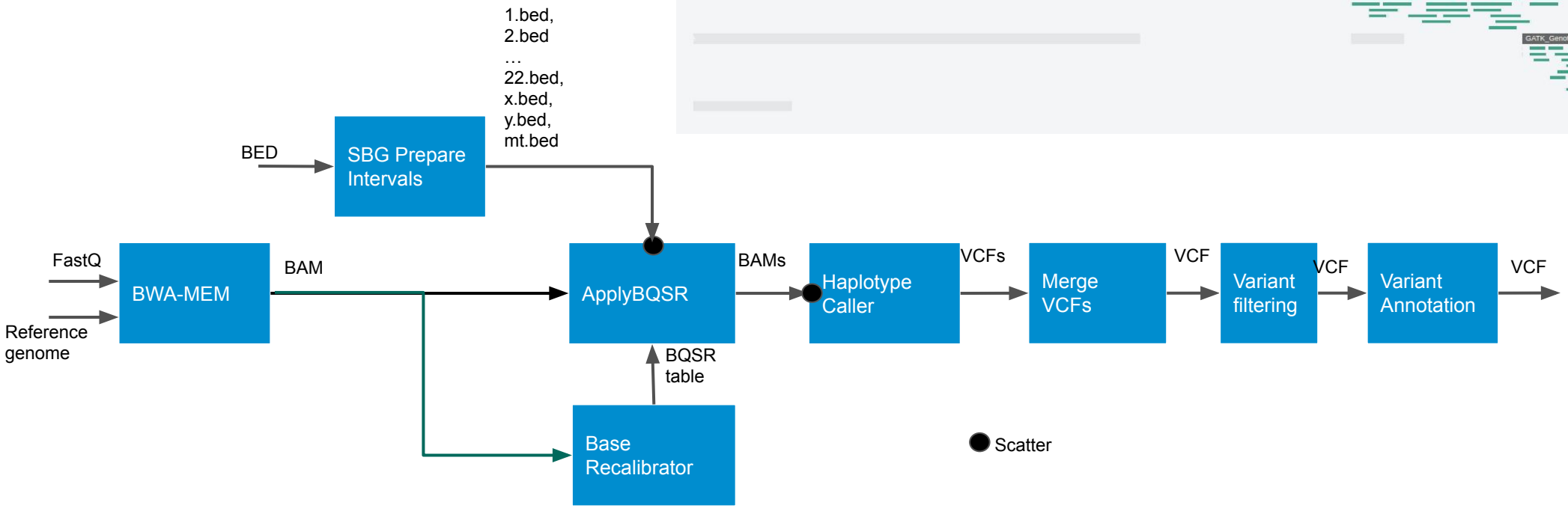


…with real analysis!

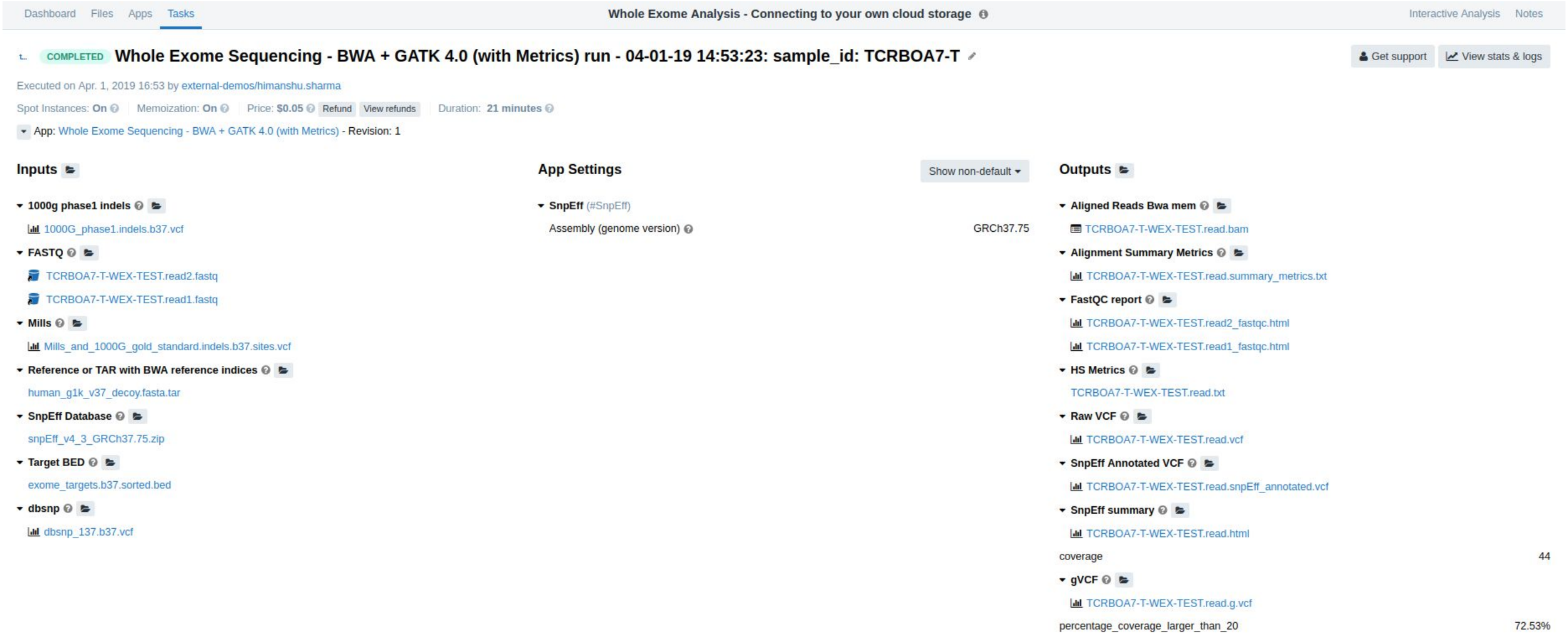


...with real analysis!





Whole exome sequencing execution



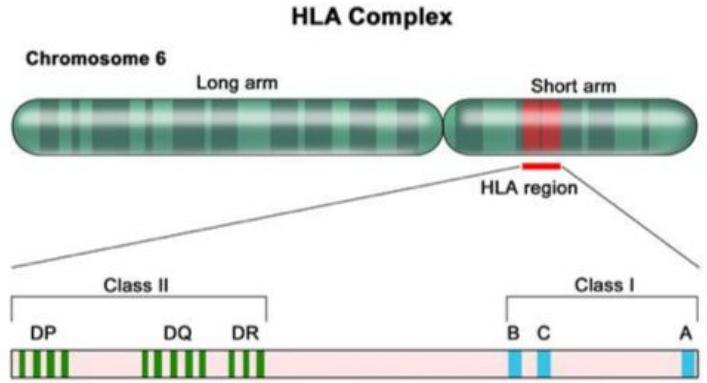
Exercise 1: Wrap FastQC tool

- Complete the tutorial
- Send the link to the executed task at your CGC project

  vladimir.kovacevic@sbgenomics.com together with name and number of index
- Do it before next lesson
- 10 (easy) points :)

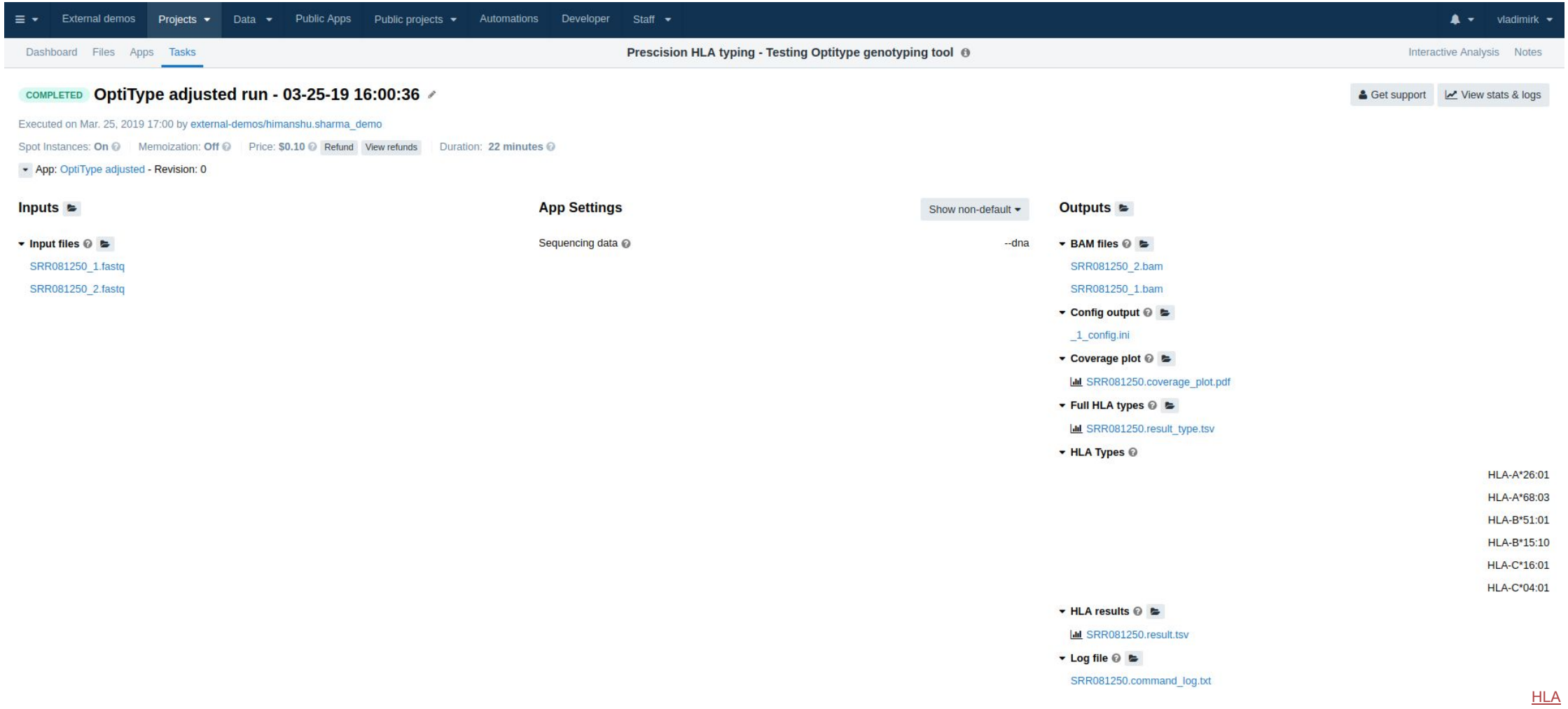
HLA Typing

- The HLA gene family provides instructions for making a group of related proteins known as the human leukocyte antigen (HLA) complex.
- The HLA complex helps the immune system distinguish the body's proteins from proteins made by foreign invaders such as viruses and bacteria.
- HLA typing has been widely used for reducing the risk of organ rejection
- Specific HLA variants are associated with both autoimmune (e.g. type 1 diabetes, rheumatoid arthritis) and infectious (e.g. HIV, Hepatitis C) diseases



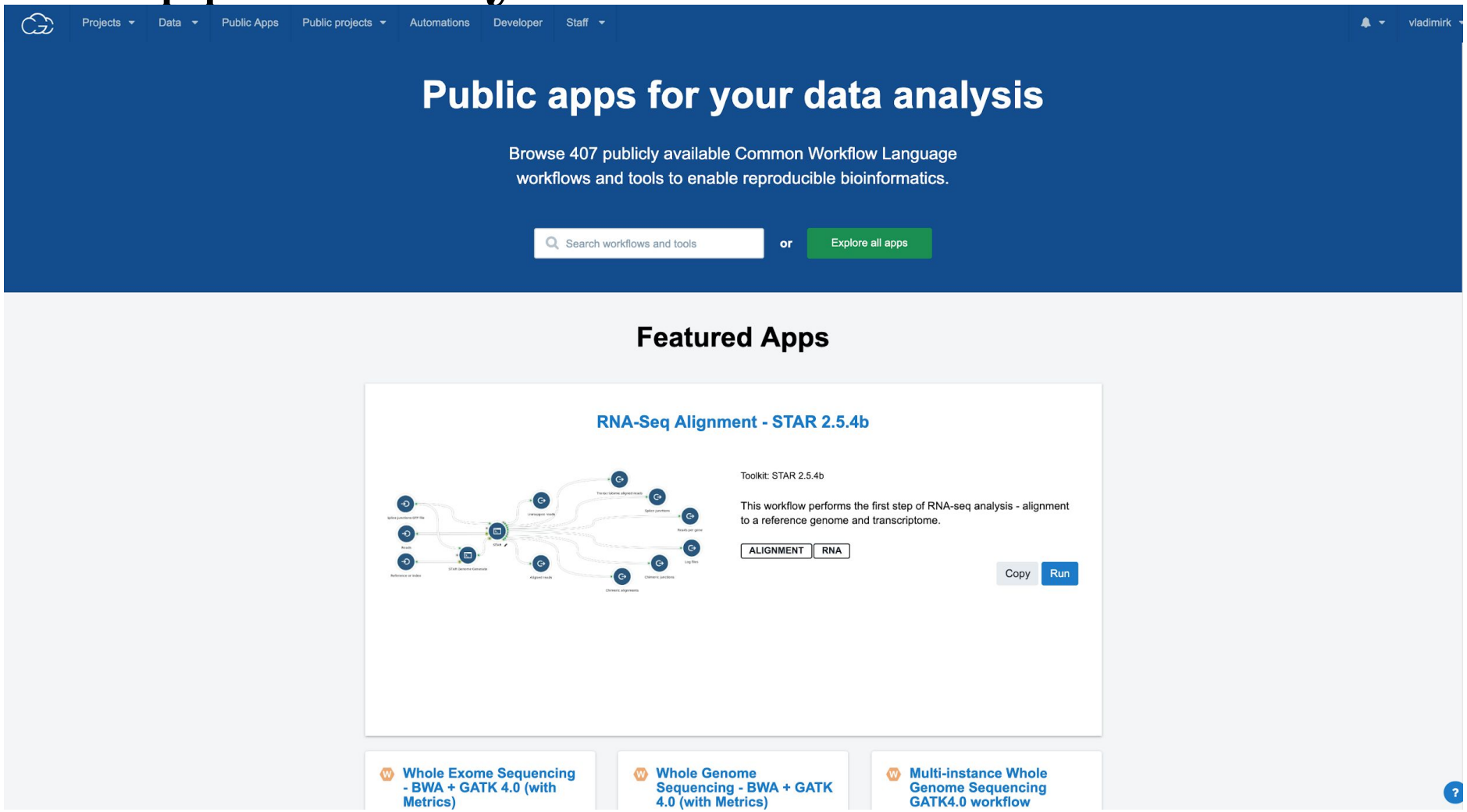
HLA

HLA Typing



HLA

Public App Gallery

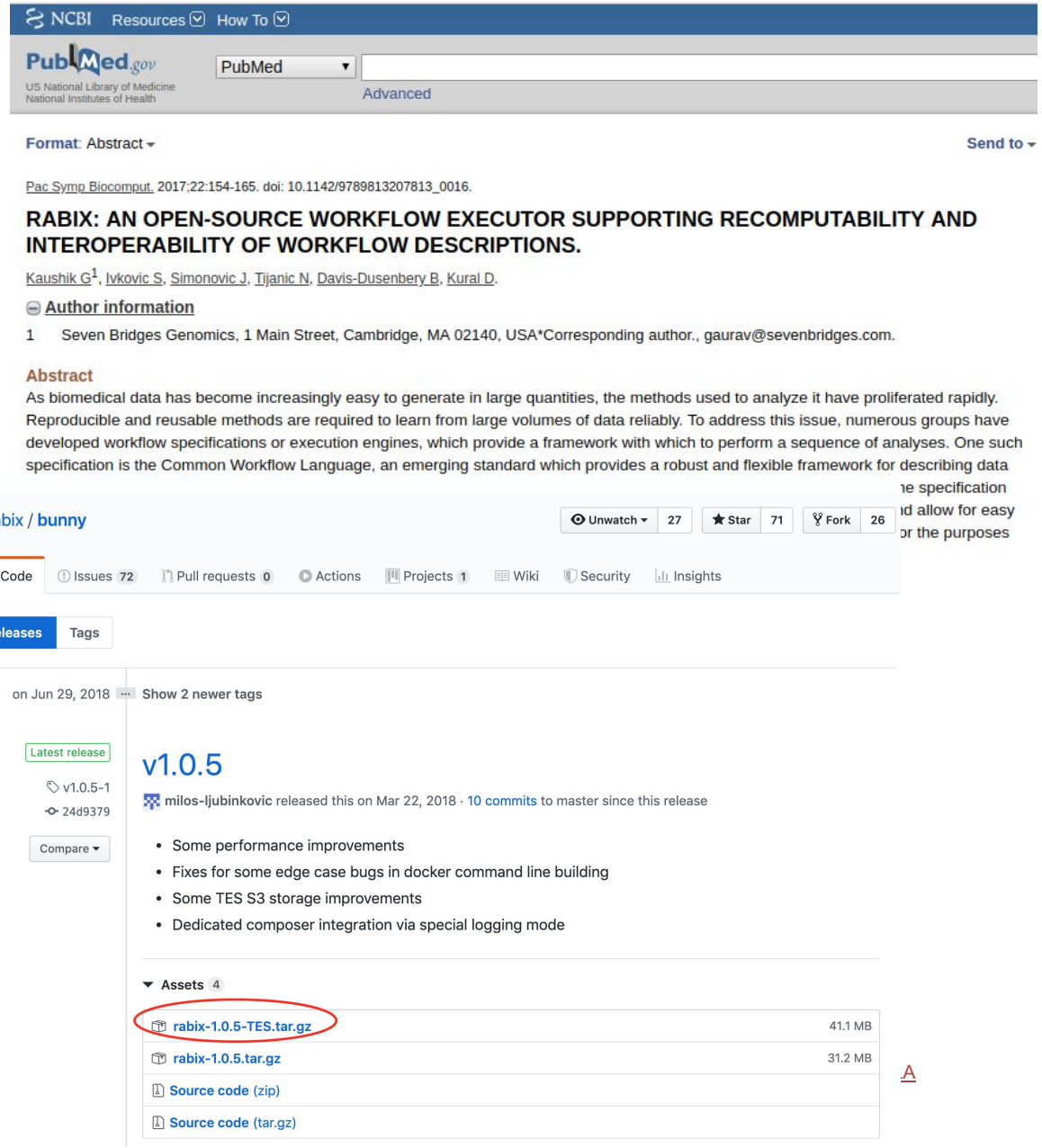


Local executor

Runnable from the command line
Suitable for local testing and development

```
./rabix [OPTIONS] <app> <inputs>
```

rabix.io
https://github.com/rabix/bunny



Local executor

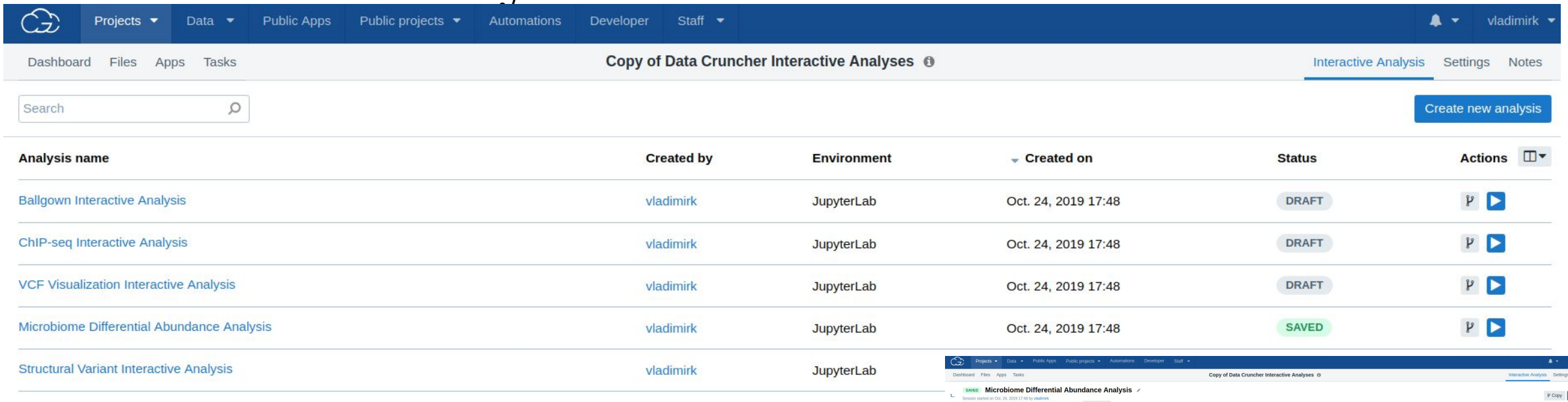
```
# Install docker download and unpack rabix
./rabix -b ./ grep.cwl.json inputs.json
ll grep-2020-02-11-155503.852/root/
-rw-r--r--  1 vladimirk  staff  100 Feb 11 15:55 cmd.log
-rw-r--r--  1 vladimirk  staff  550 Feb 11 15:55 cwl.output.json
-rw-r--r--  1 vladimirk  staff   27 Feb 11 15:55 out.txt
cat grep-2020-02-11-155503.852/root/cwl.output.json
{
  "output" : {
    "basename" : "out.txt",
    "checksum" : "sha1$0a3e8ce4ad3bcd5db0804f28752499adfe2ca5d1",
    "class" : "File",
    "dirname" : "grep-2020-02-11-155503.852/root",
    "location" : "grep-2020-02-11-155503.852/root/out.txt",
    "nameext" : ".txt",
    "nameroot" : "out",
    "path" : "grep-2020-02-11-155503.852/root/out.txt",
    "size" : 27
  }
}
cat grep-2020-02-11-155503.852/root/out.txt
ACTGA
GAGAGAGA
GA
GGGAAAGA
cat grep-2020-02-11-155503.852/root/cmd.log
grep GA dummy.fasta > out.txt
```

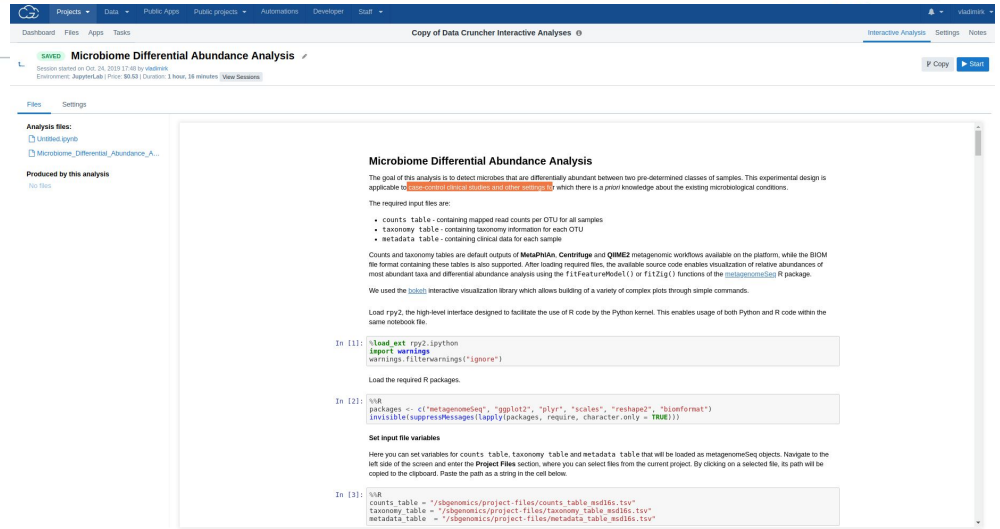
```
# grep.json
{
    "class": "CommandLineTool",
    "cwlVersion": "v1.0",
    "$namespaces": {"sbg": "https://sevenbridges.com"},
    "baseCommand": ["grep"],
    "inputs": [
        {  "id": "pattern",
            "type": "string",
            "inputBinding": {"position": 1},
            "label": "Pattern"},
        {   "id": "input",
            "type": "File",
            "inputBinding": {"position": 2}}
    ],
    "outputs": [
        {"id": "output",
            "type": "File?",
            "outputBinding": {
                "glob": "*.txt"}}
    ],
    "arguments": [
        {"position": 3,"prefix": "",
         "valueFrom": "> out.txt"}
    ],
    "requirements": [
        {"class": "ShellCommandRequirement"},
        {"class": "DockerRequirement", "dockerPull": ubuntu:14.04"}}
```

```
# inputs.json
{
    "input" : {
        "path" : "dummy.fasta",
        "class" : "File"
    },
    "pattern" : "GA"
}
```

4. Jupyter Notebook bioinformatic analysis on the cloud

Interactive analysis



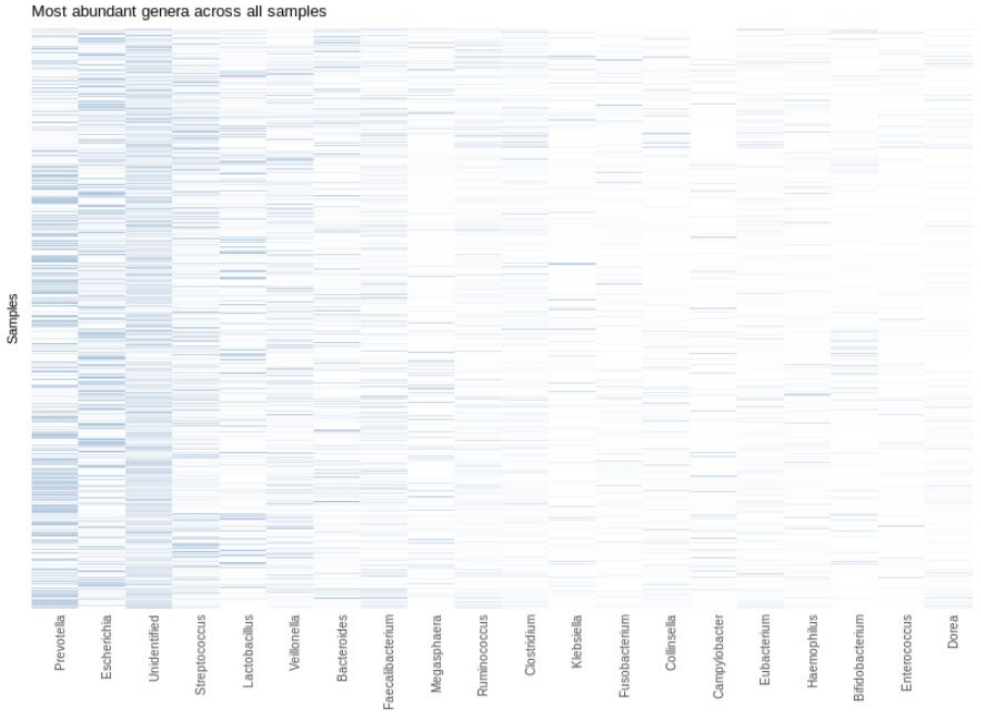


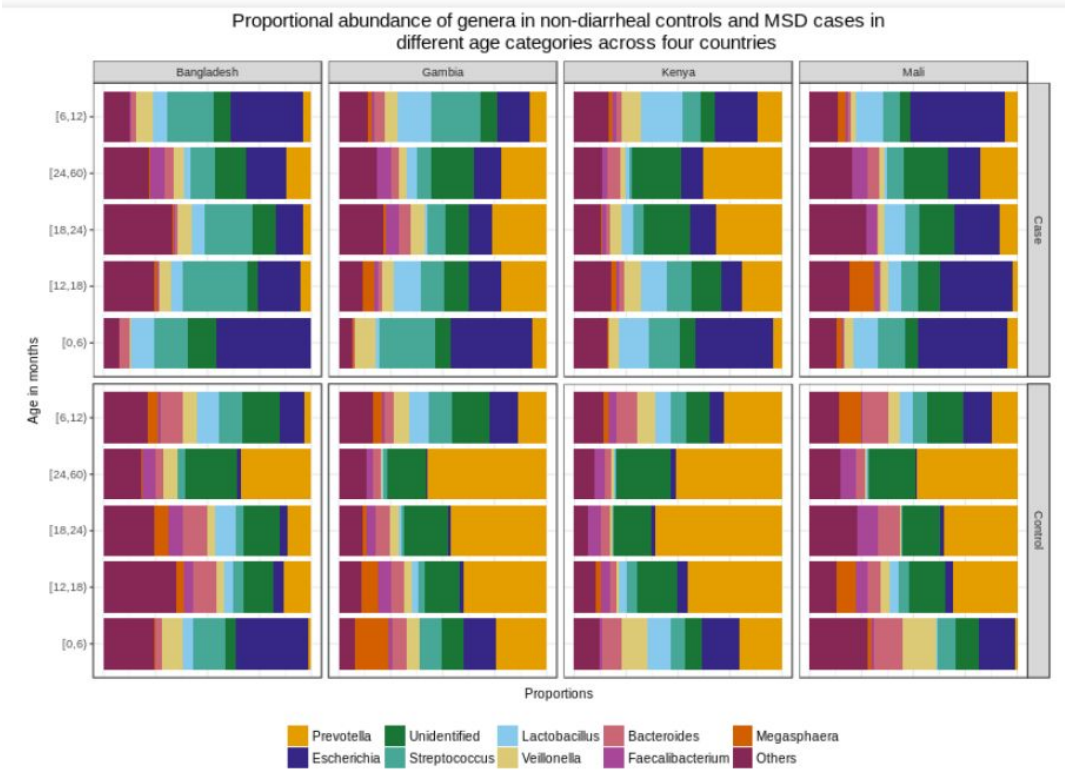
Run python/R Jupyter Notebook on the cloud
Further process outputs from bioinformatics tasks

```
pattern = 'ACCT'
with
open('/sbgenomics/project-files/PhiX_genome.fast
a', 'r') as myfile:
    data=myfile.readlines()
data = ''.join(data).replace('\n', '')
cnt = 0
for i in range(0, len(data) - len(pattern)):
    if data[i:i+len(pattern)] == pattern:
        cnt += 1
        print(cnt, i)
```

Microbiome Differential Abundance Analysis

Detect microbes that are differentially abundant between disease-control (~500 each) samples





HLA

Thank you!

Questions?

@vladimir_bio