A reproducible data analysis environment for next-generation sequencing on public cloud computer

Manabu Ishii, Akihiro Matsushima, Mika Yoshimura, Hiroki Danno, Itoshi Nikaido

Bioinformatics Research Unit, Advanced Center for Computing and Communication, RIKEN Japan
Outline

• Abstract

• How do we prepare NGS analysis environment

• Construction of reproducible and flexible NGS analysis environment

• Example: Construct NGS analysis environment

• Result/Future
Abstract

• With the progress of DNA sequencing methods, it continues to increase a quantity of data and type of data to be produced. To analyze such data, we need massive computer resources and setup of various software and databases. Many data-analysis techniques and databases are constantly developed. Accordingly, it takes plenty of time and works to construct an analysis environment, such as procurement of computers, installation of software, and construction of data analysis pipelines.
Problem Background

• Researchers MUST be concentrate research.
  • Our main target of this project is Biologists, Researchers who do not know about setup the computer.

• A lot of problem when they want to have a cluster computer system even if they have enough budget
  • Electric Power, Space
  • Computer (Not only CPU, memory, but also consumed power, weight, expandability),
  • Create specification
  • Contact with email …
  • Operation, Plan of backup, Trouble shoot…

• These are not research, this is engineering
How do we prepare NGS analysis environment?

• Computers are arrived but so many things to do

• Before starting NGS analysis environment
  • OS install
  • Network settings
  • Middleware settings
  • Monitoring setups

• After that we start to setup analysis environment
Setup analysis environment

- We must prepare software and database including its version

Requests from users

• Many data-analysis techniques and databases are constantly developed

• System administrator gets a lots of requests from users
  
  • “I want to use this software. It requires this software and need to upgrade that library”

• Problem

  • It’s hard not to break current environment.

  • System admins want to create good answer for the requests but …
Public cloud as solution

- Anyone Anytime Any computer constructs same analysis environment

- Proposed method

```bash
azure group deployment create -g AZURERESOURCEGROUP6 \
  -n Deploy1 -f azuredeploy.json -e local.parameters.json -v
```

1 click

NGS analysis environment
Construction of reproducible and flexible NGS analysis environment

• What is important point?
  • Reproducibility
    • Anyone, Anytime, Any computer, always construct same software and computing environment
  • Flexibility
    • Easy to add tools
    • Easy to change versions
Public Cloud System (Any Computer)

- Microsoft Azure
  - We construct Linux machines on Azure.
- Azure Resource Manager
  - Define number of virtual machines, networking and other things
  - JSON format
  - Resources are available on the GitHub
  - Specification, Documentation
  - A lot of Sample codes
- Problem
  - Problems are ....

```json
"vmSize": {
  "type": "string",
  "defaultValue": "Standard_D"
  "metadata": {
    "description": "Size of t"
  }
},

"nodeExecutionCount": {
  "type": "int",
  "defaultValue": 2,
  "metadata": {
    "description": "Number of"
  }
},
```
Infrastructure as Code (Anyone, Anytime)

• Chef
  • Automate IT Infrastructure
• Cookbook and recipe
  • Source code
• Why do we choose chef?
  • A lot of cookbooks
  • Test kitchen
    • testing frame work for chef
    • a test harness to execute infrastructure on one or more platforms
Docker (Anytime, Anyone)

- Good point
  - Everything inside container.
  - Fixed tool version by tag

- Potentially Problem
  - Security
  - Tag is just ‘latest’
  - Same Tag but updated …
Continuous Integration (Anytime)

- Continuous Integration
  - We MUST confirm the analysis environment is what we expected after we constructed it.

- Jenkins and Selenium

- Why do we need test?
  - Sometimes using same code but different environment created.
  - There are a lot of RISK that people don't considered deeply.
    - Security update changes expected behavior.
    - Some dependency's's's updated its make Dependency Hell.
    - Some download sites are move, down, disappeared …
Example: Construct NGS analysis environment

- We construct 2 type of pipeline
  - Galaxy
    - RNA-Seq pipeline
    - All in one docker container
  - Command Line
    - RNA-Seq pipeline
    - Multiple-containers
- Job scheduler
  - All execution nodes are automatically up and down
Galaxy Pipeline

- 15 pipelines in Galaxy
- Docker Galaxy everything inside
- Job scheduler is Sun Grid Engine
  - outside Docker
Galaxy Pipeline Test

Basic Iteration
Run -> Wait -> Check

1. WF-01
2. WF-02
3. WF-03
**Command Line Pipeline**
**Single Cell RNA-seq Pipeline Using Docker**

*This pipeline is used for next slide ‘compare’*

- **IMAGE**
  - bcl2fastq2:1.0
  - fastx_toolkit
  - fastq:1.0
  - dropseq:1.0
  - star2.5.1b:1.0
  - star2.5.1a:1.0
  - pyper:1.2

- **CMD**
  - bcl2fastq
  - fastqc
  - fastx_trimmer
  - FastqToSam
  - TrimStartingSequence
  - SamToFastq
  - STAR
  - SortSam
  - MergeBamAlignment
  - TagBamWithReadSequenceExtended
  - TagReadWithGeneExon
  - correct_barcode.py
  - DigitalExpression
  - analog_expression.py
  - BAMTagHistogram

- **FileSystem**
  - 460,000+ Dockerized Applications

- **Execution Nodes**

- **Submission Node**

- **GRID ENGINE**

- **docker run**

- **docker push**

- **bash**

- **CMD**

- **FileSystem**

- **Execution Nodes**

- **login**

- **qsub**

- **/qdel**

- **/qmod -s**

- **/qmod -us**

- **NFS**

- **docker pull**

- **docker push**

- **send**
**Compare**

<table>
<thead>
<tr>
<th>Cost</th>
<th>Execution Time</th>
<th>Procurement Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cost</td>
<td>Procurement Time</td>
</tr>
<tr>
<td>MacBookPro</td>
<td>$3000</td>
<td>MacBookPro</td>
</tr>
<tr>
<td>On-premise*1,*2</td>
<td>$50000 ~</td>
<td>On-premise</td>
</tr>
<tr>
<td>Proposed Method *3</td>
<td>$200/run</td>
<td>Proposed Method</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reproducibility</th>
<th>Execution Time</th>
<th>Procurement Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacBookPro</td>
<td>No - if manually</td>
<td>MacBookPro</td>
</tr>
<tr>
<td>On-premise</td>
<td>Hard - procedure by specification</td>
<td>On-premise</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>YES</td>
<td>Proposed Method</td>
</tr>
</tbody>
</table>

*1 Prepare own computers  
*2 Assume Government procurement  
*3 Just 1 time analysis cost

This compare is done by Previous Slide. It is not Galaxy Pipeline.  
This compare is illumine NextSeq500 1 run, almost 2000 single-cell RNA-seq.  
And compare only computational analysis.
Result

• Researchers are released from Engineering

• System Administrator concentrate maintenance without breaking current analysis environment

• Construct NGS analysis environment for both web browser and command line.

• Construct NGS analysis environment very faster and cheaper than on-premise same computing resources
Future

• We want to change virtual machine size dynamically and timing of start and stop of virtual machine for more cost effective.
• Integrated or contribute to other tools
• kubernetes

• **Avoid vendor lock in** and more cost effective, We prepare for moving to other public cloud service

![AWS](image.png)

![CloudStack](image.png)

![OpenStack](image.png)
Acknowledgement

• Bioinformatics Research Unit, Advanced Center for Computing and Communication, RIKEN

• Microsoft Japan

• https://github.com/manabuushii/GCC2017