Providing reproducible user experience using Docker containers

Bootcamp on Reproducible Research

6/7/19

Outline

1. **Why make your users do more work than they should?**
2. **What are micro services?**
3. **Micro services for reproducibility**
4. **How does Docker solve this problem?**
5. **Code Walkthrough**

Reviewer Comments

Software Requirements

1. python – Version 2.7
2. R – Version 3.4.4
3. Required python packages can be installed using `pip install python_packages_prereq.txt`
4. Required R packages – sqldf (v0.4.11), reshape2 (v1.4.3).
5. GATK (v3.5)
6. bedtools (v2.0)
7. samtools (v1.3.1)

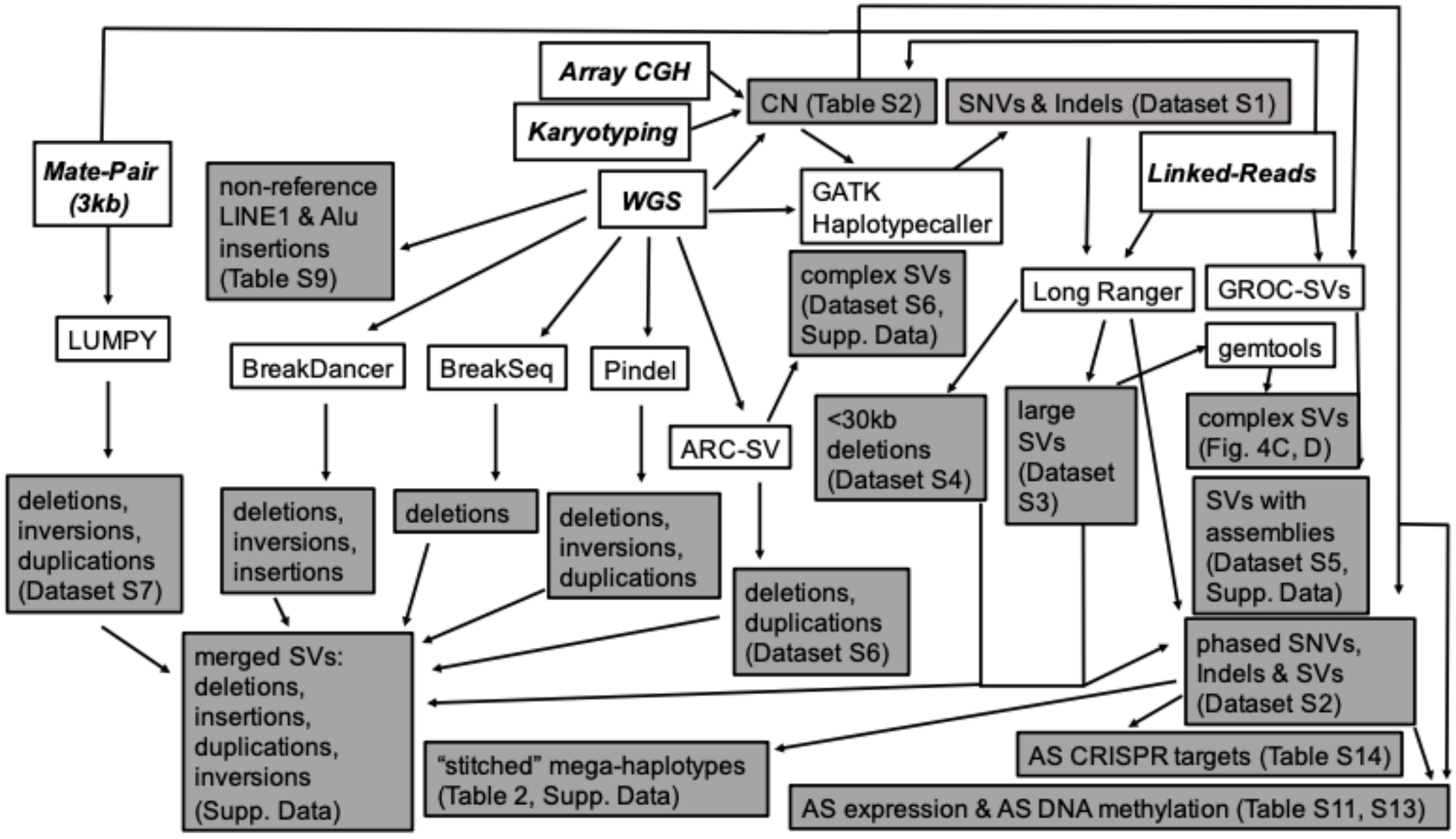
Reviewer 1

Authors should consider providing a Docker image containing all software needed, such as an appropriate version of python, R, GATK, bedtools, samtools as well as CNV callers (CANOES, CODEX, XHMM, CLAMMS). The above-mentioned software tools should be pre-installed and configured in the Docker image. Otherwise the installation and configuration process is simply impractical for external users.

Reviewer 2

There is also a practical issue. Since four specific tools were used build the prediction model, end-users will need to run these four tools before CN-learn can be applied. Authors are strongly encouraged to provide an user-friendly pipeline that can streamline the CNV calling process to generate input data for CN-learn.

Insane number of tools used in analysis pipelines



https://genome.cshlp.org/content/suppl/2019/02/08/gr.234948.118.DC1/Supplemental_Material_1.pdf

Why make your users do more work than they should?

Software Requirements

1. python – Version 2.7
2. R – Version 3.4.4
3. Required python packages can be installed using `pip install`
4. Required R packages – sqldf (v0.4.11), reshape2 (v1.4.3).
5. GATK (v3.5)
6. bedtools (v2.0)
7. samtools (v1.3.1)

Docker

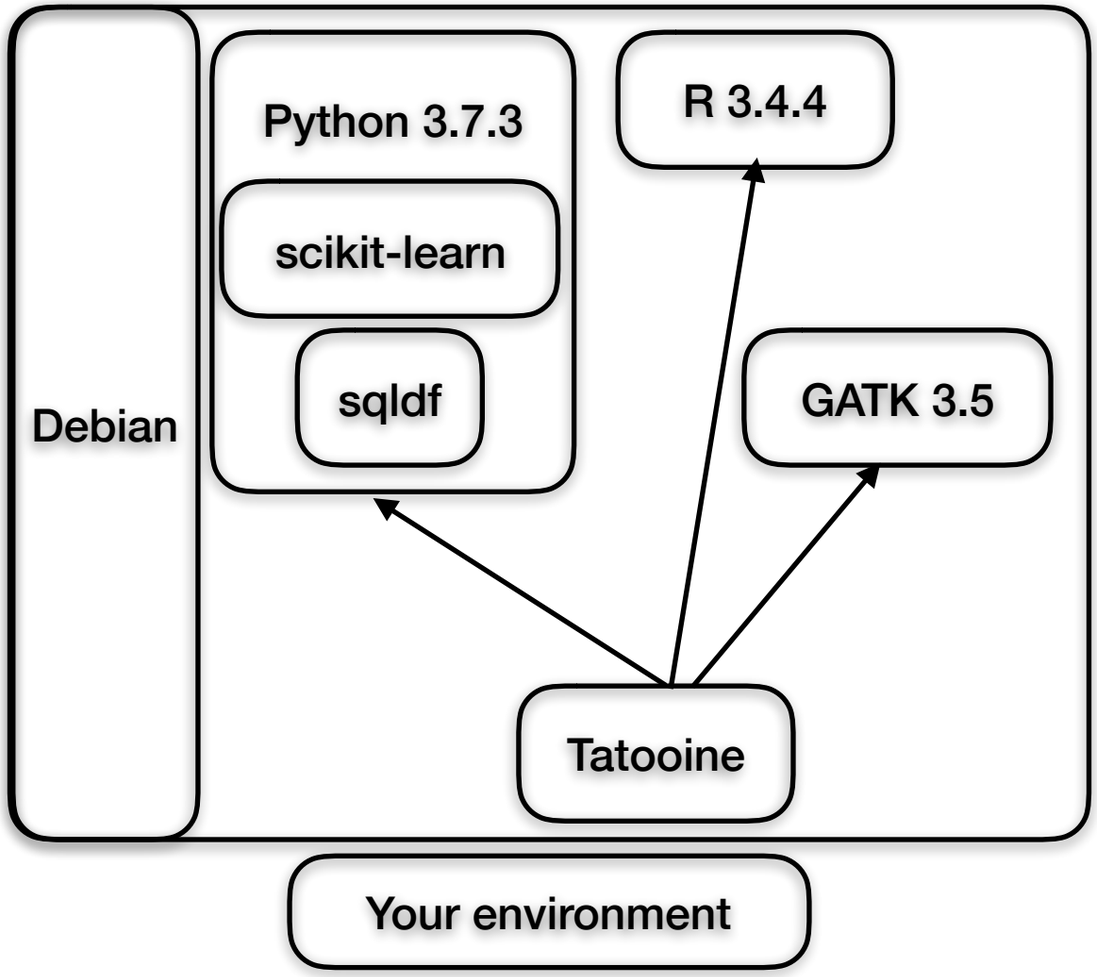
Following are some of the software tools preinstalled in the docker image,

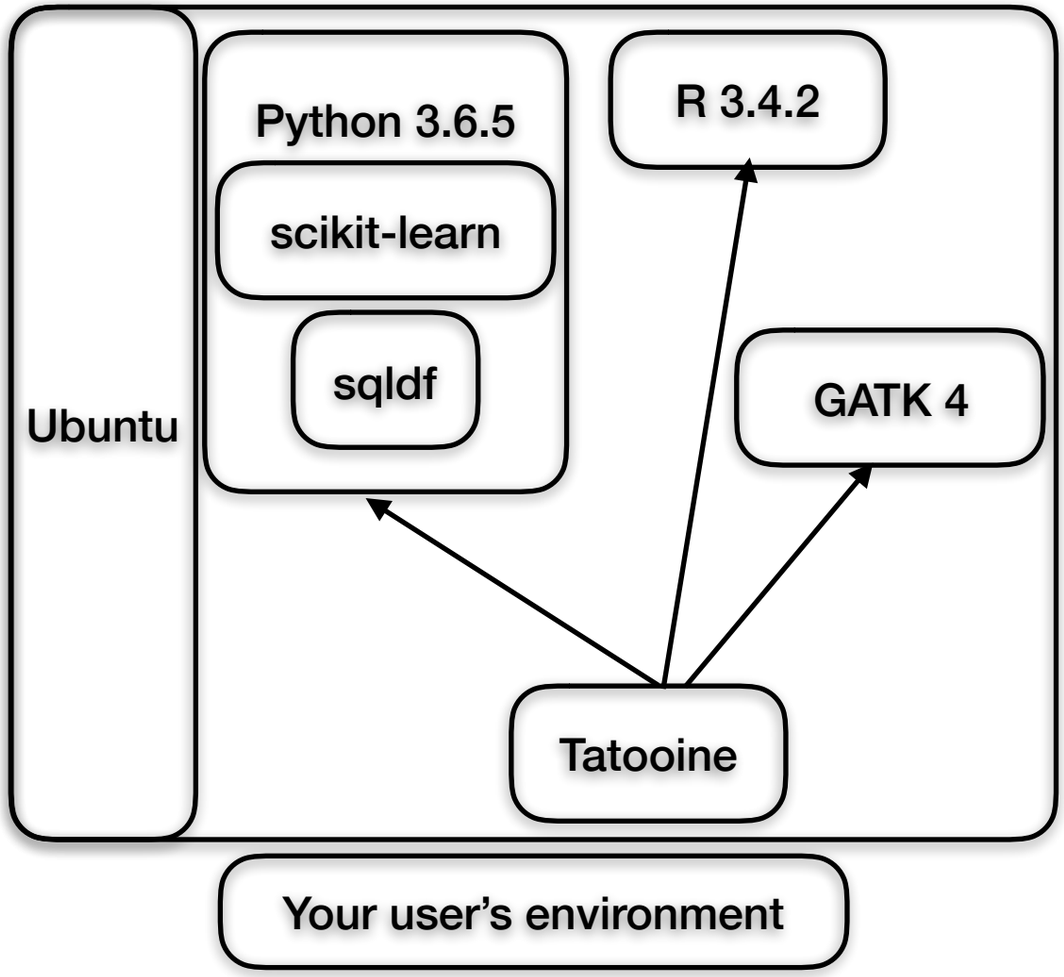
1. Python 3.7.3
2. R 3.4.4
3. Java 8
4. GATK 3.5
5. bedtools 2.27.1
6. samtools 1.3.1
7. CANOES, CODEX, CLAMMS, XHMM & CN-Learn

The complete list of preinstalled softwares can be found in the Dockerfile.

https://consumergoods.com/amazon-go-chicago
https://en.wikipedia.org/wiki/Self-checkout

Pitfalls of making your users do your job





1) Users often have compute environments with different linux distributions/versions

2) Users might have a version of software (python, R etc) that is very close to the one you recommend

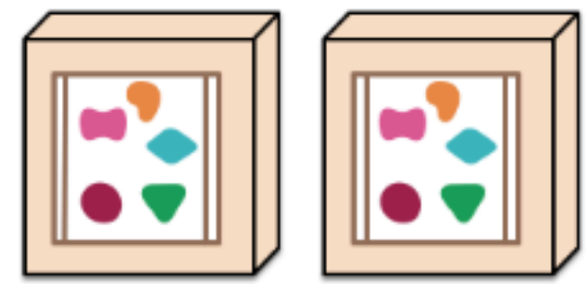
3) You will be contacted even when issues stem from version dependency issues

https://genome.cshlp.org/content/suppl/2019/02/08/gr.234948.118.DC1/Supplemental_Material_1.pdf

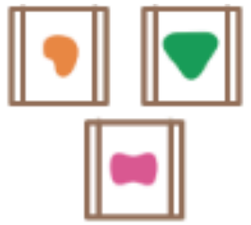
Microservices - Enterprise Applications

A monolithic application puts all its functionality into a single process...

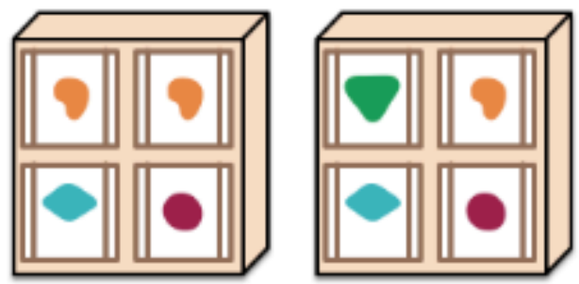
... and scales by replicating the monolith on multiple servers



A microservices architecture puts each element of functionality into a separate service...



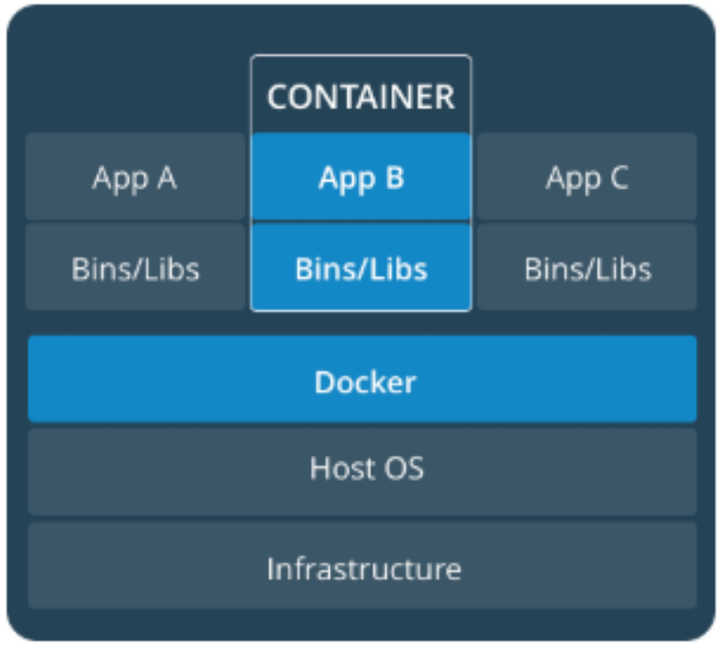
... and scales by distributing these services across servers, replicating as needed.



1. **Highly maintainable and testable**
2. **Loosely coupled (if a service breaks, its self contained)**
3. **Independently deployable**
4. **Organized around business capabilities.**

https://martinfowler.com/articles/microservices.html

What is Docker?



Definition:

- **Docker is a platform to develop, deploy, and run applications with containers.**
- **The use of Linux containers to deploy applications is called *containerization*.**
- **Containers are not new, but their use for easily deploying applications is.**

Simplified version: Docker is a software that enables you to build distributable mini operating systems with a custom list of required software tools

https://docs.docker.com/get-started/

Docker image and containers

Docker Images: **Passive entities with pre-installed software tools**

Docker Containers: **Active instances of the image**

Analogy:

Image <=> A public playlist you create on Spotify

Container <=> Multiple users running your playlist simultaneously

https://docs.docker.com/get-started/

Using micro services to enable reproducibility

Tool ‘Dantooine’

Requirements

1. Python2.7
   - All python dependencies can be installed with: pip install -r requirements.txt
2. Keras (Recommended version >= 2.02)
3. Tensorflow (Recommended verion >= v1.8)
4. Seaborn(0.9.0) for plotting (https://seaborn.pydata.org/installing.html)

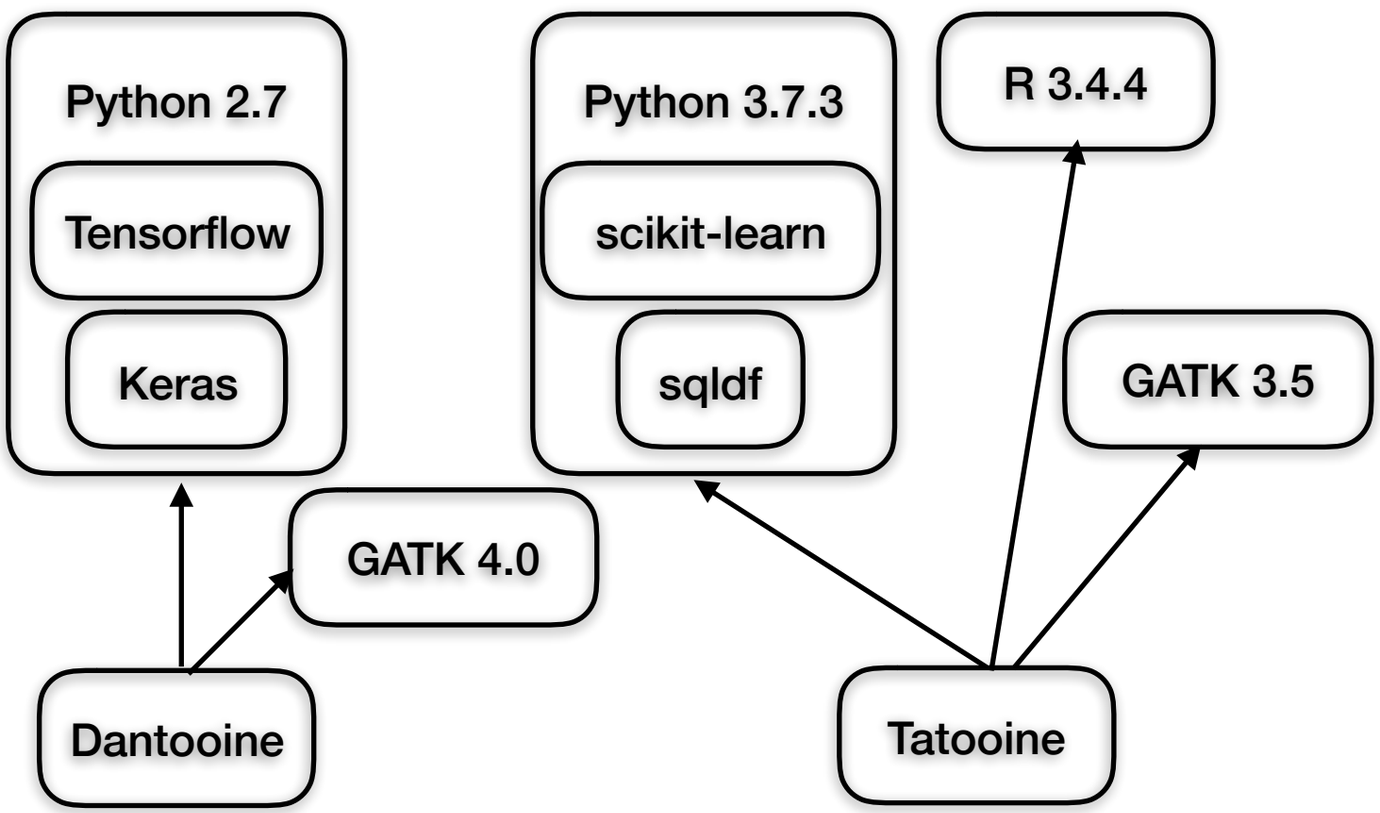
Tool ‘Tatooine’

Software Requirements

1. Python 3.7.3
2. R 3.4.4
3. Java 8
4. GATK 3.5
5. bedtools 2.27.1
6. samtools 1.3.1
7. CANOES, CODEX, CLAMMS, XHMM & CN-Learn

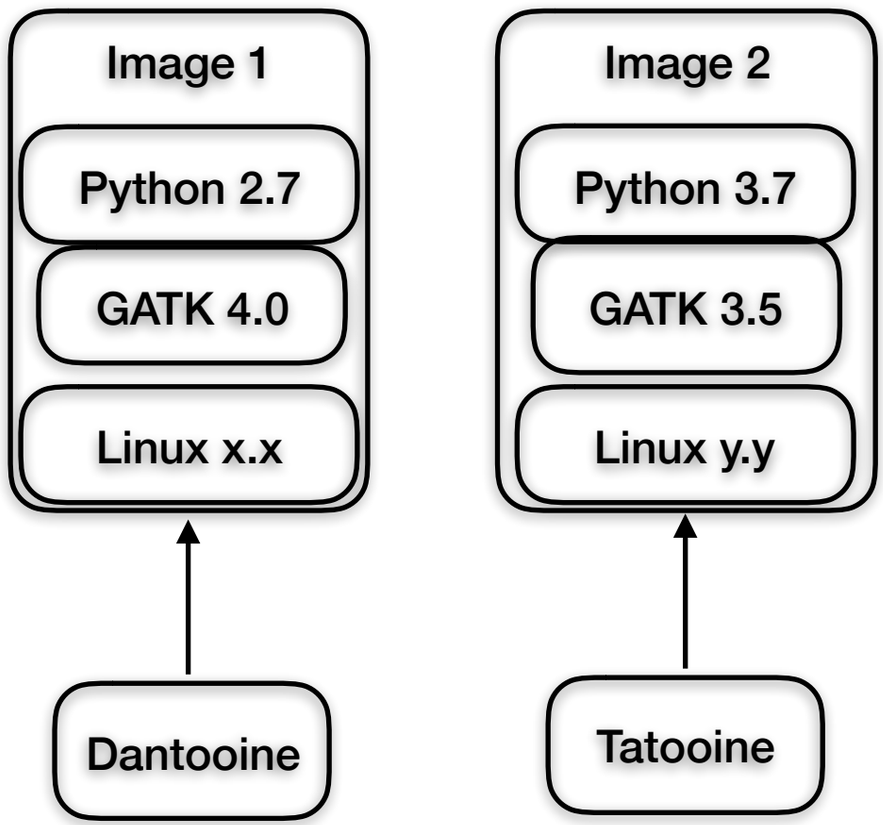
Traditional

Provide a list of all software requirements and dependencies.



Docker

Imagine taking a piece of the operating system and installing all the software prerequisites.



Micro services enable reproducibility & save time

Tool ‘Dantooine’

Requirements

1. Python2.7
   - All python dependencies can be installed with: pip install -r requirements.txt
2. Keras (Recommended version >= 2.02)
3. Tensorflow (Recommended verion >= v1.8)
4. Seaborn(0.9.0) for plotting (https://seaborn.pydata.org/installing.html)

Tool ‘Tatooine’

Software Requirements

1. Python 3.7.3
2. R 3.4.4
3. Java 8
4. GATK 3.5
5. bedtools 2.27.1
6. samtools 1.3.1
7. CANOES, CODEX, CLAMMS, XHMM & CN-Learn

Traditional

Approach: **Provide a list of all software requirements and dependencies.**

Install Time: ~7 hours

Number of users: 100

Total install time = 700 hours

Docker

Approach: **Imagine taking a piece of the operating system and installing all the softwares.**

Install Time: ~1 hours

Number of users: 100

Total install time = 100 hours

Docker : In practice

Usage:

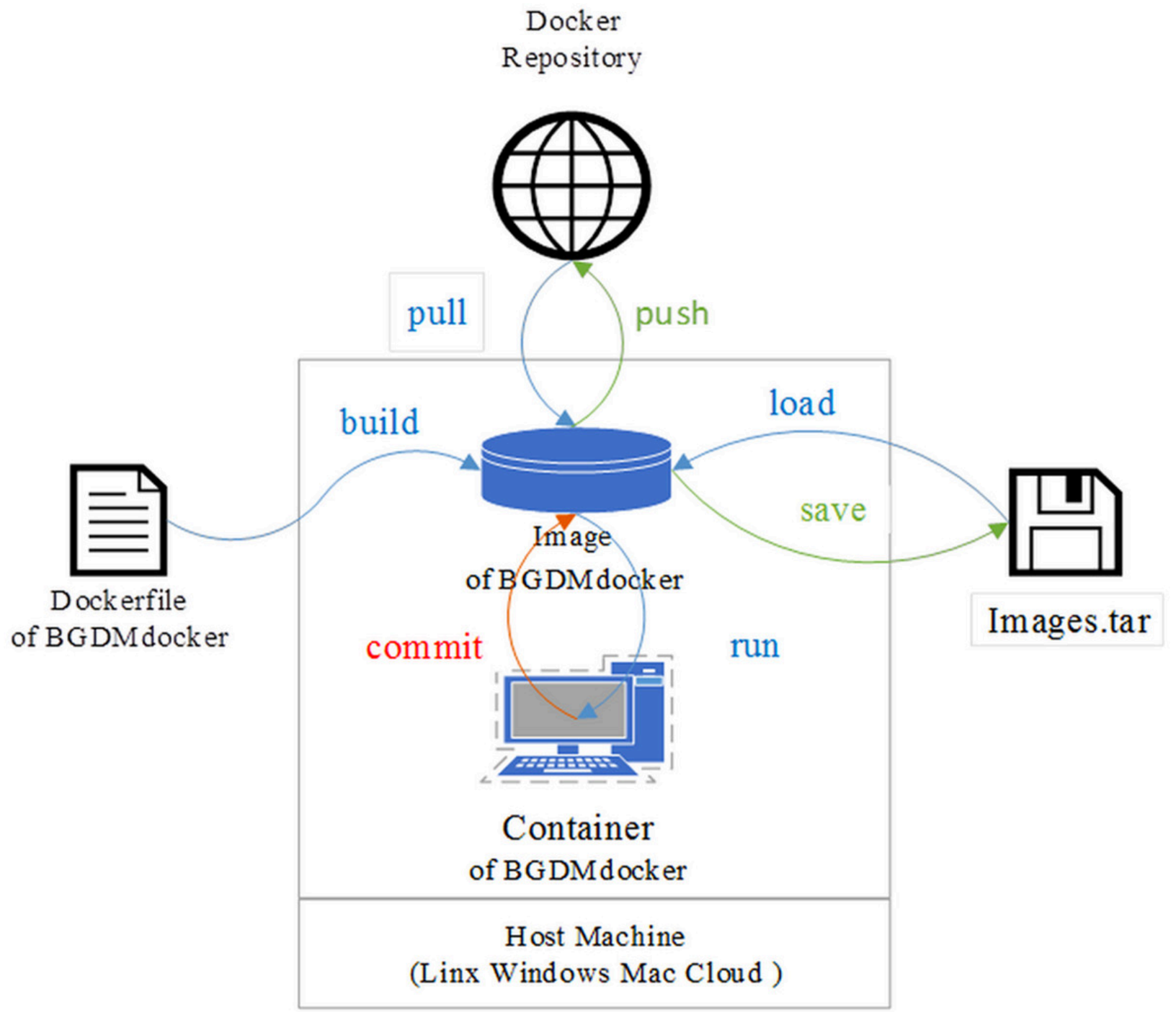
- **Create a text file with the list of softwares that Docker should install on the image it is about to create. This file is named Dockerfile**
- **Execute the Dockerfile to generate the image**
- **Deposit the docker image to docker hub**
- **Users can now download the docker image from the docker hub**
- **Users can execute the image to produce one or more containers**
- **Users can reuse or modify the Dockerfile to create their own image if necessary**

Image -> Comes with a tool that counts the number of stars in a galaxy

Container -> Users start the image and supply a galaxy name. Billions of containers could be doing the counting in parallel.

https://docs.docker.com/get-started/

Docker Workflow



https://peerj.com/articles/3948/

Dockerfile

```
From rocker/r-ver:3.4.4

# Install basic LINUX tools and Java8

RUN apt-get update && apt-get upgrade -y \

    && apt-get install -y \

    autoconf gcc git make ssh wget vim \

    && apt-get install -y --allow-unauthenticated oracle-java8-installer \

    && apt-get clean -y
.
.
RUN R -e "install.packages(c('Rcpp', 'fs', 'usethis'), repos='http://cran.rstudio.com/')"

    R -e "source('https://bioconductor.org/biocLite.R'); biocLite('Biostrings');
biocLite(‘rtracklayer')
.
.
WORKDIR /opt/tools

RUN git clone --recursive https://github.com/girirajanlab/CN_Learn.git
.
.
WORKDIR /opt/tools/CN_Learn/software

RUN wget https://www.python.org/ftp/python/3.7.3/Python-3.7.3.tgz && \

    tar xzf Python-3.7.3.tgz && \

    cd Python-3.7.3 && \

    ./configure && make && make install
```

https://github.com/girirajanlab/CN_Learn/blob/master/Dockerfile

Dockerfile

```
WORKDIR /opt/tools/CN_Learn/software

RUN wget -c https://github.com/samtools/htslib/archive/1.3.2.tar.gz && \
    tar -zxvf 1.3.2.tar.gz && \
    mv htslib-1.3.2 htslib && \
    cd htslib && \
    autoreconf && \
    ./configure && make && make install
.
.
WORKDIR /opt/tools/CN_Learn/software

RUN apt install ./libpng12-0_1.2.54_amd64.deb && \
    rm gatk-3.5.tar.gz  && rm xhmm.tar.gz  && rm clamms.tar.gz && \
    rm Python-3.7.3.tgz && rm 1.3.2.tar.gz && rm 1.3.1.tar.gz  && \
    rm bedtools-2.27.1.tar.gz && rm plinkseq-x86_64-latest.zip
.
.
ENV CLAMMS_DIR=/opt/tools/CN_Learn/software/clamms/

WORKDIR /opt/tools

CMD ["/bin/bash"]
```

https://github.com/girirajanlab/CN_Learn/blob/master/Dockerfile

Getting familiar with Docker

Install Docker

```
[vxm915@durga ~]$ docker --version
Docker version 18.06.3-ce, build d7080c1
```

Write the Dockerfile and place it in a directory

```
[Vijays-iMac:Docker vijay$ ls
Dockerfile
```

Execute the Dockerfile to build the Docker image

```
$ docker build -t svendowideit/ambassador .
Sending build context to Docker daemon 15.36 kB
Step 1/4 : FROM alpine:3.2
 ---> 31f630c65071
Step 2/4 : MAINTAINER SvenDowideit@home.org.au
 ---> Using cache
 ---> 2a1c91448f5f
Step 3/4 : RUN apk update &&      apk add socat &&        rm -r /var/cache/
 ---> Using cache
 ---> 21ed6e7fbb73
Successfully built 7ea8aef582cc
```

List of all images available on the machine

```
[vxm915@durga ~]$ docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
girirajanlab/cnlearn    latest              9fc876f3d069        8 weeks ago         5.13GB
```

https://github.com/girirajanlab/CN_Learn/blob/master/Dockerfile

Getting familiar with Docker

Starting a container using the ‘docker run’ command

```
$ docker run [OPTIONS] IMAGE[:TAG|@DIGEST] [COMMAND] [ARG...]
```

Here is what happens when you run the image and generate a container

```
[vxm915@durga ~]$ docker run -ti giririajanlab/cnlearn
root@88eb4a9e6f1f:/opt/tools# pwd
/opt/tools
root@88eb4a9e6f1f:/opt/tools# ls
CN_Learn
root@88eb4a9e6f1f:/opt/tools# whoami
root
root@88eb4a9e6f1f:/opt/tools# echo docker is fun
docker is fun
root@88eb4a9e6f1f:/opt/tools# exit
exit
[vxm915@durga ~]$ pwd
/afs/bx.psu.edu/user/v/vxm915
```

https://github.com/giririajanlab/CN_Learn/blob/master/Dockerfile

Getting familiar with Docker

```
# Set the docker command based on the indicator set by the user #


if [ ${DOCKER_INDICATOR} = 'Y' ] || [ ${DOCKER_INDICATOR} = 'y' ];

then

DOCKER_COMMAND="docker run --rm -v ${PROJ_DIR}:${PROJ_DIR} -v ${BAM_FILE_DIR}:${BAM_FILE_DIR} -v
${REF_GENOME_DIR}:${REF_GENOME_DIR} --user $(id -u):$(id -g) girirajanlab/cnlearn "

else

DOCKER_COMMAND=''

fi

################################################
# STEP 2: Extract GC content for each interval
################################################
${DOCKER_COMMAND}java -Xmx2000m -Djava.io.tmpdir=${DATA_LOGS_DIR} \
                      -jar ${GATK_SW_DIR}GenomeAnalysisTK.jar \
                      -T GCContentByInterval -L ${TARGET_PROBES} \
                       -R ${REF_GENOME} -o ${DATA_CANOES_DIR}gc.txt

#############################################################################################
# STEP 3: Execute R script to merge data. This is needed because multicov command was executed
#         for just four samples via separate jobs, to parallalize data extraction manually.
#############################################################################################
${DOCKER_COMMAND}Rscript --vanilla ${RSCRIPTS_DIR}canoes_merge_files.r \
                                   ${RSCRIPTS_DIR} ${DATA_CANOES_DIR} ${CONS_READS} canoes_calls.csv
```

Example:

docker run --rm -v /host/directory/:/container/directory -v girirajanlab/cnlearn Rscript …

https://github.com/girirajanlab/CN_Learn/blob/master/Dockerfile

Take aways

1. **Be considerate of your users' time**
2. **Avoid/simplify/offload unnecessary work off your users**
3. **The less installation issues your software runs into, the more likely that it will actually be used**
4. **Docker is just one of the many options available for containerization**
5. **Docker could help avoid user experience related criticisms from the reviewers**