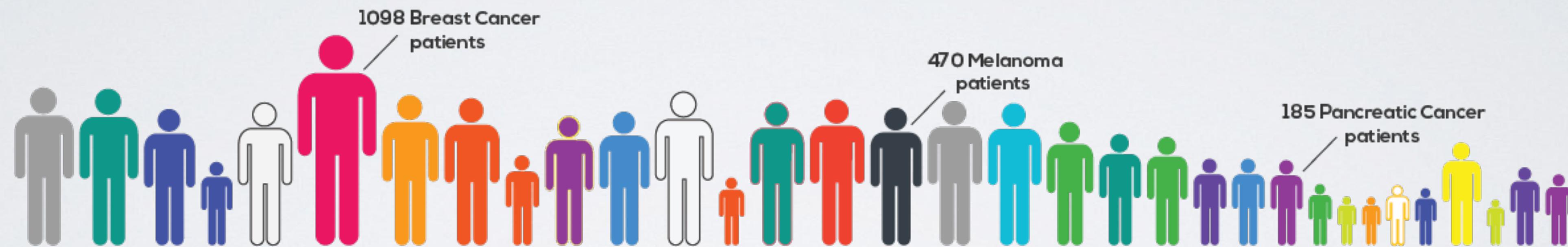


Cancer Genomics Cloud & R: Find, Access, and Analyze Petabyte-Scale Cancer Genomic Data on the Cloud

Nan Xiao
Seven Bridges



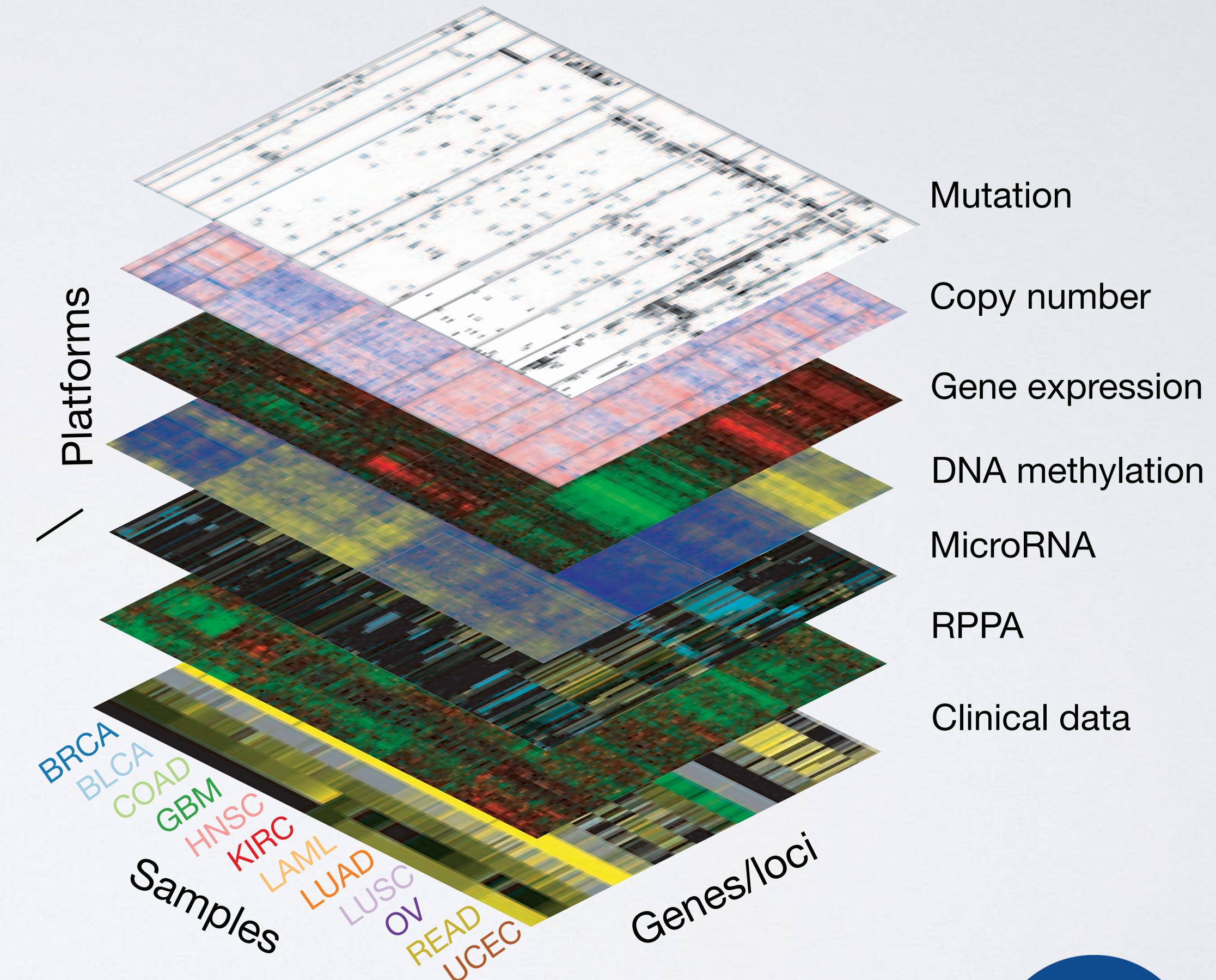
TCGA IS A TREMENDOUS GIFT TO THE CANCER RESEARCH COMMUNITY ...



More than 11,000 cases representing 33 cancer types

UNDERSTANDING TCGA DATASET

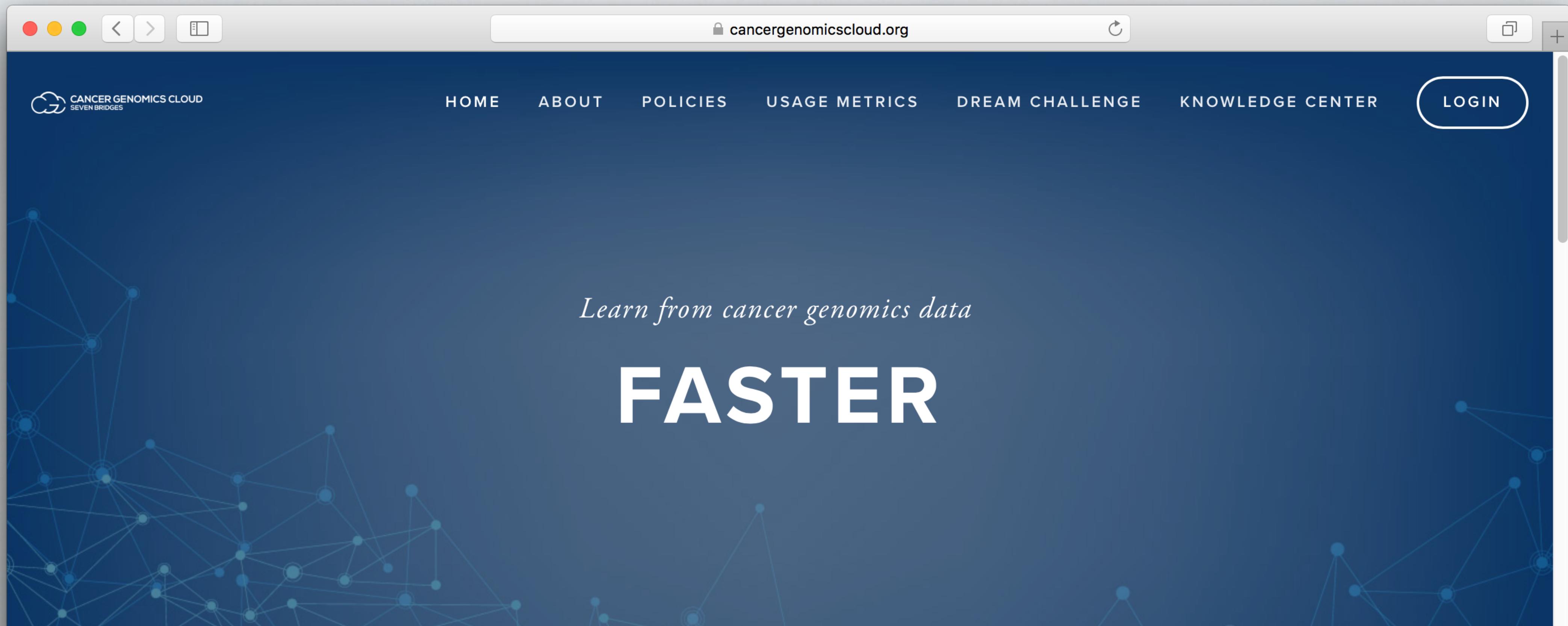
- Multiple Samples per Case
- Multiple Analyses per Sample
- Rich metadata: barcodes, UUIDs, XMLs...



Nature Genetics 45, 1113-1120 (2013)

As the amount and diversity of data increases, it becomes more difficult to learn from them.

The CGC aims to provide a collaborative environment where researchers can take advantage of co-localized public data (like TCGA) and public tools; but also recombine these with their private data and tools.

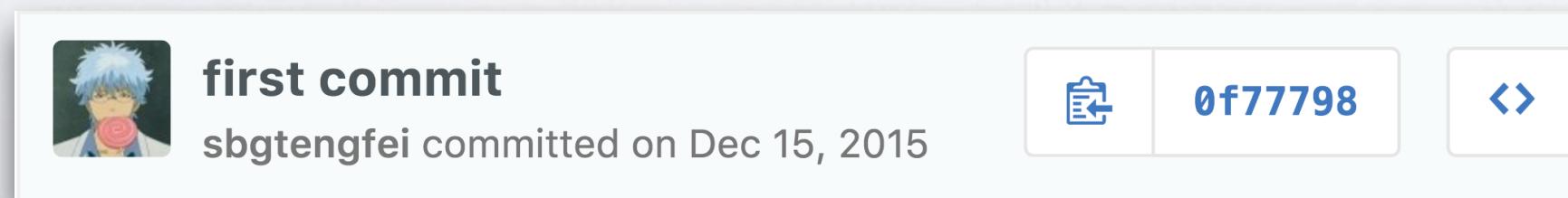


sevenbridges is an R/Bioconductor package offering easier programmatic access to CGC.

The screenshot shows a web browser window displaying the Bioconductor.org website. The URL in the address bar is bioconductor.org. The page is for the 'sevenbridges' package, which is categorized under 'Software Packages' for 'Bioconductor 3.4'. The main content area includes the package name 'sevenbridges' in green, its Bioconductor version (3.4), and various performance metrics: platforms (all), downloads (top 20%), posts (0), build status (ok), commits (1.83), and test coverage (5%). Below this is a brief description: 'Seven Bridges Platform API Client and CWL Tool Builder in R'. The Bioconductor navigation bar at the top includes links for Home, Install, Help, Developers, and About. A search bar is also present. On the right side, there is a 'Documentation' sidebar with links to vignettes, workflows, course material, videos, and community resources, as well as links to R and CRAN packages and documentation.

The `sevenbridges` package offers:

- Complete R API client for CGC/Seven Bridges API
- Common Workflow Language Tool Interface
- Task monitoring / Batch tasks support
- and many helper functions/utilities in R.



2015

downloads top 20%

2017

PROJECT MANAGEMENT

The screenshot shows the CGC Platform's project management interface for the 'demo-brca' project. The top navigation bar includes links for Projects, Data, Public Apps, Public projects, a cloud icon, a help icon, and the user 'nanx'. The main content area has tabs for Dashboard, Files, Apps, Tasks, Interactive Analysis, Settings, and Notes. The 'Files' tab is selected.

Description:

Welcome to your new project!

Projects are the core building blocks of the CGC Platform. Each project corresponds to a distinct scientific investigation, serving as a container for its data, analysis pipelines, and results. Projects are shared only by designated project members.

Within your project, you can:

- Start exploring TCGA dataset straight away
- Install your tools on the CGC and create workflows
- Upload your own private data and analyze it along with TCGA data
- Collaborate securely with other researchers

Please record the details of your project here, such as its aims, experimental context, and any other ideas that you'd like to share with your project members. Remember that details of each pipeline execution you run on the CGC are logged on the task page. This notepad is just for your own notes.

You can also use markdown here to add formatting to your notes.

Good luck with your research! If you get stuck, take a look at the Knowledge Center

The Seven Bridges CGC Team

Members:

nanx OWNER
Write, Copy, Execute, Admin

Email notifications

Don't work alone.
The best research happens in teams.

Invite new members

Share your tools, data, and ideas with collaborators

Tasks:

View all Search

0/1 DESeq2 run - 01-13-17 15:17:35
Submitted by nanx · Jan. 13, 2017 10:18

PROJECT MANAGEMENT VIA API

```
library("sevenbridges")

# Create Auth object
a = Auth(platform = "cgc", token = "your_token")

# List project and details
a$project(owner = "user")
a$project(detail = TRUE)

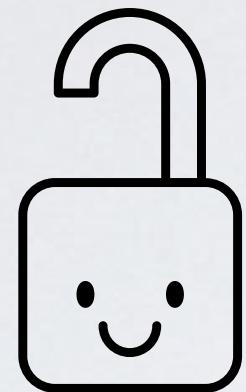
# Get billing group ID
bid = a$billing()$id

# Create new project
a$project_new("brca_test", bid,
              description = "BRCA Test")

# Add member to project
m = a$project(id = "demo/brca_test")$member_add(username = "new_user")
```

FIND DATA ON CGC

MORE THAN ONE PETABYTE OF TCGA DATA AT YOUR FINGERTIPS



Open Data

Information NOT unique
to an individual.

- de-identified clinical data
- gene expression data
- copy number alterations
- epigenetic data



Controlled Data

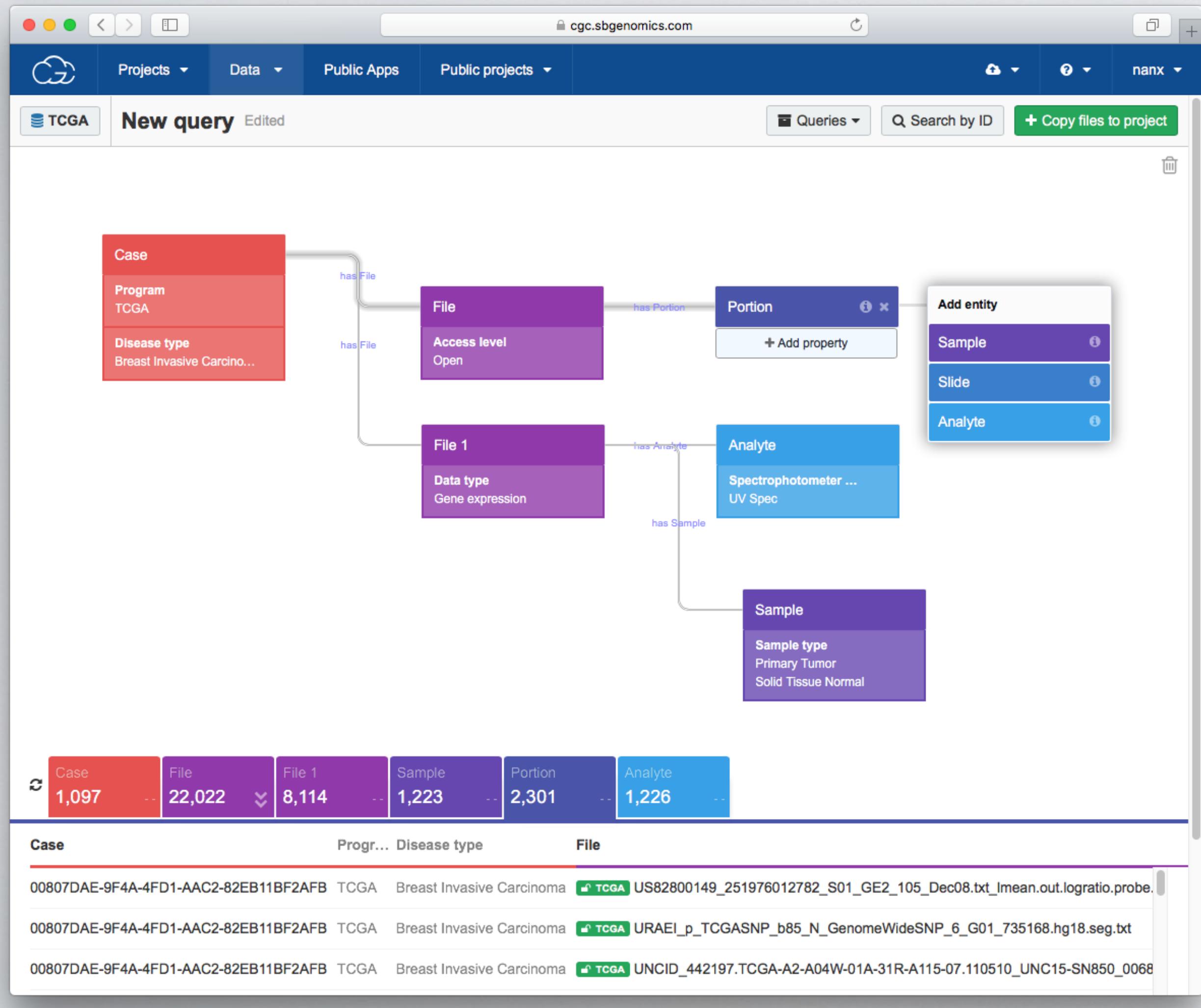
Information that IS unique
to an individual.

- primary sequencing data
- raw & processed SNP6 array data
- raw exon array data
- mutation calls for an individual



CANCER
GENOMICS
CLOUD
SEVEN BRIDGES

CGC DATA BROWSER



- Query metadata
- Add files to project



CANCER
GENOMICS
CLOUD
SEVEN BRIDGES

SPARQL QUERY

```
library("SPARQL")
endpoint = "https://opensparql.sbggenomics.com/blazegraph/namespace/tcga_metadata_kb/sparql"
query = "
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix tcga: <https://www.sbggenomics.com/ontologies/2014/11/tcga#>

select distinct ?case ?sample ?file_name ?path ?xs_label ?subtype_label
where
{
  ?case a tcga:Case .
  ?case tcga:hasDiseaseType ?disease_type .
  ?disease_type rdfs:label 'Lung Adenocarcinoma' .

  ...
}

qd = SPARQL(endpoint, query)
df = qd$results
head(df)
```

DATASETS API

```
term = list(  
  "entity" = "cases",  
  "hasSample" = list(  
    "hasSampleType" = "Primary Tumor",  
    "hasPortion" = list(  
      "hasPortionNumber" = 11  
    )  
  ),  
  "hasNewTumorEvent" = list(  
    "hasNewTumorAnatomicSite" = c("Liver", "Pancreas"),  
    "hasNewTumorEventType" = list(  
      "filter" = list(  
        "contains" = "Recurrence"  
      )  
    )  
  )  
  
a$api(path = "query", body = term, method = "POST")
```

ADD & ANNOTATE DATA

UPLOAD YOUR DATA

- CGC Uploader (GUI)
- Command Line Uploader
- FTP / HTTP / S3 Volume
- API

Add files to Thyroid_tumor_normal

Public reference files

My files

Import from...

Case Explorer and Data Browser

My computer

Cluster or workstation

FTP or HTTP server

Projects

Braf_tumor_normal

testing_api

test project

fusion

QuickStart

opendata only

How to upload files from your computer

We offer a standalone uploading client as a convenient way to upload your datasets from your laptop or desktop computer to Cancer Genomics Cloud.

Cancer Genomics Cloud Uploader is a flexible, fast and secure client that installs on your local computer, can be started and stopped at your convenience and accommodates to a wide range of network topologies.

need it for [Windows](#) or [Linux](#)?

Installing the uploader on Mac OS X

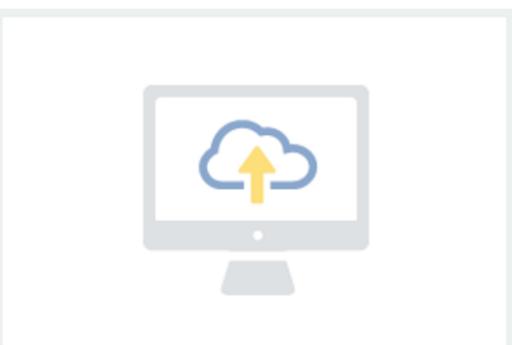
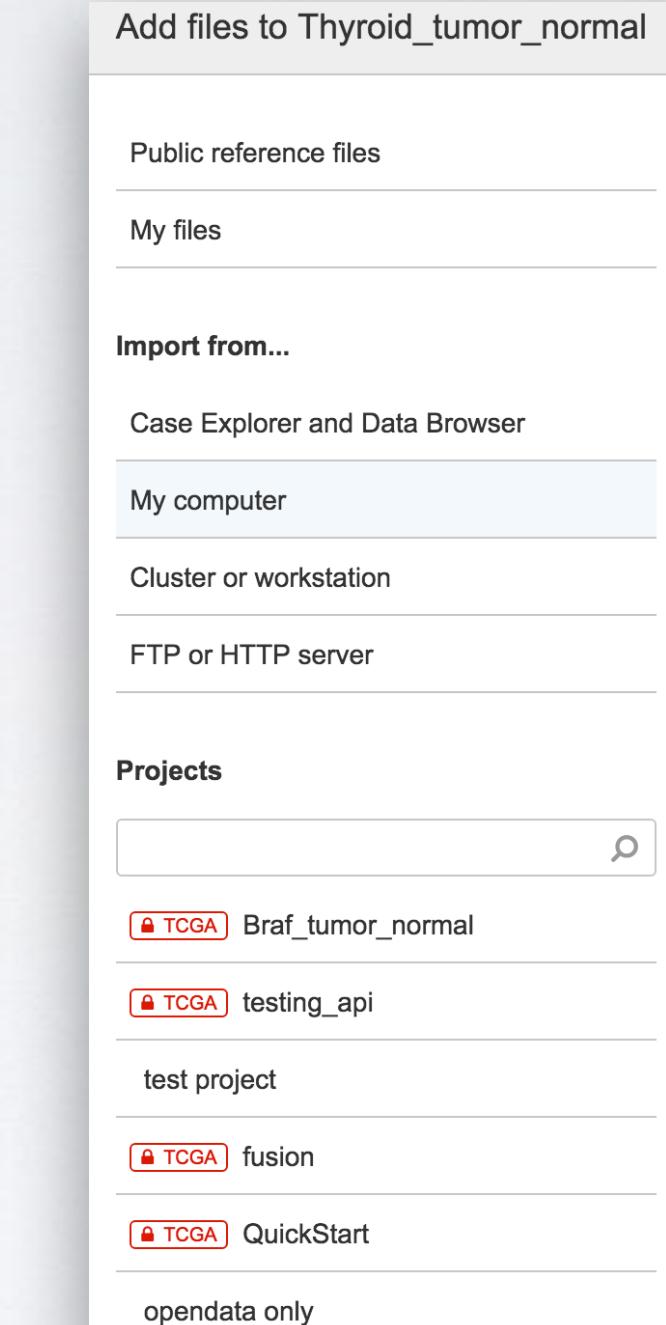
Note:
Cancer Genomics Cloud Uploader works on OS X 10.4 or newer.
If you have an older version of OS X, please use the [command-line uploader](#) instead.

1. Download
Click the button below to download the installer. Double-click the downloaded .dmg file to open it.

2. Install
Drag and drop the Uploader icon to the Applications folder.

3. Run
Locate the Uploader in your Applications folder. Right-click it and select "Open", then "Open" again.

Cancer Genomics Cloud Uploader
Mac OS X



For more information on configuring and using the Uploader, please consult our [User Guide](#).

UPLOAD A FILE VIA API

```
myfile = "file_path.fastq"
p = a$project(id = "demo/tcga-demo")

# Load ` `.meta` for the file by default
p$upload(myfile, overwrite = TRUE)

# Pass metadata manually
p$upload(myfile, overwrite = TRUE,
         metadata = list(library_id = "test_id",
                          platform    = "Illumina x11"))

# Rename file
p$upload(myfile, overwrite = TRUE,
         metadata = list(library_id = "new_id"))
name = "sample_new_name.fastq")
```

UPLOAD FILES / FOLDERS

```
# Upload a folder

dirpath = "path_to_dir"
list.files(dirpath)
p$upload(dirpath, overwrite = TRUE)

# Upload a list of files

dirpath = "path_to_dir"
myfiles = list.files(dirpath, recursive = TRUE, full.names = TRUE)
p$upload(myfiles, overwrite = TRUE)
```

ANNOTATE FILES VIA API

```
# Locate a sample BAM file in project
p = a$project(id = "demo/tcga-demo")
fl = p$file("sample.bam", exact = TRUE)

# Show tags for single file
fl$tag()

# Add new tags
fl$add_tag("new year new tag")

# Set tags to overwrite existing
x = list("this", "is", 2017)
fl$set_tag(x)

# Set metadata
fl$meta()
fl$set_meta()
```

UPLOAD MANIFEST FILE (DEFINES METADATA OF FILES)

```
# attach all metadata except "bad_field" and "sample_id"

p$upload(manifest_file = "~/manifest.csv",
          overwrite      = TRUE,
          subset         = score < 0.5,
          select         = -c(bad_field, sample_id))
```

ANALYZE THE DATA

Run an analysis immediately, with ~230 tools and workflows on the CGC today.

The screenshot shows a web browser window for cgc.sbggenomics.com. The header includes navigation links for Projects, Data, Public Apps, and Public projects, along with user account information for 'nanx'. The main content area has a blue header with the text 'Public apps for your data analysis' and a sub-instruction 'Browse 233 publicly available Common Workflow Language workflows and tools to enable reproducible bioinformatics.' Below this is a search bar with the placeholder 'Search workflows and tools' and an 'Explore all apps' button. Three workflow cards are displayed:

- RNA-seq Alignment - STAR**: STAR 2.4.2a. Description: This pipeline performs the first step of RNA-Seq analysis - alignment to a reference genome and transcriptome. Buttons: 'Alignment' (blue), 'RNA' (blue), 'Copy' (grey), 'Run' (green).
- Whole Exome Analysis - BWA + GATK 2.3.9-Lite (with Metrics)**: SBGTools 1. Description: WES pipeline analyzes all protein-coding genes in a genome (known as Exome). The exome is estimated to comprise ~1– Buttons: 'WES-(WXS)' (blue), 'Copy' (grey), 'Run' (green).
- Fusion Transcript Detection - ChimeraScan**: ChimeraScan 1.0. Description: Fusion Transcript Detection - ChimeraScan detects and identifies fusion transcripts from paired-end RNA-. Buttons: 'RNA' (blue), 'Variant-Calling' (blue), 'Copy' (grey), 'Run' (green).

At the bottom of the page, there are partial views of other workflow cards: 'VarScan2 Workflow from BAM', 'FastQC Analysis', and 'Alignment Metrics QC'.

RUNNING ANALYSIS TASKS

The screenshot shows a web browser window for the URL cgc.sbggenomics.com. The page is titled "demo-brca". The main content displays a "DESeq2 run" task that is currently "RUNNING".

Task Status: RUNNING (01-13-17 15:17:35)

Executed on Jan. 13, 2017 10:18 by nanx

Progress: (0/1) Initializing instance(s) for this task | Duration: Less than a minute

App: deseq2

Inputs:

- Raw count data:
 - TCGA US82800149_251976011805_S01_GE2_105...
 - TCGA US82800149_251976011806_S01_GE2_105...
 - TCGA US82800149_251976011807_S01_GE2_105...
 - TCGA US82800149_251976011808_S01_GE2_105...
 - TCGA US82800149_251976011809_S01_GE2_105...
- Additional metadata: No files selected

App Settings:

- Fit type: parametric
- FDR cutoff: 0.1
- Reference (control) level: No value

View parameters ▾

Outputs:

- DESeq2 analysis summary: No value
- MA plots: No value
- Collected outputs: No value
- DESeq2 analysis results: No value
- A plot of dispersion estimates: No value

COPY APPS & RUN TASKS

```
# List public apps
a$app(visibility = "public")

# Copy an app to project
aid = a$public_app()[[1]]$id
a$copy_app(aid, project = pid, name = "new_app_name")

# Add & run new tasks
tsk = p$task_add(name      = "new_task",
                  description = "new task",
                  app        = "demo/tcga-demo/rna-seq-alignment-star/0",
                  inputs     = list(...))

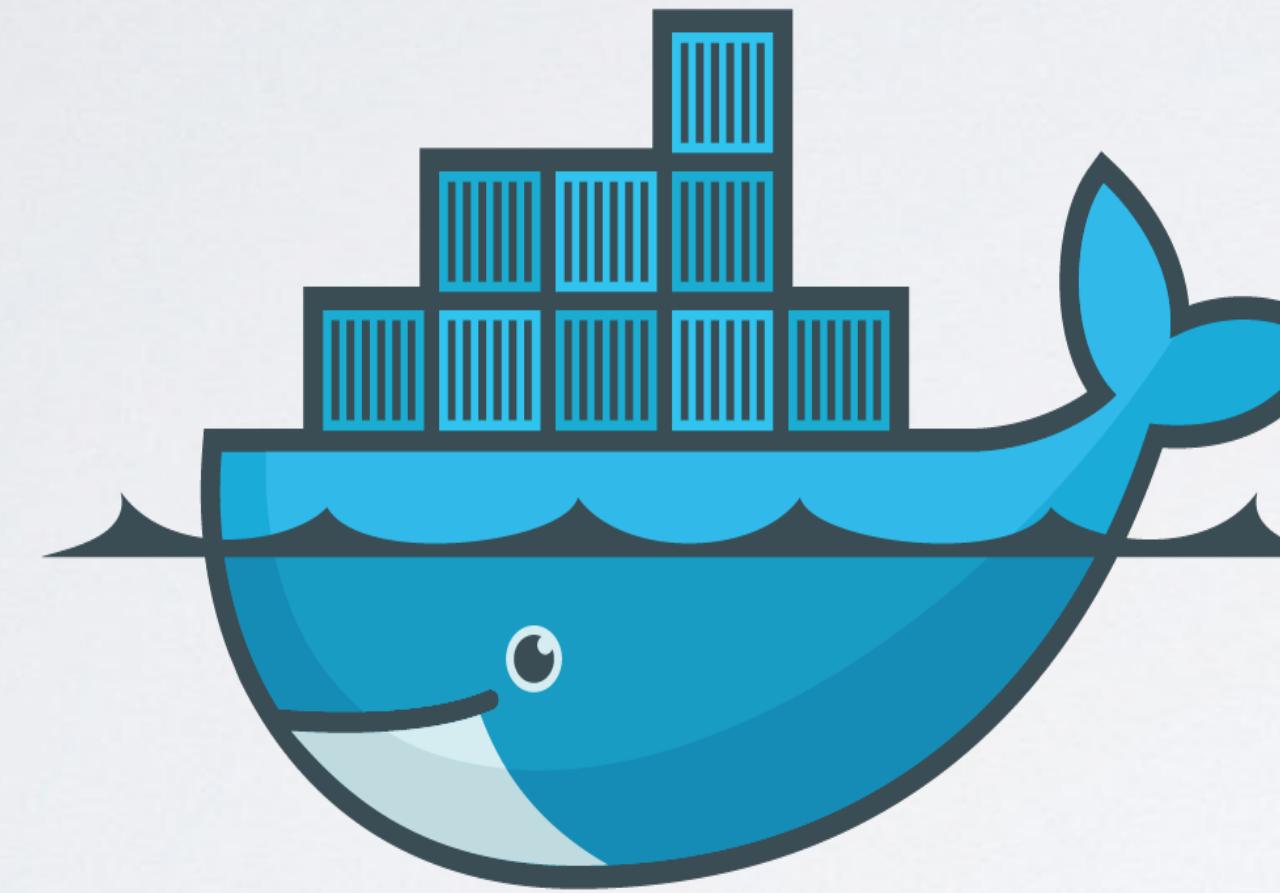
tsk$run()
tsk$abort()
```

RUN TASKS IN BATCH MODE

```
# Batch by items
tsk = p$task_add(name      = "RNA DE Report Batch 1",
                  description = "RNA DE Analysis Report",
                  app        = rna_app$id,
                  batch     = batch(input = "bamfiles"),
                  inputs    = list(...))

# Batch by metadata
tsk = p$task_add(name      = "RNA DE Report Batch 2",
                  description = "RNA DE Analysis Report",
                  app        = rna_app$id,
                  batch     = batch(input = "fastq",
                                    c("metadata.sample_id",
                                      "metadata.library_id")),
                  inputs    = list(...))
```

CONTAINERS + CWL MAKES IT EASY TO PUT NEW
TOOLS ON THE CGC... AND OTHER PLACES.



+



COMMON
WORKFLOW
LANGUAGE



CANCER
GENOMICS
CLOUD
SEVEN BRIDGES



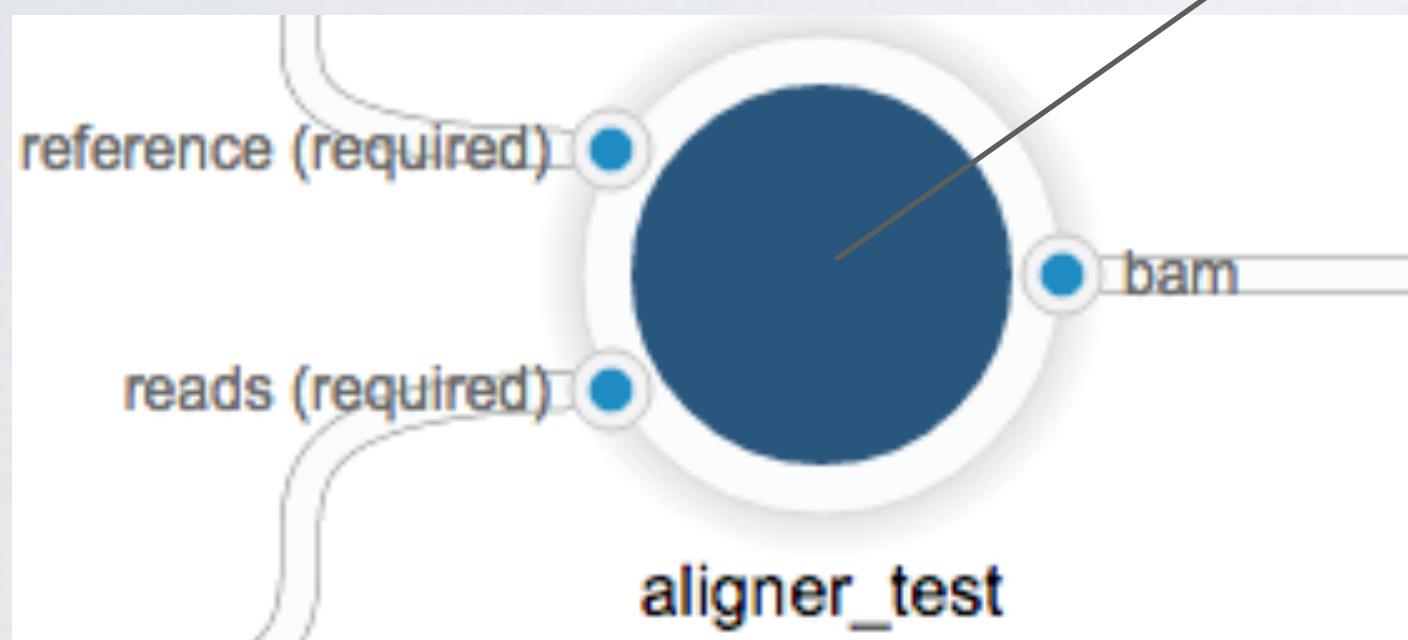
COMMON WORKFLOW LANGUAGE

Specification for describing scalable, portable and reproducible computational workflows.

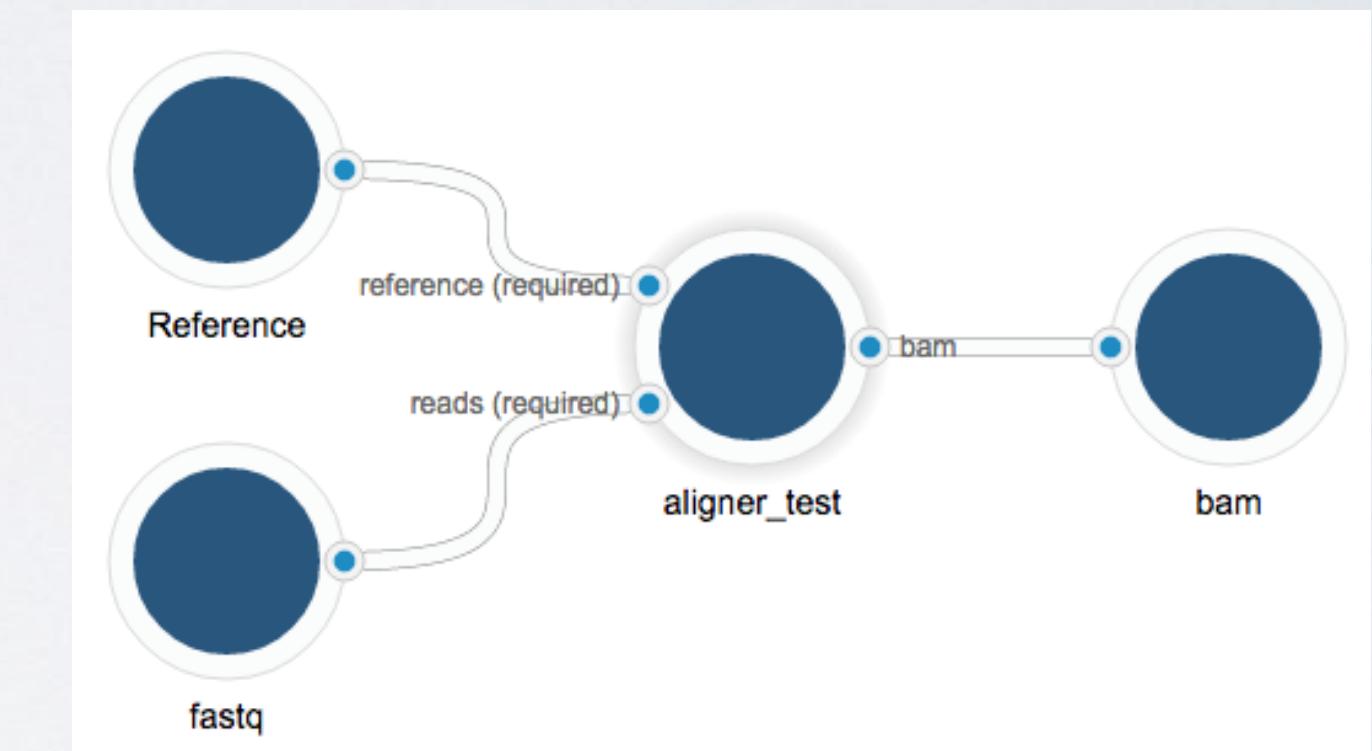
FOR DEVELOPERS, THIS MEANS...

- Easier and faster to deploy your tools.
- Write once, runs everywhere, regardless of operating system/infrastructure.

CWL TOOL: INPUTS, OUTPUTS, AND PARAMETERS



```
"softwareDescription": {  
    "name": "myaligner",  
    "description": "Aligns reads to a reference"  
},  
"documentAuthor": "kaushikghose@sbggenomics.com",  
"requirements": {  
    "environment": {  
        "container": {  
            "type": "docker",  
            "uri": "",  
            "imageId": ""  
        }  
    },  
    "resources": {  
        "cpu": 0,  
        "mem": 5000,  
        "ports": [],  
        "diskSpace": 0,  
        "network": false  
    }  
},
```



WRITE CWL WITH WEB IDE

The screenshot shows a web-based interface for managing Docker tasks. The top navigation bar includes 'Projects', 'Data', 'Apps', and a user dropdown for 'BDAVIS144'. The current view is for the task 'open-whale-say / whalesay / 0'. The interface has tabs for 'GENERAL', 'INPUTS', 'OUTPUTS', 'ADDITIONAL INFO', and 'TEST'. The 'GENERAL' tab is active.

Docker Container

Docker Repository[:Tag]: docker/whalesay:latest

Image ID: (empty input field)

Resources

CPU: 1

Memory (MB): 1000

Command

Base Command: + cowsay

Stdin: Enter value </>

Stdout: output.txt </>

Success Codes: + Click the plus button to add codes

Temporary Fail Codes: + Click the plus button to add codes

Arguments: + Click the plus button to add command line binding.

Resulting command line

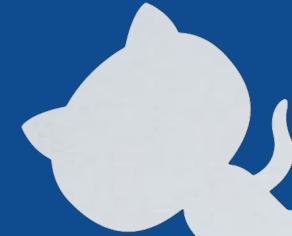
```
cowsay say_what > output.txt
```

Copy button

BUILD CWL TOOLS WITH R

```
# Create CWL Tool
runif = Tool(id      = "runif",
             label    = "runif",
             hints    = requirements(docker(pull = "rocker/r-base")),
             baseCommand = "Rscript runif.R",
             stdout    = "output.txt",
             outputs   = output(id = "random", glob = "*.txt"))

# Convert to JSON or YAML
runif$toJSON()
runif$toYAML()
```

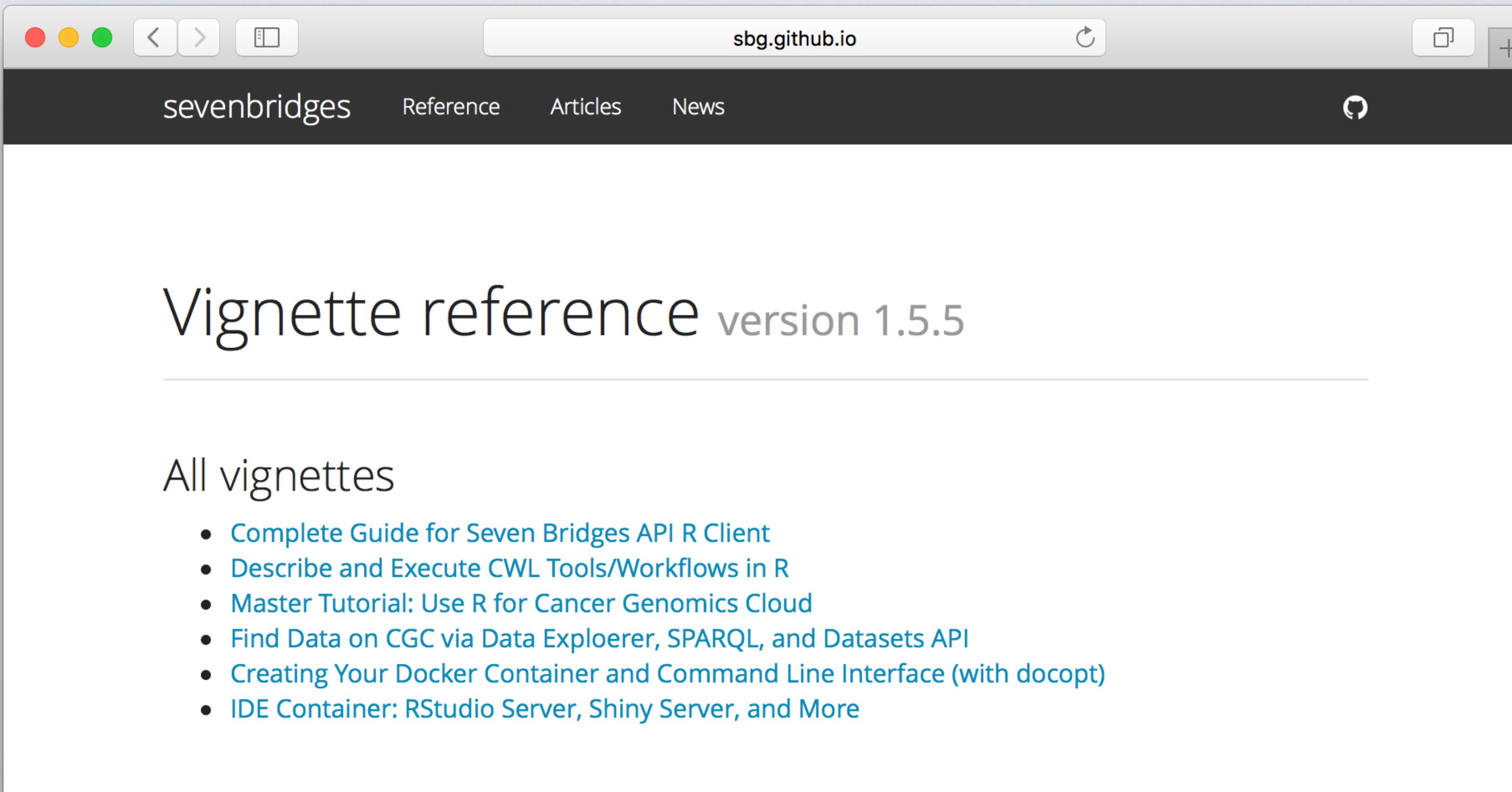


GETTING STARTED

- Try `sevenbridges` package from Bioconductor
- `docker pull sevenbridges/sevenbridges-r`
- Pull requests: github.com/sbg/sevenbridges-r

DOCUMENTATION

<https://sbg.github.io/sevenbridges-r/>



The screenshot shows a web browser window with a dark-themed header bar. The address bar contains the URL "sbg.github.io". Below the header, there is a navigation bar with links for "sevenbridges", "Reference", "Articles", and "News". To the right of the navigation bar is a GitHub icon. The main content area features a large, bold title "Vignette reference version 1.5.5". Below this title is a horizontal line. Under the line, the heading "All vignettes" is displayed, followed by a bulleted list of six items, each with a blue link:

- Complete Guide for Seven Bridges API R Client
- Describe and Execute CWL Tools/Workflows in R
- Master Tutorial: Use R for Cancer Genomics Cloud
- Find Data on CGC via Data Explorer, SPARQL, and Datasets API
- Creating Your Docker Container and Command Line Interface (with docopt)
- IDE Container: RStudio Server, Shiny Server, and More

NEED MORE RESOURCES?

- Apply to NIHCommons Credits Pilot for new opportunities in cloud computing & storage before Jan 16:
- <https://datascience.nih.gov/commons>
- <https://www.commonscredit-portal.org/>

MORE ON CANCER GENOMICS CLOUD AND SEVEN BRIDGES PLATFORMS

- www.cancergenomicscloud.org
- www.sevenbridges.com/tcga
- www.sevenbridges.com

Thank you!

This project has been funded in whole or in part with Federal funds from the National Cancer Institute, National Institutes of Health, Department of Health and Human Services, under Contract No. HHSN261201400008C.

NATIONAL
CANCER
INSTITUTE