

# FAIR\_Bioinfo

## Practical session : Docker

This session will be divided into four parts:

- General observations on Docker, images and containers
- Pull a pre-defined Docker image, and bind a local folder
- Run commands with the pulled image
- Automate the building process of an image using a Dockerfile

### Table of Contents

<b>FAIR_Bioinfo</b> .....	<b>1</b>
<b>Practical session : Docker</b> .....	<b>1</b>
A. General information.....	2
B. Introduction to Docker.....	3
1. <i>Client / Server</i> .....	3
2. <i>Skeleton of a Docker command</i> .....	3
3. <i>Download a pre-defined image available on the DockerHub</i> .....	4
4. <i>Use a pulled image</i> .....	5
5. <i>Bind a local folder into a container</i> .....	6
C. Creating and interacting with a Docker container.....	7
1. <i>Pulling a Docker image</i> .....	7
2. <i>Interacting with a Docker container</i> .....	8
D. Automating the process.....	9
<b>Annex : create a mini SAM/BAM file.</b> .....	<b>11</b>
<b>Annex : load a saved image</b> .....	<b>12</b>

Version : 1.1 (30/08/2020)

Contacts :

- Thomas Denecker [thomas.denecker@u-psud.fr](mailto:thomas.denecker@u-psud.fr)
- Celine Hernandez [celine.hernandez@i2bc.paris-saclay.fr](mailto:celine.hernandez@i2bc.paris-saclay.fr)
- Claire Toffano-Nioche [claire.toffano-nioche@universite-paris-saclay.fr](mailto:claire.toffano-nioche@universite-paris-saclay.fr)

## A. General information

### Websites

Docker : <https://www.docker.com/>

DockerHub : <https://hub.docker.com/>

Genomic Paris Centre hub : <https://hub.docker.com/u/genomicpariscentre/>

### Resources

#### Docker

Understanding Docker :

<https://docs.docker.com/get-started/overview/>

Get started with Docker :

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

More on Dockerfiles :

[https://docs.docker.com/develop/develop-images/dockerfile\\_best-practices/](https://docs.docker.com/develop/develop-images/dockerfile_best-practices/)

#### Samtools

Official website : <http://www.htslib.org/>

Documentation (v0.1.19) : <http://www.htslib.org/doc/0.1.19/samtools.html>

Documentation (v1.7) : <http://www.htslib.org/doc/1.7/samtools.html>

Note that the commands you have to enter are inside a blue box. The first '\$' character should **not** be copied.

## B. Introduction to Docker

### 1. Client / Server

Docker is a command line-based software allowing to manipulate images and create application containers.

As presented in the course, Docker consists of two elements:

- a server, to manage images and containers
- a client, used to interact with the server

Typing this command will give the Client and Server versions available on your computer.

```
$ docker version
```

### 2. Skeleton of a Docker command

The general usage of the Docker command line is as follows:

```
Usage: docker [OPTIONS] COMMAND [arg...]  
       docker [ --help | -v | --version ]
```

If you consider the first line, a Docker command is made of four parts.

- The 'docker' keyword
- [OPTIONS] : optional parameters to run the client
- COMMAND : the name of a command to be run by the client
- [arg...] : parameters for the chosen command

Usage, options and a full list of available commands can be accessed by typing :

```
$ docker
```

Among the most interesting for this tutorial are the commands pull, run, rm/rmi and build. All commands have a dedicated documentation page.

Try to access the basic usage of one Docker command :

```
$ docker pull
```

Q: How many arguments are absolutely required by the command 'docker pull' ?

Try to access the extended help page of a Docker command :

```
$ docker pull --help
```

This lists also the options specific to the command.

### 3. Download a pre-defined image available on the DockerHub

**Goal: On the Docker hub, how do we find and download the 'Samtools' image made by the GenomicParisCentre user ?**

In a web browser, navigate to the DockerHub website :

<https://hub.docker.com/>

Search for user : GenomicParisCentre

As shown during the presentation, this user made many images available to the scientific community. We will have a look at the available images.

Q: How many times was the STAR image downloaded ?

NB: 'pull' is a synonym for 'download'.

Among the list of images, find Samtools. Click on DETAILS to access the page description for this image.

Note that we could also have searched for all samtools images available on the DockerHub and only then, tried to find the one made by user genomicpariscentre.

Q: On the Samtools image description page, can you find the 'pull' command of Samtools?

Now, copy the pull command and execute it inside a terminal.

By default, Docker will pull the most recent and stable version of the image (tag: latest).

Q: How many times do you see 'Pull complete' displayed ? Why ?

You can find more information on layers here:

<https://docs.docker.com/storage/storagedriver/#images-and-layers>

Now, to be sure that the image was correctly pulled, let's see the list of all available downloaded images inside our workspace.

```
$ docker images
```

Q: What is the size of the genomicpariscentre/samtools image ?

Optional exercise : inspect the image detailed description.

Run this command:

```
$ docker inspect genomicpariscentre/samtools
```

What is displayed on the terminal is a description of the image in JSON format.

#### 4. Use a pulled image

**Goal: Now that we have a locally installed image containing Samtools, how do we use it?**

Among the Docker commands, we will now use the 'run' command.

Q: What are the options and parameters of the 'run' command ?

As displayed on the terminal, the description of the command is 'Run a command in a new container'.

Q: What is the difference between an image and a container ?

See the Docker documentation:

<https://docs.docker.com/get-started/overview/#docker-objects>

Samtools has been installed as a global program in the image we pulled. Consequently, Samtools is directly accessible when interacting with the image.

Now, to run the samtools application, execute the following command:

```
$ docker run genomicpariscentre/samtools samtools
```

```
Program: samtools (Tools for alignments in the SAM format)  
Version: 1.4.1 (using htslib 1.5)
```

Q: What was displayed on the terminal ? Is it a message from Docker or from Samtools ?

Congratulations!

You just successfully downloaded and used your first Docker image !

## 5. Bind a local folder into a container

**Goal: map a local folder and run samtools on a provided BAM file.**

Running samtools without parameters was interesting as a demonstration of Docker's powerful functionalities. But if we want to really run samtools, we also need to provide parameters and, most importantly, input files. Because of encapsulation, this functionality is not available by default. We need to explicitly map a local folder to a corresponding one inside the container (usually /home/), binding the two separated worlds together.

To achieve this mapping, we use the `-v` option of the run command. Before trying it, you can check the command options once again.

```
$ docker run --help
```

To bind our current folder to the /home/ folder located inside a container, we first need the absolute path of the current folder, obtained through the unix `pwd` command.

```
$ pwd
```

This path will be used in further commands through `${PWD}`.

To perform the folder mapping between the current folder and /home inside Docker, the syntax is simple. The `-v` option takes one parameter made out of the two paths we want to map concatenated together, with the character `:` as separator.

Instead of running the samtools command, we will now just list the content of the /home/ folder inside the container.

First without the mapping:

```
$ docker run genomicpariscentre/samtools ls /home/
```

Then with the mapping:

```
$ docker run -v ${PWD}:/home/ genomicpariscentre/samtools ls /home/
```

Q: Is the file list the same as what is in your current folder?

Finally, we can run samtools on a SAM/BAM file, here on a file named "file.bam" located in the current folder. Change the name of the BAM file to one of the provided files or create your own (see last page of this document).

```
$ docker run -v ${PWD}:/home/ genomicpariscentre/samtools samtools view /home/file.bam | less
```

## C. Creating and interacting with a Docker container

Our next goal is to understand what was really done by Docker and create our own image based on a newer version of samtools.

### 1. Pulling a Docker image

**Goal: Requesting the available versions of an image, and downloading an image locally.**

We want to install samtools, but before that, we need to prepare a base Operating System (OS). We would like to use Ubuntu, but we don't know which Ubuntu images are available on the Docker registry.

First, we need to list the available images for Ubuntu. We can either explore the Docker hub website or use a docker command to search for it.

```
$ docker search ubuntu
```

A table with four columns is now displayed. In the first column are the names of the available images. We will use these names to specify which image we are interested in. In the second column are descriptions for each image. The third column is the number of stars given by users to this particular image. The fourth and fifth columns indicate if a given image is an official release, and if it has been automatically created using a Dockerfile. For this practical session, we are only interested in the official ubuntu release, named ubuntu.

Note that the website ([https://hub.docker.com/\\_/ubuntu/](https://hub.docker.com/_/ubuntu/)) gives you more details concerning the available versions of Ubuntu downloadable as images, as well as which one is the default version (tagged 'latest'). For this exercise, we specify the version we want (18.04).

Now that we have found the image we want, we need to download it.

```
$ docker pull ubuntu:18.04
```

```
18.04: Pulling from library/ubuntu
f08d8e2a3ba1: Pull complete
3baa9cb2483b: Pull complete
94e5ff4c0b15: Pull complete
1860925334f9: Pull complete
Digest: sha256:05a58ded9a2c792598e8f4aa8ffe300318eac6f294bf4f49a7abae7544918592
Status: Downloaded newer image for ubuntu:18.04
docker.io/library/ubuntu:18.04
```

Q: What is the size of the downloaded image? Compare that to the 2GB required for a full Ubuntu 18.04 Desktop installation.

<https://help.ubuntu.com/18.04/installation-guide/amd64/apds03.html>

## 2. Interacting with a Docker container

**Goal: from the command line, connect to a container based on the downloaded Ubuntu image.**

Let's start a terminal (bash) session based on the image we just downloaded.

```
$ docker run -ti ubuntu:18.04 bash
```

Normally, the first part of the terminal line has changed to `root@` followed by a series of letters and numbers.

You are now connected to lightweight version of Ubuntu 18.04 !

Next step is to run the command that will install samtools. Don't copy the beginning of the line (`root@c39b1e3c1cf3:/#`).

```
root@c39b1e3c1cf3:/# apt update && apt --yes install samtools=1.7-1 && apt clean
```

`c39b1e3c1cf3` corresponds to the beginning of the container identifier. This identifier is random and different for each container.

Samtools has been installed! You can test the installation by calling the tool.

```
root@c39b1e3c1cf3:/# samtools
```

```
Program: samtools (Tools for alignments in the SAM format)
Version: 1.7 (using htslib 1.7-2)
```

You can now exit the container.

```
root@c39b1e3c1cf3:/# exit
```

The first part of the line should be back to normal.

Note that as we exited the container, it has now stopped. If we type again the first command of this exercise, we will get a brand new container, without samtools installed.

We can check the status of the latest container created.

```
$ docker ps -l
```

This will display a table with seven columns. In the column titled STATUS, you can check that we "Exited" the latest container we created a certain time ago.

Using the 'commit' command, Docker also provides a way to save the containers that have been created and make a static image out of them. But for time reasons, this functionality is not presented in this practical session.

## D. Automating the process

**Goal: Automatically build the container we manually created.**

As explained during the presentation, all the steps that we manually did in the preceding exercise can be collected into a “recipe” called a Dockerfile. A Dockerfile can then be used to build again the image on-demand.

Let’s have a look at the Dockerfile corresponding to what we just did. Lines starting by a # sign are comments and are completely ignored by Docker. Only the two lines starting by FROM and RUN are important.

```
#####  
#  
# Dockerfile to build Samtools 1.7 image  
# Based on Ubuntu 18.04  
#  
#####  
  
# Set the base image to Ubuntu 18.04  
FROM ubuntu:18.04  
  
# Update the repository sources list and install samtools package  
RUN apt update && apt --yes install samtools=1.7-1 && apt clean
```

FROM specifies the base image that should be used as a starting point to create our new image.

RUN specifies any command that should be run for our purpose. In our case, we just needed one command line.

You can find more information on what can be included in a Dockerfile at this URL:

[https://docs.docker.com/develop/develop-images/dockerfile\\_best-practices/](https://docs.docker.com/develop/develop-images/dockerfile_best-practices/)

Note that, for instance, it is usually a good practice to give also the name of a MAINTAINER, responsible for the writing and maintenance of the Dockerfile.

Now that we saw what is a Dockerfile, we will use it to build an image containing Samtools version 1.3.1, and most importantly, save it as an image.

```
$ docker build -t student/samtools:1.7 .
```

Docker will now execute one by one each instruction written inside the Dockerfile.

We can now check that we have locally a new image called student/samtools with the tag 1.7.

```
$ docker images
```

And we can run a simple command to check if Samtools is working and what is the installed version in the image.

```
$ docker run student/samtools:1.7 samtools
```

This is the end of the practical session. We hope you enjoyed it.  
Don't hesitate to ask any questions and feel free to contact us any time after  
the session (email addresses on the front page).

## Annex : create a mini SAM/BAM file.

Open a text editor (simple, like Notepad or BBEdit).

In the blank page, paste the text below:

```
@HD VN:1.0 SO:coordinate
@SQ SN:ERCC-00002 LN:1037
@SQ SN:chr12 LN:133275309
@PG ID:TopHat VN:2.1.1 CL:/opt/tophat-2.1.1/bin/tophat
NB552053:112:H7LFHBGXG:4:13404:24475:13506 385 ERCC-00002 243 1 35M
chr12 124911843 0 AAGCTTGTTCCGTTAGCTGGCATAAGATTCCATGC
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AS:i:0 XN:i:0 XM:i:0 XO:i:0
XG:i:0 NM:i:0 MD:Z:35 YT:Z:UU NH:i:3 CC:Z:= CP:i:243 HI:i:0
NB552053:112:H7LFHBGXG:4:13404:24475:13506 129 ERCC-00002 243 1 35M
chr12 124913211 0 AAGCTTGTTCCGTTAGCTGGCATAAGATTCCATGC
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA AS:i:0 XN:i:0 XM:i:0 XO:i:0
XG:i:0 NM:i:0 MD:Z:35 YT:Z:UU NH:i:3 CC:Z:= CP:i:243 HI:i:1
```

Save the file and name it `file.sam`

Consecutively, you can use a Docker image of Samtools to convert it into a BAM file.

```
$ docker run -v ${PWD}:/home/ genomicpariscentre/samtools samtools view -b -o
/home/file.bam /home/file.sam
```

## Annex : load a saved image

If you have problems with your Internet connection during the practical session, you can ask for a downloaded version of the Samtools image used here.

In order to load the image, use this command.

```
$ docker load < savedsamtoolsimage.tar
```

```
0f5ff0cf6a1c: Loading layer [=====>]
126MB/126MB
f1c896f31e49: Loading layer [=====>]
15.87kB/15.87kB
51db18d04d72: Loading layer [=====>]
14.85kB/14.85kB
f51f76255b02: Loading layer [=====>]
5.632kB/5.632kB
174a611570d4: Loading layer [=====>]
3.072kB/3.072kB
bfb4c941db16: Loading layer [=====>]
58.08MB/58.08MB
Loaded image: genomicpariscentre/samtools:latest
```

You can then check that it has been correctly loaded using 'docker images'.

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	18.04	6526a1858e5d	10 days ago	64.2MB
genomicpariscentre/samtools	latest	2ec1cf58981b	2 years ago	179MB

For your information, this is how you save an image to a transportable archive.

First you need to identify the image that you want to export.

```
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	18.04	6526a1858e5d	10 days ago	64.2MB
genomicpariscentre/samtools	latest	2ec1cf58981b	2 years ago	179MB

Then, you need to save the image to an archive. More information on the 'docker save' command is available here:

<https://docs.docker.com/engine/reference/commandline/save/>

```
$ docker save genomicpariscentre/samtools > ./savedsamtoolsimage.tar
```

You can check the image has been saved.

```
$ ls -alh
```

```
drwxr-xr-x  4 chernand  staff  128B Aug 30 16:42 .
drwxr-xr-x 20 chernand  staff  640B Aug 30 16:39 ..
-rw-r--r--  1 chernand  staff  176M Aug 30 16:42 savedsamtoolsimage.tar
```