Read Between the Lines: Closing Gaps of Materials and Methods to Build Workflow from the Publication

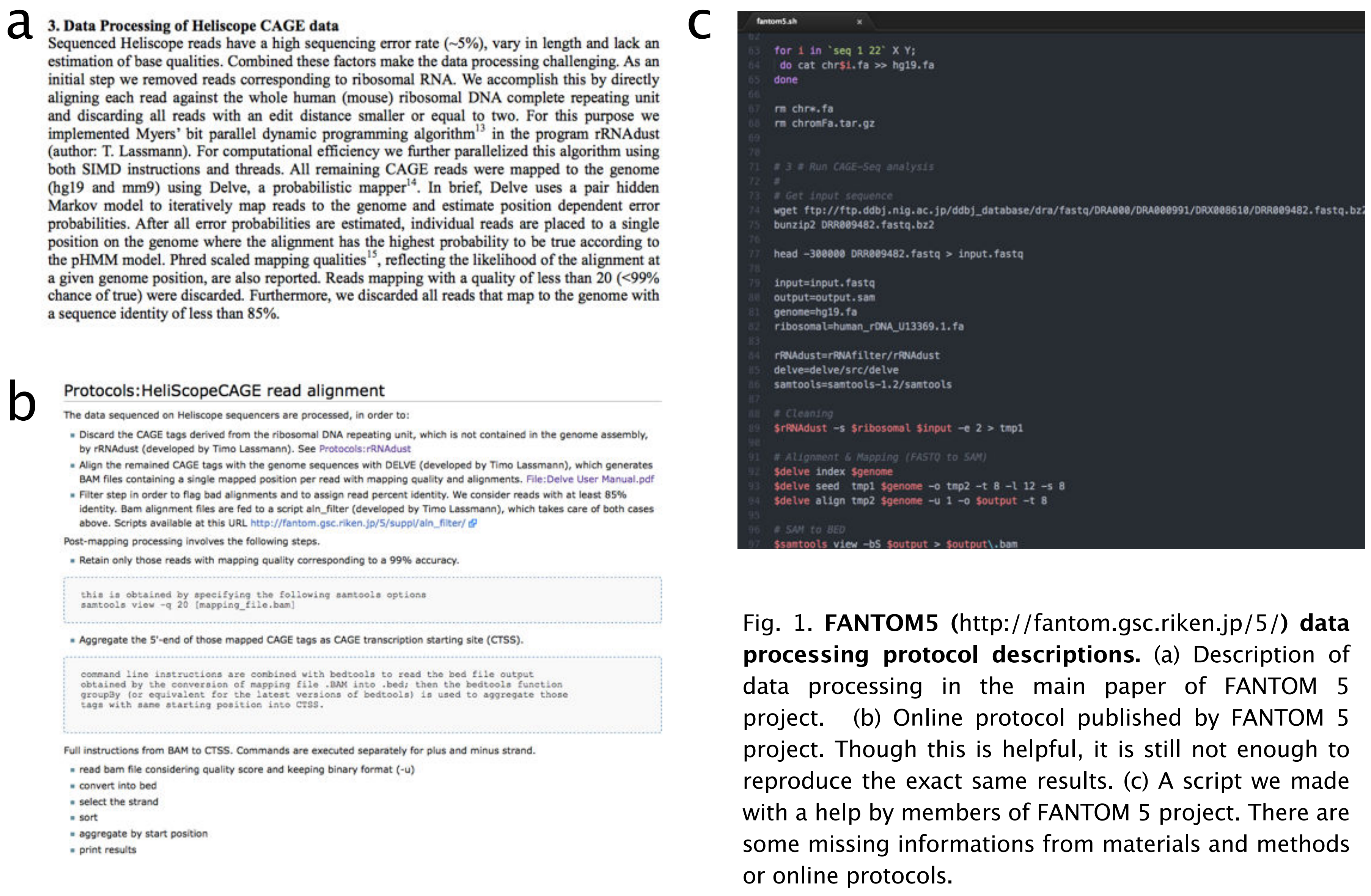
Tazro Ohta[1], Osamu Ogasawara[2], Yoshinobu Masatani[3], Shigetoshi Yokoyama[3], Kento Aida[3]

1. Database Center for Life Science, Chiba, Japan 2. DNA DataBank of Japan, Shizuoka, Japan, 3. National Institute of Informatics, Tokyo, Japan

Background

Increasing cost of translation from text to executable scripts

Publishing and sharing data analysis workflow using the galaxy platform has spectacularly reduced the cost of reproducing one's research, but following the description of data analysis which had been performed by other researchers to get the exact same result is still a big challenge. To evaluate the cost of data analysis workflow from the natural language description, we have performed to rebuild the workflow of CAGE sequencing data processing done by FANTOM5 team on the galaxy platform. Though the project has already published a set of papers with a lot of supplementary of methods and online protocols, it was not that straightforward to get the same result from the raw sequencing data available in the public data repository. The results processed by the rebuilt workflow are compared with the results published online by FANTOM5 team. This case study showed that some of the important information to rebuild the workflow is missing even in the well-described documents, for example, the location of the older source code, or the parameters for command execution. As the speed of biological data production increases, it will be more important to build the framework of cost-effective research reproducibility such as an automated evaluation process of published workflow. We will provide the details of our case study, and discuss how we can assure the reproducibility with the galaxy and other possible ways to perform, share, and publish the workflow as it is "executable materials and methods".



a

3. Data Processing of Heliscope CAGE data

Sequenced Heliscope reads have a high sequencing error rate (~5%), vary in length and lack an estimation of base qualities. Combined these factors make the data processing challenging. As an initial step we removed reads corresponding to ribosomal RNA. We accomplish this by directly aligning each read against the whole human (mouse) ribosomal DNA complete repeating unit and discarding all reads with an edit distance smaller or equal to two. For this purpose we implemented Myers' bit parallel dynamic programming algorithm[13] in the program rRNAdust (author: T. Lassmann). For computational efficiency we further parallelized this algorithm using both SIMD instructions and threads. All remaining CAGE reads were mapped to the genome (hg19 and mm9) using Delve, a probabilistic mapper[14]. In brief, Delve uses a pair hidden Markov model to iteratively map reads to the genome and estimate position dependent error probabilities. After all error probabilities are estimated, individual reads are placed to a single position on the genome where the alignment has the highest probability to be true according to the pHMM model. Phred scaled mapping qualities[15], reflecting the likelihood of the alignment at a given genome position, are also reported. Reads mapping with a quality of less than 20 (<99% chance of true) were discarded. Furthermore, we discarded all reads that map to the genome with a sequence identity of less than 85%.

b

Protocols:HeliScopeCAGE read alignment

The data sequenced on Heliscope sequencers are processed, in order to:

- Discard the CAGE tags derived from the ribosomal DNA repeating unit, which is not contained in the genome assembly, by rRNAdust (developed by Timo Lassmann). See Protocols:rRNAdust
- Align the remained CAGE tags with the genome sequences with DELVE (developed by Timo Lassmann), which generates BAM files containing a single mapped position per read with mapping quality and alignments. File:Delve User Manual.pdf
- Filter step in order to flag bad alignments and to assign read percent identity. We consider reads with at least 85% identity. Bam alignment files are fed to a script aln_filter (developed by Timo Lassmann), which takes care of both cases above. Scripts available at this URL http://fantom.gsc.riken.jp/5/suppl/aln_filter/

Post-mapping processing involves the following steps.

- Retain only those reads with mapping quality corresponding to a 99% accuracy.

```
this is obtained by specifying the following samtools options
samtools view -q 20 [mapping_file.bam]
```

- Aggregate the 5'-end of those mapped CAGE tags as CAGE transcription starting site (CTSS).

```
command line instructions are combined with bedtools to read the bed file output
obtained by the conversion of mapping file .BAM into .bed; then the bedtools function
groupBy (or equivalent for the latest versions of bedtools) is used to aggregate those
tags with same starting position into CTSS.
```

Full instructions from BAM to CTSS. Commands are executed separately for plus and minus strand.

- read bam file considering quality score and keeping binary format (-u)
- convert into bed
- select the strand
- sort
- aggregate by start position
- print results

c

fantom5.sh

```
for i in `seq 1 22` X Y;
 do cat chr$i.fa >> hg19.fa
done

rm chr*.fa
rm chromFa.tar.gz


# 3 # Run CAGE-Seq analysis
#
# Get input sequence
wget ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra/fastq/DRA000/DRA000991/DRX008610/DRR009482.fastq.bz2
bunzip2 DRR009482.fastq.bz2

head -300000 DRR009482.fastq > input.fastq

input=input.fastq
output=output.sam
genome=hg19.fa
ribosomal=human_rDNA_U13369.1.fa

rRNAdust=rRNAfilter/rRNAdust
delve=delve/src/delve
samtools=samtools-1.2/samtools

# Cleaning
$rRNAdust -s $ribosomal $input -e 2 > tmp1

# Alignment & Mapping (FASTQ to SAM)
$delve index $genome
$delve seed  tmp1 $genome -o tmp2 -t 8 -l 12 -s 8
$delve align tmp2 $genome -u 1 -o $output -t 8

# SAM to BED
$samtools view -bS $output > $output\.bam
```

Fig. 1. **FANTOM5 (**http://fantom.gsc.riken.jp/5/**) data processing protocol descriptions.** (a) Description of data processing in the main paper of FANTOM 5 project. (b) Online protocol published by FANTOM 5 project. Though this is helpful, it is still not enough to reproduce the exact same results. (c) A script we made with a help by members of FANTOM 5 project. There are some missing informations from materials and methods or online protocols.

Galaxy as an inter-laboratory workflow sharing platform

workflow runs on any galaxy, and the galaxy runs anywhere

Application: Community 'Pitagora' Galaxy VM

We have been trying to develop the system that we are able to share community's workflows developed and used in the various studies which deal with NGS data, such as Exome-Seq, RNA-Seq, ChIP-Seq, or Bisulfite-Seq. We developed Virtual appliance that includes workflows and some tools to help execution of the workflows, and the Virtualbox image and Amazon Machine Image are distributed. We also maintain public server for the test use, and documents and test data are hosted on our project website.

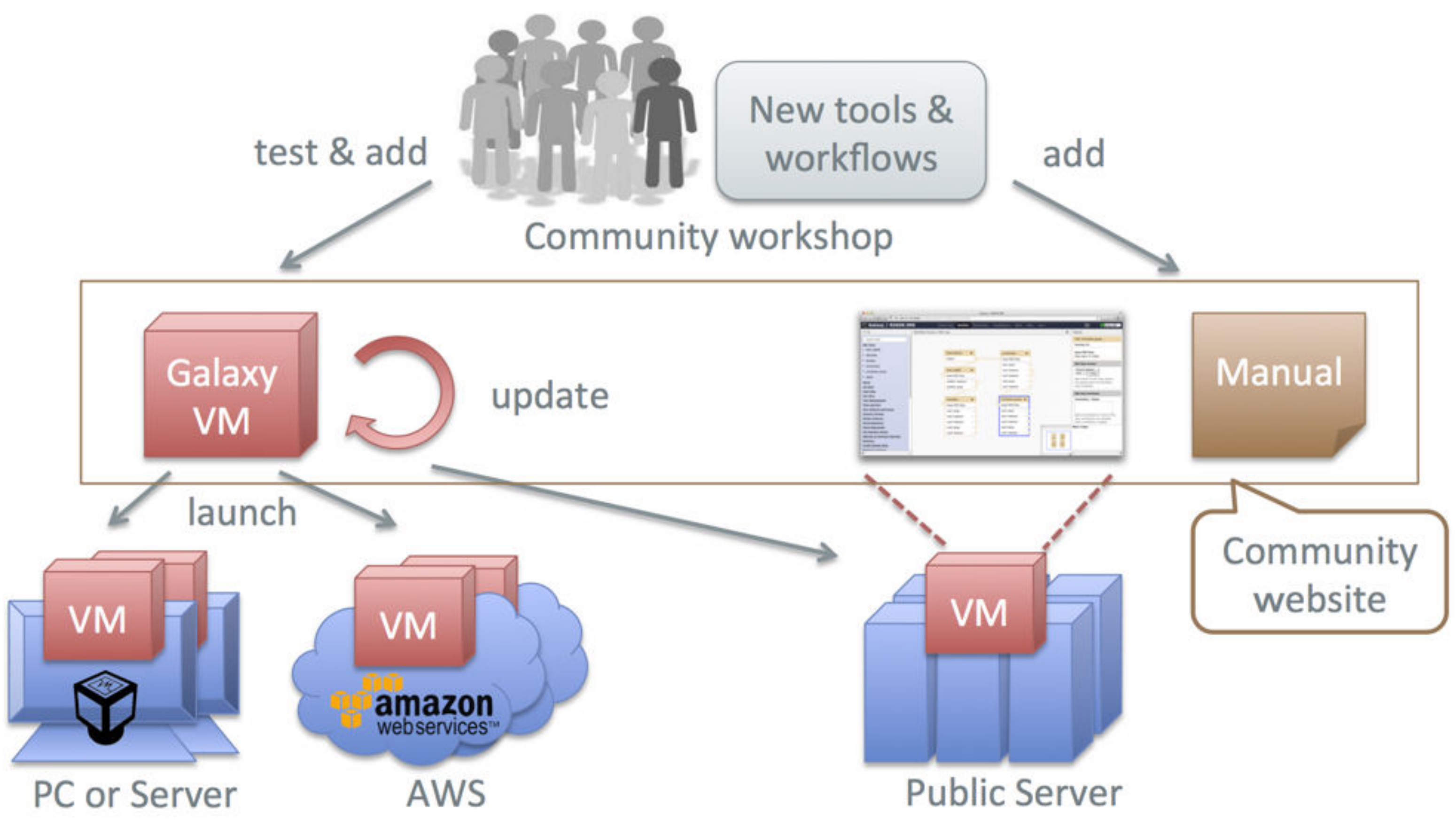
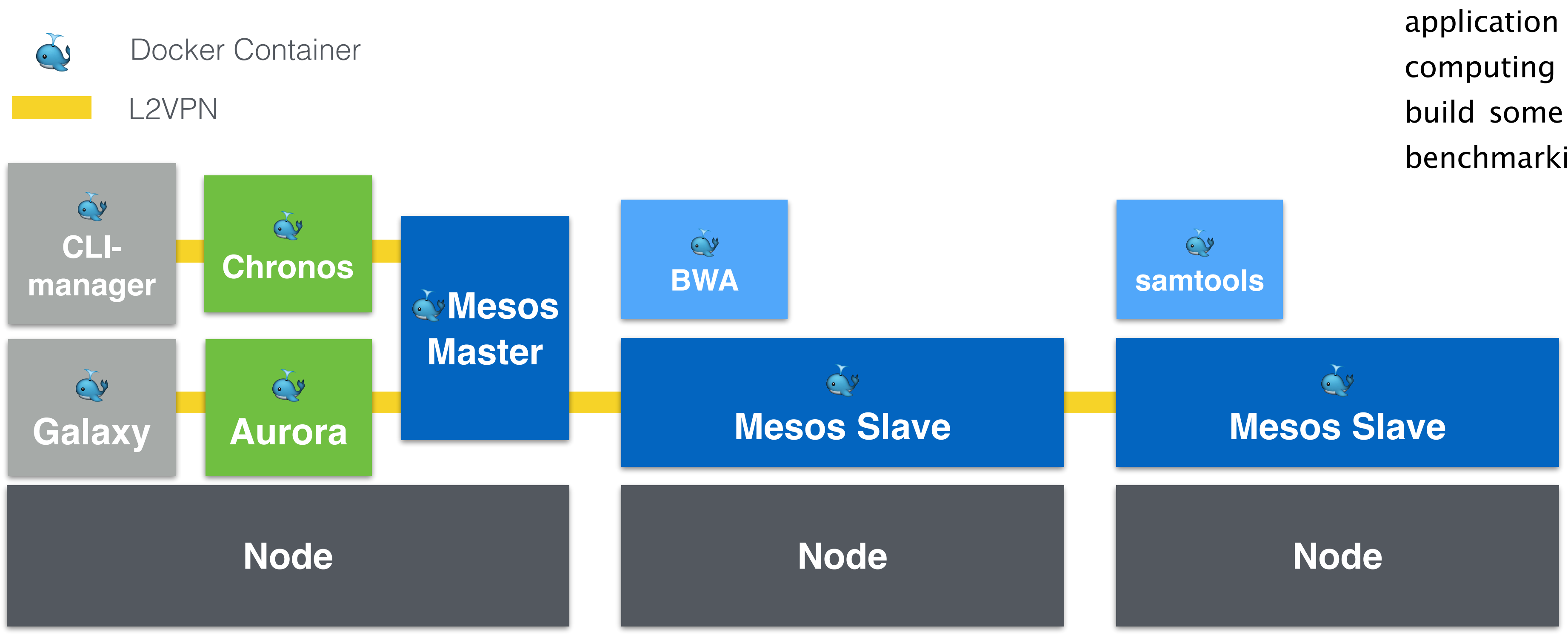


Fig. 2. **Community Galaxy VM and Galaxty Community Japan.** We have been organizing Japanese local community of the galaxy developers and users. The community is sharing the tools and workflows that are used in their labs. Workflows are implemented and fully documented online with the test data. Automated test of the workflow is being developed. Packed VM is distributed as a normal Virtualbox image and Amazon Machine Image via Amazon Web Service. The community is also hosting public server (listed on the page of galaxy project 'virtual appliance').

Platform: 'Overlay Cloud' datacenters

"Overlay Cloud" project led by National Institute of Informatics (Tokyo, Japan) is building large scale cloud computing infrastructure that can distribute computing tasks via their high-speed network system "sinet". We are testing to build galaxy environment on their cloud system, which runs all the process or application as a docker container. Apache Mesos running over the nodes and datacenter is managing computing resource, and galaxy runs on it as a container, then run tools as a docker container. We have build some tool.conf and job runner for containerized tools and are still being tested and under the benchmarking.



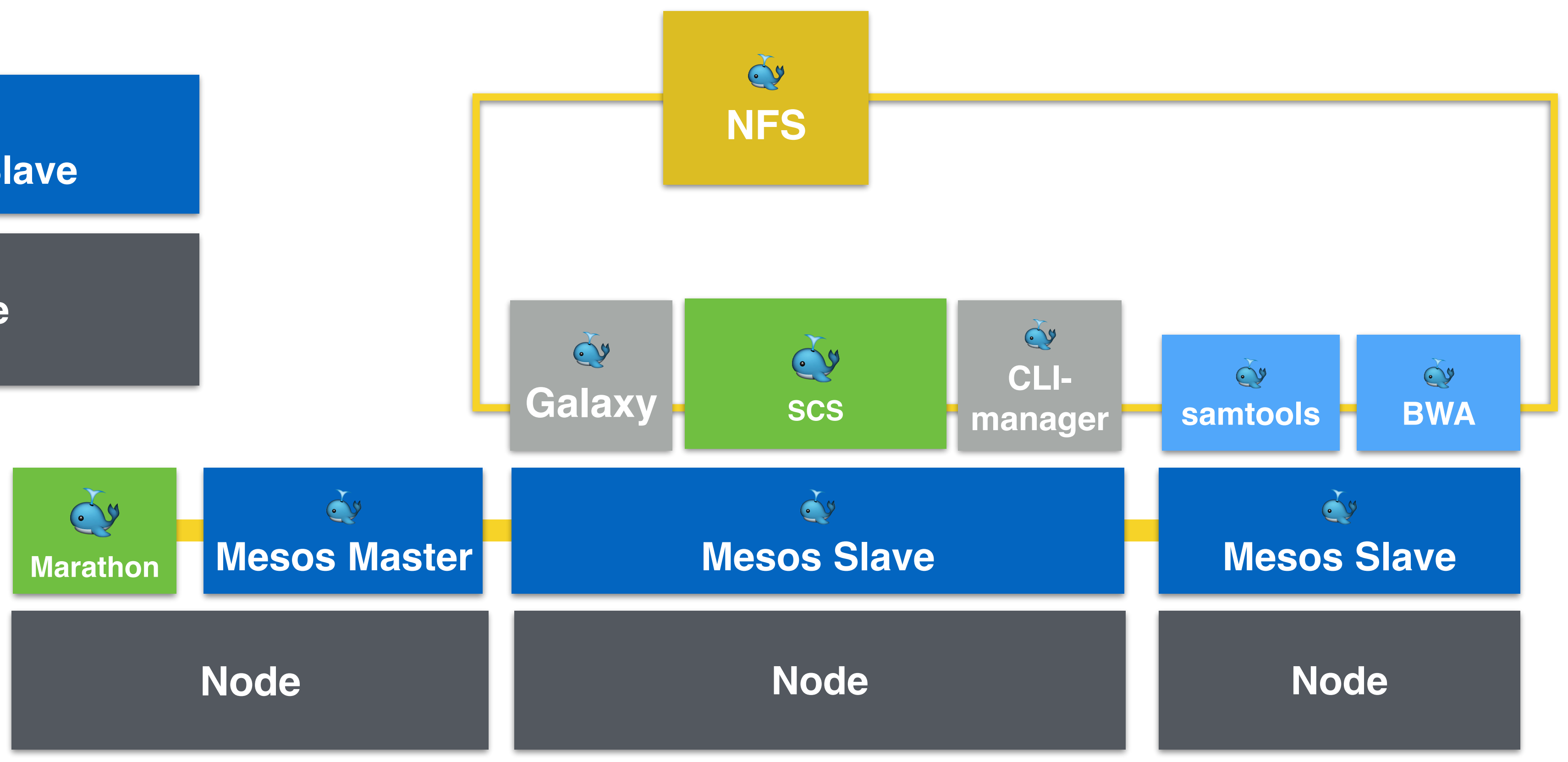


Fig. 3. **Overlay cloud system and container-based galaxy and galaxy tools.** (top-left) Current system which is fully containerized computing infrastructure for the galaxy. Each application runs as a container and connected by L2VPN. Data are stored in the NFS mounted on each node. (bottom-right) Current idea of ver. 2.0 system. Apache mesos and marathon container are detached from the network of the containers of galaxy and galaxy tools which are attached to the NFS via L2VPN. This enables more portability to the running system.

Challenges

- Performance Comparison and benchmarking
- Connecting published workflows to the publication
    - Toolshed or other data repository
- Standard description or annotation of workflows
- Data transfer via internet
    - burst buffer like cloud volume?