Analyzing large scale soil metagenomes using workflows

Folker Meyer, folker@anl.gov

Andreas Wilke, wilke@mcs.anl.gov
William Trimble, trimble@anl.gov

Argonne National Laboratory and University of Chicago

do this now, if you want to follow along later:

- start download of Docker using info from

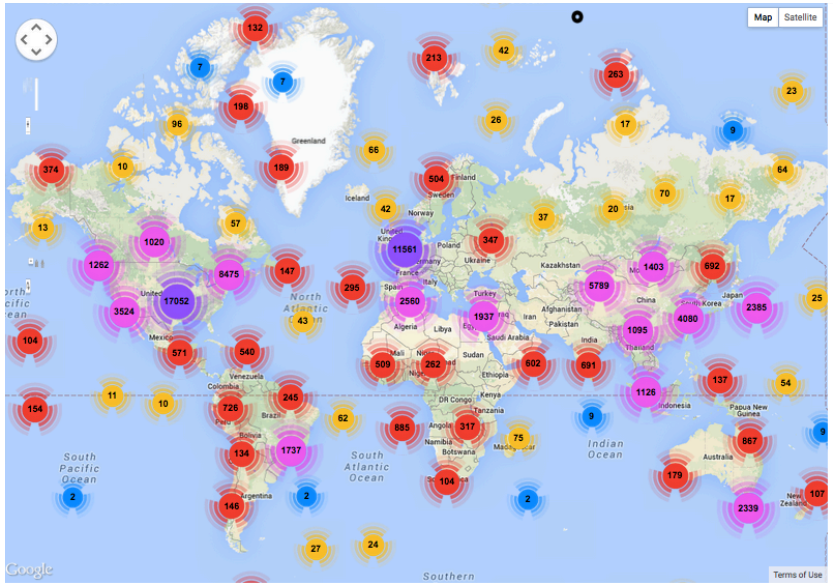
https://bit.ly/2rzqCQg

- BTW all slides are available at: see above
- for a 3 day version: https://github.com/MG-RAST/Skyport2
- all source code is open and available (WiFi might block download via `git`)

Outline for the 2 hours

- Introduction
  - Why?
  - Geek speak explained
  - Overview
- Install worker on your laptop (hands on)
- Introduction into CWL and Docker
- 2 Workflows examples (hands on)
- Results and Outlook

My background

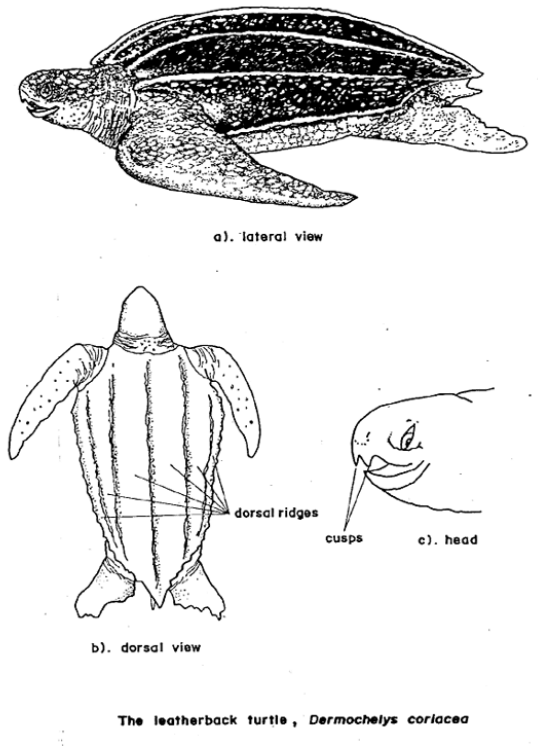


- CS, then large scale bioinformatics and later “cloud”
  - GenDB (Meyer et al, Nuc Acid Res, 2001)
  - MG-RAST (Meyer et al, BMC Bioinformatics, 2008)
  - RAST (Aziz et al, BMC Genomics 2008)
- Distributed execution for thousands of users via interactive web portal
- Portal also does data archival and data integration
- Security is always required, per DOE
- 50k users annually, 30k data submitters
- 400,000 data sets, 200+ Terabases, **>2 Petabyte**

Why all of this?

- data is becoming cheaper and bigger
- analysis is complex
- needs many steps and often many machines for analysis
- building tool chains (aka pipelines, aka workflows) is hard
- what did I run last month?

Biology



http://www.oneocean.org/ambassadors/track_a_turtle/biology

http://www.the-aps.org/education/

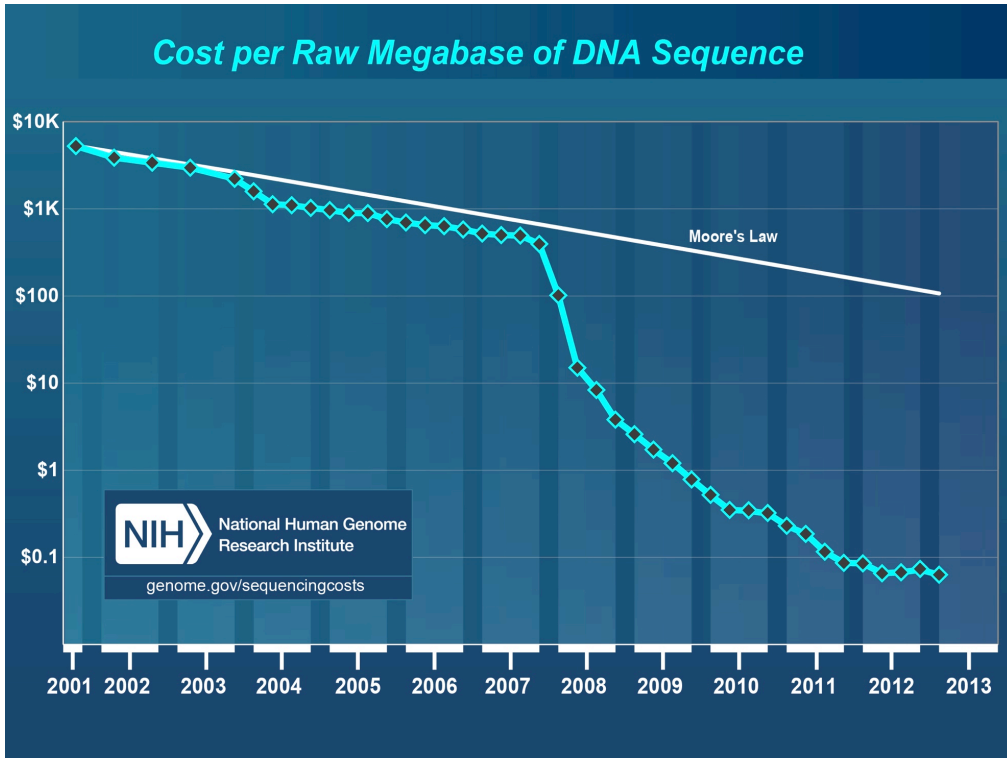
http://www.ferrum.edu/majors/biology.jpg

These are: “`biology.png`”, “`biology.gif`” and “`biology.jpg`”.

Genomics revolution has changed bio-medical research

- Now data and processing is ubiquitous
- Data typical >10GByte per sample
- Prior landscape had genome centers with supercomputers
- Today sequencing is democratized
- You have to provide your own supercomputer
- Computing is bottleneck for many

- Cost is key factor in new ecosystem



Source: NHGRI

$30k sequencing cost ~= 1,000,000 US$ naïve analysis

suggestion: lets not be naive :-)

Geek Speak explained

- Workflow
- CWL
- Container
- Docker
- Singularity
- Provenance
- Reproducibility

Shopping list

To get things done I need lots of moving parts

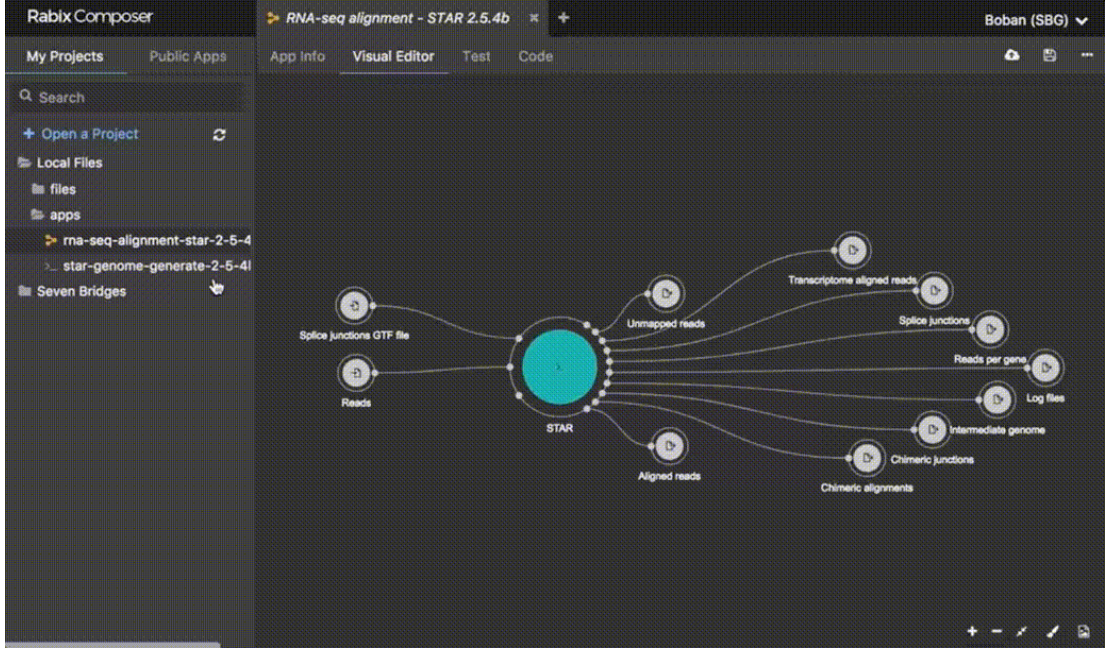
- a place to store intermediate data
- place to store finished data
- interface to control computation
- the software (fancy: execution environment)
- computers

Workflow?

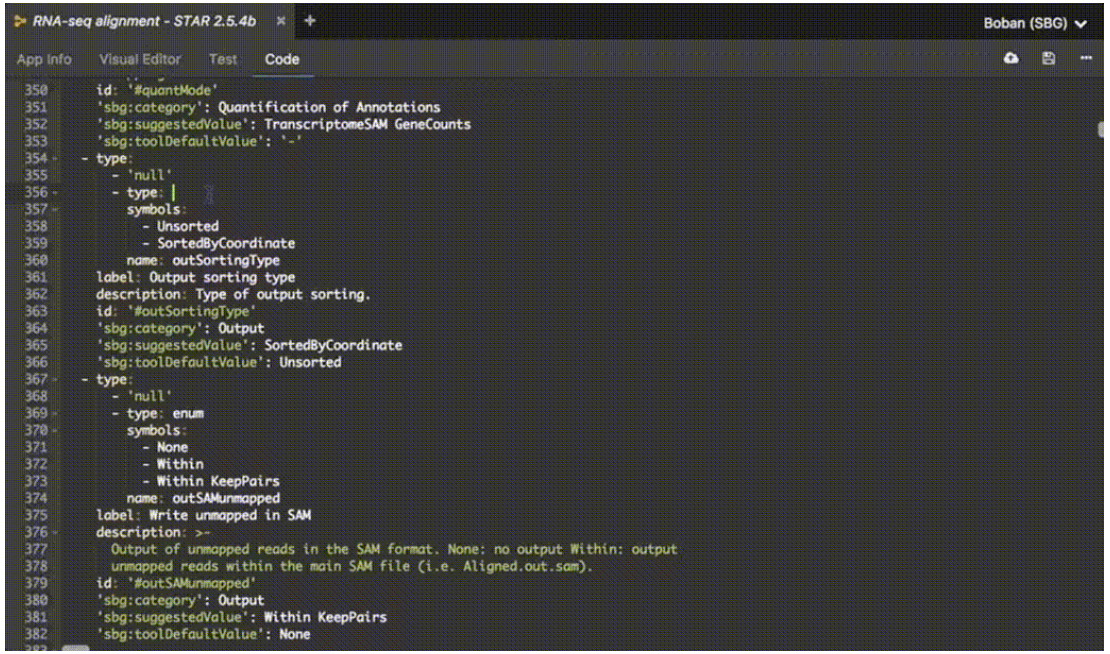
- series of commands (on Linux command line)
- one option: type things in…. (error prone)
- workflow is file with series of steps (no typing)
- many formats exists
- many tools exist
- “players” (workflow manager systems), “editors”, “vizualizers”
- Examples:
  - Galaxy
  - Snakemake
  - CWL

CWL

- **C**ommon **W**orkflow **L**anguage
  - CWL (see https://www.commonwl.org/)
    - “players:” AWE, AirFlow, Xenon, Apache Taverna, Calrissian, cwl-test, …
- CWL is “exchange language” for workflows

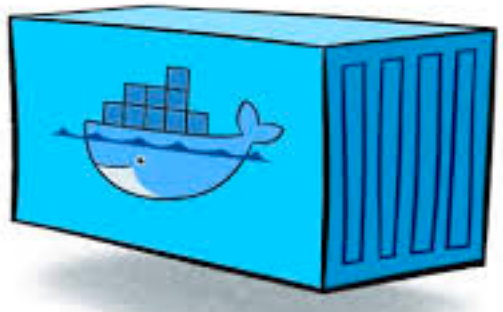


Rabix workflow composer



Containers?

- “Linux containers, in short, contain applications in a way that keep them isolated from the host system that they run on. Containers allow a developer to package up an application with all of the parts it needs, such as libraries and other dependencies, and ship it all out as one package.”
- https://opensource.com/resources/what-are-linux-containers

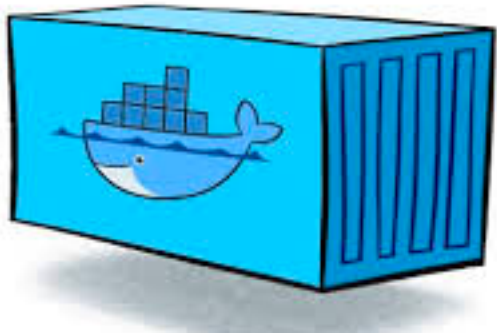


Docker and Singularity

- “players” (like Windows media player) for containers
- installed by Systems admin
- Singularity does NOT require root access to use (only for install)
- allow “installing” complex software with one command:

```
$ docker run -it --rm mysql mysql
-hmy.computer.de -usql-admin -p
```

- will install and run a MySQL database on your computer

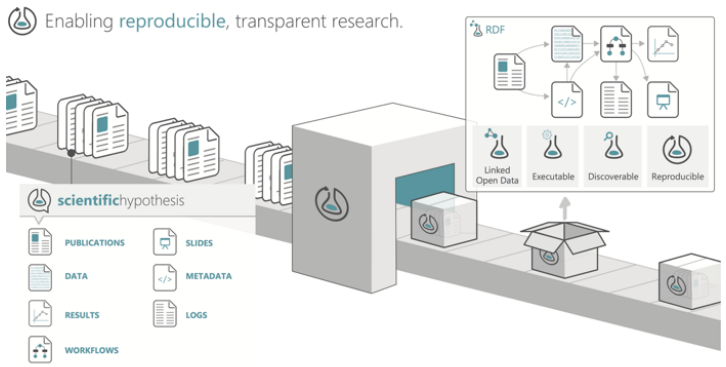


Provenance and Reproducibility

- What was done to compute this?
- Can you run this again?
- Can you run this for my data?

- researchobjects.org (RO) encode provenance

- use CWLprov to produce RO provenance
- CWL allows reproducibility




```
{
  aggregates: [
    {
      bundledAs: {
        folder: "/data/",
        filename: "mgm4441680.3.050.upload.fna"
      },
      mediatype: "text/plain; charset=\"UTF-8\"",
      uri: "http://api-dev.mg-rast.org/download/mgm4441680.3?file=050.1"
    },
    {
      uri: "http://api-dev.mg-rast.org/download/mgm4441680.3?file=100.1",
      mediatype: "text/plain; charset=\"UTF-8\"",
      bundledAs: {
        folder: "/data/",
        filename: "mgm4441680.3.100.preprocess.passed.fna"
      }
    },
    {
      mediatype: "text/plain; charset=\"UTF-8\"",
      uri: "http://api-dev.mg-rast.org/download/mgm4441680.3?file=100.2",
      bundledAs: {
        filename: "mgm4441680.3.100.preprocess.removed.fna",
        folder: "/data/"
      }
    },
    {
      bundledAs: {
        filename: "mgm4441680.3.150.dereplication.passed.fna",
        folder: "/data/"
      },
      mediatype: "text/plain; charset=\"UTF-8\"",
      uri: "http://api-dev.mg-rast.org/download/mgm4441680.3?file=150.1"
    },
    {
      bundledAs: {
        folder: "/data/",
        filename: "mgm4441680.3.150.dereplication.removed.fna"
      },
      mediatype: "text/plain; charset=\"UTF-8\"",
      uri: "http://api-dev.mg-rast.org/download/mgm4441680.3?file=150.2"
```

Why not x?

- x == “shell scripts”
  - Shell scripts are not as portable, require tool installation
- x == “slurm|grid_engine|torque|…”
  - they are fine, but require a lot more more setup
- x == “a shared file system”
  - see “slurm etc”

How can I…?

- Convince my local SysAdmin to install Singularity?
  - I need 15+ tools and approx. 150 libraries installed on MANY machines. When can you install this by? Or would you rather install Singularity for me
- use pre-existing containers?
  - https://hub.docker.com/
- learn about Containers/Docker?
  - use e.g. https://stackify.com/docker-tutorial/
- create my own Docker or Singularity containers?
  - see above

Are there existing workflows?

- MG-RAST
  - shotgun + amplicon annotation and abundance: https://github.com/MG-RAST/pipeline/tree/cwl/4.03
- EBI-MGnify
  - assembly
  - annotation
- QIIME
- MGTap (Martin Hartmann's high precision workflow for amplicons)

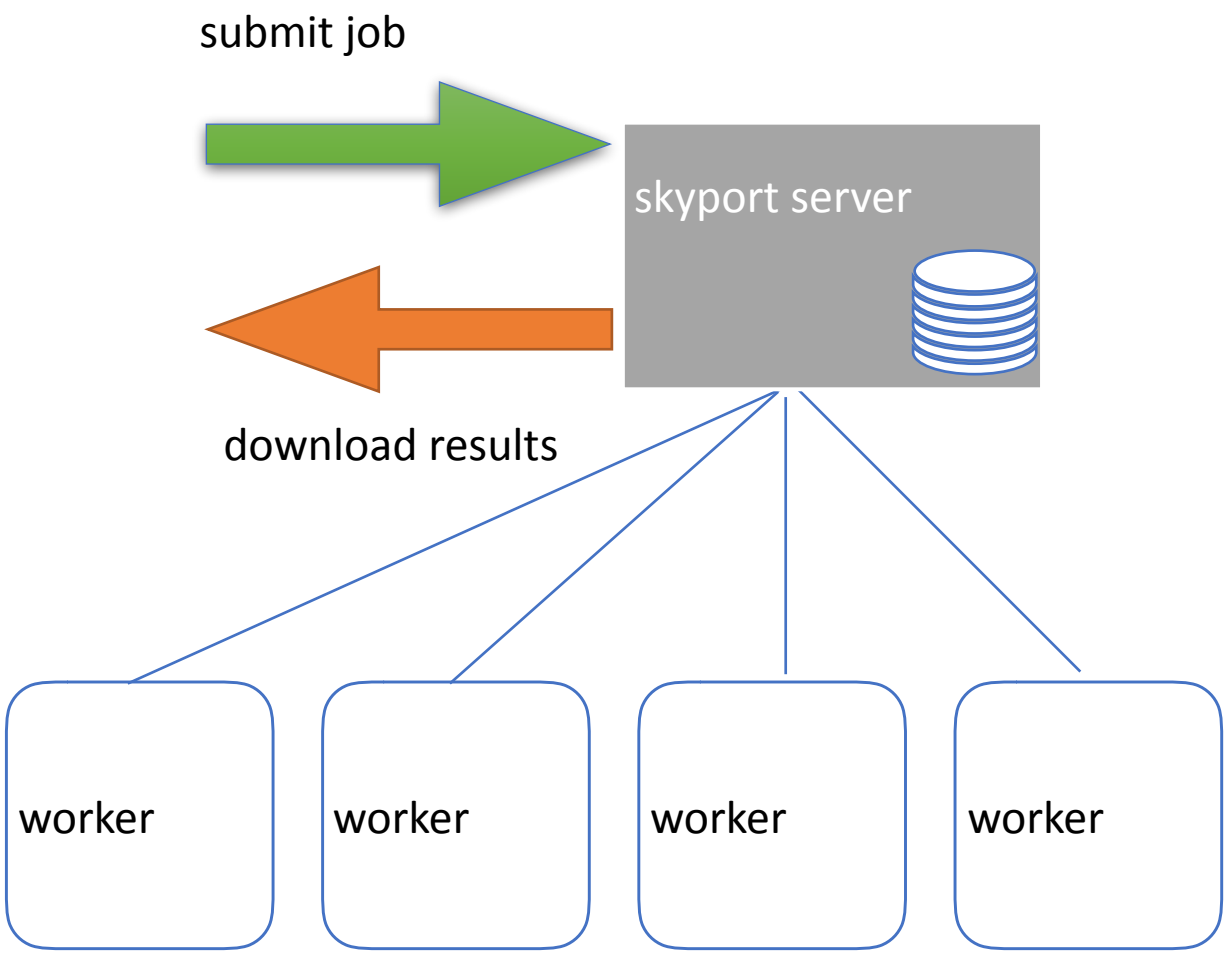
What do I have to do to run those workflows?

Answer:

- part A: the rest of this tutorial
- part B: you might have to work with your SysAdmin or rent machines on e.g. Amazon EC2 (AWS)

What does the end-state look like?

1. submit job with data to server
2. server with distribute data and work to “workers”
3. add as many as you can
4. wait for compute
5. download results



Handover

next bit: install a Docker on your laptop