Docker Minitutorial

Robert Ladwig

Summary

Creates a container that enables your machine to run **the latest GLM-AED2 version** without downloading or compiling it by yourself.

The hydrobert/glm-aed2 container (https://hub.docker.com/r/hydrobert/glm-aed2) is based on the jsta/GLM_docker container. Original docker image and code by Joseph Stachelek (https://github.com/jsta/GLM_docker)

Installation

Install docker: https://docs.docker.com/v17.12/install/; and (optional) create account at docker hub: https://hub.docker.com.

Open a terminal and get the glm-aed2 container:

```
docker pull hydrobert/glm-aed2
```

Once docker is installed, you can:

(1) See if it works:

```
docker run -it hydrobert/glm-aed2 /bin/bash
```

Typing `ls` in terminal will list all the files in the current directory (docker image). You should see `glm` as one of the files

(2) Run it linked to a local simulation folder

```
docker run -i -t -v /<YOUR_FOLDER_WITH_GLM_SIMULATION>:/GLM/<yourmodel>
  hydrobert/glm-aed2 /bin/bash
```

An example:

```
docker run -i -t -v /Users/Documents/Testfolder:/GLM/TestLake
  hydrobert/glm-aed2 /bin/bash
```

If you type `ls` into terminal, you should now see your model listed as a file. Using our example, `TestLake` now appears

(3) In both cases: if docker is running, go to the simulation path, e.g.

```
cd <yourmodel>
```

Our example:

```
cd TestLake
```

and run the simulation via

```
/GLM/glm
```

Useful commands

Once you have sucessfully run the model, we recommend quitting docker so the images are not running on your computer.

Quitting docker:

- `exit`

List all installed images:

- `docker images -a`

Remove an image:

- `docker rmi <IMAGE_ID>`

List all finished containers:

- `docker ps -a -q`

Stop all running containers:

- `docker kill $(docker ps -q)`

Delete all finished or stopped containers:

- `docker rm $(docker ps -a -q)`

Docker in R

Please also open the run_docker_glmaed.R script to run a docker example in Rstudio, which automatically downloads files for a GLM run. If you want to use it in R (exchange the simulation path with the path to your specific simulation folder):

```
# start docker as background process (detached)
system('docker run -it -d -v /<YOUR_FOLDER_WITH_GLM_SIMULATION>:/GLM/<yourmodel>
       hydrobert/glm-aed2 /bin/bash')
# get the id of your running container
dockerps <- system('docker ps',intern = TRUE)
dockerid <- strsplit(dockerps, split = "/t")
dockerid <- dockerid[[2]][1]
# start the simulation (i - interactive, t - tty (user input))
system(paste('docker exec -ti',dockerid,'/bin/bash -c \"cd <yourmodel>; /GLM/glm\"'))
```

If you want to run above code in Rstudio, modify the last line to:

```
# no tty command for Rstudio
system(paste('docker exec -t',dockerid,'/bin/bash -c \"cd <yourmodel>; /GLM/glm\"'))
```

Our example in a nutshell:

```
# Open the docker container
system('docker run -it -d -v /Users/Documents/Testfolder:/GLM/TestLake hydrobert/glm-aed2 /
       bin/bash')
# get the id of your running container
dockerps <- system('docker ps',intern = TRUE)
dockerid <- strsplit(dockerps, split = " ")
dockerid <- dockerid[[2]][1] #If you have more than one container running,
# you will need to change this start the simulation (i - interactive, t - tty (user input))
system(paste('docker exec -ti',dockerid,'/bin/bash -c \"cd TestLake; /GLM/glm\"'))
```