Sign in

Need Help?

Search our documentation

Enter a question, topic or keyword...

GATK / Getting Started / Tutorials

Articles in this section

(How to) Run GATK in a Docker container

Follow

GATK Team
2 months ago · Updated

This document explains how to install and use Docker to run GATK on a local machine. For a primer on what Docker containers are for and related terminology, see this Dictionary entry.

Contents

1. Install Docker
2. Test that it works
3. Get the GATK container image
4. Start up the GATK container
5. Run a GATK command in the container
6. Use a mounted volume to access data that lives outside the container

1. Install Docker

Follow the relevant link below depending on your computer system; on Mac and Windows,

select the "Stable channel" download. Run through the installation instructions and initial setup page; they are very straightforward and should only take you a few minutes (not counting download time).
We have included instructions below for all steps after that first page, so you shouldn't need to go to any other pages in the Docker documentation. Frankly their docs are targeted at people who want to do things like run web applications on the cloud and can be quite frustrating to deal with.

MacOS systems

Click here for the MacOS install instructions

On Mac, the installation adds a menu bar item that looks like a whale/container-ship, which conveniently shows you the status of the Docker "daemon" (= program that runs in the background) and gives you GUI access to various Docker-related functionalities. But you can also just use it from the command-line, which is what we'll do in the rest of this tutorial.

Windows systems

Click here for the Windows install instructions

Note that on some Windows systems (including non-Pro versions like Windows Home, and older versions) the "normal" Docker app doesn't work, and you have to use an older app called Docker Toolbox, which you can find here.

Linux systems

Here is the full list of supported systems and their install pages.

2. Test that it works

Now, open a terminal window and invoke the `docker` program directly. Checking the version is always a good way to test that a program will run without investing too much effort into finding a command that will work, so let's do:

```
docker --version
```

This should return something like "Docker version 17.06.0-ce, build 02c1d87".

If you run into trouble at this step, you may need to run one or more of the following commands:

```
docker-machine restart default
docker-machine regenerate-certs
```

```
docker-machine env
```

Note that we have had reports that Docker is not compatible with some other virtual machine software; if you run into that problem you may need to uninstall other software. Or, uh, install Docker in a virtual machine? Ahhhh, too many layers! Let's just assume your Docker install worked fine. (If not, let us know in the forum and we'll try to help you)

3. Get the GATK container image

Still in your terminal (it doesn't matter where your working directory is), run the following command to retrieve the GATK image from Docker Hub:

```
docker pull broadinstitute/gatk:4.1.3.0
```

Note that the last bit after `gatk:` is the version tag, which you can change to get a different version than what we've specified here. At time of writing we're using the latest released version.

The GATK container image is quite large so the download may take a little while if you've never done this before. The good news is that next time you need to pull a GATK image (e.g. to get another release), Docker will only pull the components that have been updated, so it will go faster.

4. Start up the GATK container

There are several different ways to do this in Docker. Here we're going to use the simplest invocation that gets us the functionality we need, i.e. the ability to log into the container once it's running and execute commands from inside it.

```
docker run -it broadinstitute/gatk:4.1.3.0
```

If all goes well, this will start up the container in interactive mode, and you will automatically get logged into it. Your terminal prompt will change to something like this:

```
root@ea3a5218f494:/gatk#
```

At this point you can use classic shell commands to explore the container and see what's in there, if you like.

5. Run a GATK command in the container

The container has the `gatk` wrapper script all set up and ready to go, so you can now run any GATK or Picard command you want. Note that if you want to run a Picard command, you need to use the new syntax, which follows GATK conventions (`-I` instead of `I=` and so on). Let's use `--list` to list all tools available in this version.

```
./gatk --list
```

The output will start with a usage message (shown below) then a full list of tools and their summary descriptions.

```
Using GATK wrapper script /gatk/build/install/gatk/bin/gatk
Running:
    /gatk/build/install/gatk/bin/gatk --help
USAGE:   [-h]
```

Once you've verified that this works for you, you know you can run any GATK commands you want. But before you proceed, there's one more setup thing to go through, which is technically optional but will make your life much easier.

6. Use a mounted volume to access data that lives outside the container

This is the final piece of the puzzle. By default, when you're inside the container you can't access any data that lives on the filesystem outside of the container. One way to deal with that is to copy things back and forth, but that's wasteful and tedious. So we're going to follow the better path, which is to **mount a volume** in the container, i.e. establish a link that makes part of the filesystem visible from inside the container.

The hitch is that you can't do this after you started running the container, so you'll have to shut it down and run a new one (not just restart the first one) with an extra part to the command. In case you're wondering why we didn't do this from the get-go, it's because the first command we ran is simpler so there's less chance that something will go wrong, which is nice when you're trying something for the first time.

To shut down your container from *inside* it, you can just type exit while still inside the container:

```
exit
```

That should stop the container and take you back to your regular prompt. It's also possible to exit the container without stopping it (a move called *detaching*) but that's a matter for another time since here we do want to to stop it. You'll probably also want to learn how to clean up and delete old instances of containers that you no longer want.

For now, let's focus on starting a new instance of the GATK4 container, specifying in the following command what is your particular container ID and the filesystem location you want to mount.

```
docker run -v ~/my_project:/gatk/my_data -it broadinstitute/gatk:4.1.3.0
```

Here I set the external location to be an existing directory called `my_project` in my home directory (the key requirement is that it has to be an absolute path) and I'm setting the **mount point** inside the container's `/gatk` directory. The name of the mount point can be the same as the mount directory, or something completely different; the main constraint is that it should not conflict with an existing directory, otherwise that would make the existing directory unattainable.

Assuming your paths are valid, this command starts up the container and logs you into it the same way as before; but now you can see by using `ls` that you have access to your filesystem. So now you can run GATK commands on any data you have lying around. Have fun!

f in 3

Was this article helpful?

✓ Yes × No

7 out of 7 found this helpful

Return to top

Related articles

Getting started with GATK4

(How to) Execute Workflows from the gatk-workflows Git Organization

Docker - container - image - registry

RNAseq short variant discovery (SNPs + Indels)

* Tool Documentation Index

3 comments

Sort by ⌄

vivekruhela
5 months ago

Hi,

Thanks for this tutorial. I have successfully pulled the gatk docker image as described above but when I tried this command

```
docker run -v ~/my_project:/gatk/my_data -it broadinstitute/gatk:4.1.3.0
```

but I am getting an error of invalid mount specifications. '~/bam_files: gatk/my_data' invalid mount config for type "bind" : invalid mount path. mount path must be absolute. Kindly suggest.

↑ 0 ↓ ⚙

Humphrey Gardner
2 months ago

did you already create the "my_project" folder in your home directory?

↑ 0 ↓ ⚙

vivekruhela
2 months ago

Humphrey Gardner

I tried it few months before. As far as I remember, I have already created the directory `bam_files` in my home directory and then I was trying to mount it to docker. I am not using docker now. I am using the java version of GATK in my work. Thanks.

↑ 0 ↓

Please sign in to leave a comment.