

Como hacer un container con singularity

Voy a probar a hacer y ejecutar un container para el [WGS](#) ya que esto al menos he conseguido ejecutarlo localmente.

Construir el container

El container se hace partiendo de un container existente al que se le añaden las herramientas que necesite y las configuraciones necesarias. Aqui se copian archivos, se instalan bibliotecas y programas, se compila lo que sea necesario, etc. Cada seccion esta descrita en la [documentacion de singularity](#)

Receta:

```
Bootstrap: docker
From: centos:centos8
%files
  /data/wgs_image/tools/gatk3.jar /opt/
  /data/wgs_image/tools/gatk4 /opt/
  /data/wgs_image/tools/picard.jar /opt/
  /data/wgs_image/tools/bwa /opt/
  /data/wgs_image/tools/samtools /opt/
  /data/wgs_image/tools/verifyBamID /opt/
  /data/wgs_image/tools/jdk8u265-b01 /usr/lib/jvm/
%post
  yum -y install epel-release
  yum -y install zlib-devel make gcc ncurses-devel bzip2-devel xz-devel
  gcc-c++ openssl-devel R-core python2 python38 git-lfs git
  alternatives --install /usr/bin/java java /usr/lib/jvm/jdk8u265-
  b01/bin/java 1
  alternatives --auto java
  ln -s /usr/bin/python3 /usr/bin/python
  cd /opt/bwa
  make
  cd /opt/samtools
  ./configure --prefix=/opt
  make
  make install
  cd /opt/verifyBamID/verifyBamID
  make
  echo '#!/bin/sh' > /opt/bin/picard
  echo 'java -jar /opt/picard.jar $*' >> /opt/bin/picard
  chmod +x /opt/bin/picard
  echo '#!/bin/sh' > /opt/bin/gatk3
  echo 'java -jar /opt/gatk3.jar $*' >> /opt/bin/gatk3
  chmod +x /opt/bin/gatk3
%environment
```

```
export
PATH=$PATH:/opt/gatk4:/opt/bin:/opt/bwa:/opt/verifyBamID/verifyBamID/bin:/usr/lib/jvm/jdk8u265-b01/bin
%test
  picard -h
  exit 0
```

Un poco de explicacion

- *Bootstrap* y *From* indican que para empezar singularity debe bajarse el centos 8 de un hub docker al disco duro.
- *%files* indica que archivos del disco duro local han de copiarse dentro del container y donde
- *%post* es el procedimiento de post install, como si se tratara de una maquina nueva. indica al container las operaciones que han de hacerse. Aqui se instalan paquetes nuevos, se compilan programas, se hacen scripts customized
- *%environment* se usa para fijar cuales son las variables que queremos añadir cuando ejecutemos el container. Aqui se actualiza PATH pero puede ser LD_LIBRARY_PATH o cualquier otra
- *%test* esto solo se usa cuando se construye el container. Son ordenes que se ejecutan cuando se termina de construir y sirven para comprobar que las cosas funcionan bien.

Para construir el container guardo esta receta en un archivo de texto (ejemplo wgs.singularity) y ejecuto,

```
[root@cthulhu wgs_image]# singularity build wgs.sif wgs.singularity
...
...
...
```

Usar el container

La manera mas sencilla,

```
[root@cthulhu wgs_image]# singularity run wgs.sif bwa

Program: bwa (alignment via Burrows-Wheeler transformation)
Version: 0.7.17-r1188
Contact: Heng Li <lh3@sanger.ac.uk>

Usage:   bwa <command> [options]

Command: index          index sequences in the FASTA format
         mem            BWA-MEM algorithm
         fastmap        identify super-maximal exact matches
         pmerge         merge overlapping paired ends (EXPERIMENTAL)
         aln            gapped/ungapped alignment
         samse          generate alignment (single ended)
         sampe          generate alignment (paired ended)
```

bwasw	BWA-SW for long queries
shm	manage indices in shared memory
fa2pac	convert FASTA to PAC format
pac2bwt	generate BWT from PAC
pac2bwtgen	alternative algorithm for generating BWT
bwtupdate	update .bwt to the new format
bwt2sa	generate SA from BWT and Occ

Note: To use BWA, you need to first index the genome with `bwa index`. There are three alignment algorithms in BWA: `mem`, `bwasw`, and `aln/samse/sampe`. If you are not sure which to use, try `bwa mem` first. Please `man ./bwa.1` for the manual.

Un call mas realista es algo asi,

```
[osotolongo@brick03 wgs]$ singularity run --cleanenv -B /nas:/nas -B /the_dysk:/the_dysk /nas/osotolongo/wgs/bin/wgs.sif samtools index /home/osotolongo/wgs/seq-6/tmp/seq6_sorted.bam
```

Ojo que aqui `-B /nas:/nas` o `-B /the_dysk:/the_dysk` no son estrictamente necesarios. Estan puestos para que si se ha de cambiar de entorno se puedan montar las unidades que se montan.

Usando un container ya hecho

```
[root@brick04 build]# singularity build saige.0.43.2.simg docker://wzhou88/saige:0.43.2
INFO: Starting build...
...
[root@brick04 build]# ls
saige.0.43.2.simg
[root@brick04 build]# cp saige.0.43.2.simg /nas/usr/local/opt/singularity/
```

y ya esta,

```
[osotolongo@brick03 f5cehbi]$ singularity run --cleanenv /nas/usr/local/opt/singularity/saige.0.43.2.simg step1_fitNULLGLMM.R --help
```

From:
<https://cortafuegos.fundacioace.com/wiki/> - **Detritus Wiki**

Permanent link:
https://cortafuegos.fundacioace.com/wiki/doku.php?id=cluster:make_container

Last update: **2020/12/17 10:21**

