Running Cactus on AWS Protocol

By Martina Veit Acosta

Notes Before Starting

1. Genomes must be soft masked (make sure to download the correct version from NCBI)
2. Remove bootstrap values from tree using **R language**:

```
# Import libraries
library(phylogram)
library(ape)

# Create tree using Newick's format
myTree <-
ape::read.tree(text='((Homo_sapiens:0.0793732,Mus_musculus:0.0844682)0.970633:0.2343059,((Le
pisosteus_oculatus:0.130925,Amia_calva:0.189194)0.640456:0.0391649,(Oryzias_latipes:0.121261
,((Kryptolebias_marmoratus:0.0513167,Nematolebias_whitei:0.0863117)0.703993:0.0258997,Nothob
ranchius_furzeri:0.1190887)0.515438:0.0293196)0.471387:0.2025449)0.389323:0.0935769)0.819953
;')

# make sure that tree is looking good
plot(myTree)

# remove bootstrap
myTree$node.label <- NULL

# save new file
write.tree(myTree, file = "MyTreeNoBoots.tre")
```

1. Setting Up AWS Account

Root User: Follow this link to create a root user account. Note: In order to create a root user account, credit card info is needed but no costs are billed at this initial step.

IAM User: While still logged in with the root user access, go to IAM → Users → Add users

Give IAM user admin permissions: Once the IAM user is created, click on the user name (ex. IAM → Users → Add users → Martina). Go to Permissions → Add Permissions. In Permissions Options, select Attach policies directly. For Admin permissions, select AdministratorAccess. Now the IAM user has permission to launch instances.

Create an access key for IAM User: In order to sign programmatic requests that you make to AWS, the IAM user needs access keys. Go to IAM → Users → Security Credentials. Select Create Access Key, and download a CSV file with easy access for later with the Access Keys values.

2. Creating a Key Pair

For the next steps, we need the Linux Operating System. Make sure to partition the computer memory correctly so that it can run both Linux and Windows, or only Linux if preferred. We are using Ubuntu software for our purposes. On the computer terminal, write the following code:

```
$ ssh-keygen -t rsa -b 4096                           # create the key
Generating public/private rsa key pair.
Enter file in which to save the key
(/home/xtremodevolab/.ssh/id_rsa): /home/xtremodevolab/.ssh/id_aws
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/xtremodevolab/.ssh/id_aws.
Your public key has been saved in /home/xtremodevolab/.ssh/id_aws.pub.
The key fingerprint is:
SHA256:XXXXXXXX xtremodevolab@XXXXXXXXX

$ touch ~/.ssh/authorized_keys                         # create an authorized_keys file
$ chmod 0600 ~/.ssh/authorized_keys                    # set correct permissions on that
$ cat ~/.ssh/id_aws.pub >> ~/.ssh/authorized_keys      # put id_aws in authorized_keys
$ eval `ssh-agent -s`                                  # add the key to ssh-agent
$ ssh-add /home/me/.ssh/id_aws
```

Next, go back to AWS and login using your IAM User account. For this step, you will need the Account Number (top right corner of the AWS console, select the Account name, it is under Account ID).

Once you are logged in with the IAM User Account, go to EC2 > Key Pairs (side panel) > Import Key Pair (top of page). Make sure that the timezone is set to us-east-1 (in our case). Paste the contents of your .pub key from the previous step. *Note: .ssh folder will be hidden, so you need to select the "sandwich like" icon on Ubuntu files and select "show hidden files".

3. Increasing Instance Limits

The way that I did this is by going to Service Quotas in the AWS dashboard and selecting EC2. I made sure that the dashboard was in N. Virginia time zone. Next, I used the filter " All Standard (A, C, D, H, I, M, R, T, Z) Spot Instance Requests" and increased from 32 default to 70. Next, I typed Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances and requested an increase to 300 for r4.8xlarge. The requests were approved, however it took up to a couple of days for the services to be available. See AWS info website for more details: https://aws.amazon.com/ec2/instance-types/

4. Launching an Instance

Go to the Terminal again and install the necessary packages. Check if you have the commands by typing `$conda --help` for example.

Conda: https://conda.io/projects/conda/en/latest/user-guide/install/index.html
Git: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git
Awscli: https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html

```
$ conda create -n cactus python=3 awscli          # make the conda environment with python
$ conda activate cactus                           # activate the environment
$ pip install "toil[aws]"                         # install toil
$ sudo apt-get install build-essential python-dev-is-python3
$ git clone https://github.com/comparativegenomicstoolkit/cactus.git --recursive && cd
cactus
```

Next, get your IAM access key values that you saved as a .csv file previously.

```
$ head '/home/xtremodevolab/Downloads/accessKeys.csv'
```

You will need the key values to “make requests” through the terminal. Fill it out with your own info:

```
$ aws configure
```

The next step is to launch the instance. In our case this is the data specs that we decided on:

- Cluster name: xtremodevolab-cactus
- KeyPairName: id_aws (only viewable if timezone is adjusted to us-east-1a)
- leaderNodeType: t2.medium
- Leader storage: 1000 GB
- Node storage: 250 GB

Command to launch the instance:

```
$ toil launch-cluster -z us-east-1a --keyPairName id_aws --leaderNodeType t2.medium
--leaderStorage 1000 --nodeStorage 250 xtremodevolab-cactus
```

You can check if the instance was successfully created by going to the EC2 dashboard and looking up instances.Don’t forget to adjust the timezone in the console so that the instance is visible to you. *Note: make sure to stop the instance if not using it, but don’t terminate it.

5. Storing FASTA files for analysis

The next step is to get the FASTA genomes that you wish to align. The easiest way to do so is to go to https://www.ncbi.nlm.nih.gov/ , select “Assembly” for databases, type the organism name in the search bar and download the chromosome level RefSeq. Unzip the folders and place it in an easy access location.

We need to store this data somewhere on Amazon. In order to do so, we will create a “bucket”.

```
$ aws s3api create-bucket --bucket bucket-xtremodevo-cactus --region us-east-1
{
    "Location": "/bucket-xtremodevo-cactus"
}
```

To check if the bucket was successfully created, go to S3 in the Amazon dashboard and look for buckets. The next step is to upload the .fna files into the bucket. The easiest way to do so, and prevent unnecessary docs from being uploaded into the bucket is to go to S3 bucket in the Amazon Dashboard and select Import files. For 16GB, the wait time until everything was imported was about 1.5 hours.

6. Logging into the instance

Go to the Amazon dashboard and make sure your instance is running. In your terminal activate the virtual environment and login to the instance using toil:

```
$ conda activate cactus
$ toil ssh-cluster -z us-east-1a xtremodevolab-cactus
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

We still need to install some stuff for this virtual environment:

```
apt update
apt install -y git tmux
virtualenv --system-site-packages -p python3 venv
source venv/bin/activate
git clone https://github.com/comparativegenomicstoolkit/cactus.git --recursive
cd cactus
pip install --upgrade .
cd /
```

The next step is to create a txt file using the vim command. First, launch vim and create a txt file:

```
vim seqFile.txt
```

Next, edit the text editor with the data that you need (our lab example):

```
((Homo_sapiens:0.0793732,Mus_musculus:0.0844682):0.2343059,((Lepisosteus_oculatus:0.130925,A
mia_calva:0.189194):0.0391649,(Oryzias_latipes:0.121261,((Kryptolebias_marmoratus:0.0513167,
Nematolebias_whitei:0.0863117):0.0258997,Nothobranchius_furzeri:0.1190887):0.0293196):0.2025
449):0.0935769);
Homo_sapiens s3://xtremo-devo-cactus/GCA_000001405.29_GRCh38.p14_genomic.fna
Mus_musculus s3://xtremo-devo-cactus/GCA_000001635.9_GRCm39_genomic.fna
Lepisosteus_oculatus s3://xtremo-devo-cactus/GCA_000242695.1_LepOcu1_genomic.fna
Amia_calva s3://xtremo-devo-cactus/GCA_017591415.1_AmiCal1_genomic.fna
Oryzias_latipes s3://xtremo-devo-cactus/GCA_002234675.1_ASM223467v1_genomic.fna
Kryptolebias_marmoratus s3://xtremo-devo-cactus/GCA_001649575.2_ASM164957v2_genomic.fna
Nematolebias_whitei s3://xtremo-devo-cactus/GCA_014905685.2_NemWhi1_genomic.fna
Nothobranchius_furzeri s3://xtremo-devo-cactus/GCA_001465895.2_Nfu_20140520_genomic.fna
```

To save and quit the text file in vim press ESC:

```
:w seqFile.txt
:q
```

7. Running Cactus

The key parameters you'll care about changing (besides the usual Cactus parameters) are the auto scaling parameters --nodeTypes, --minNodes, --maxNodes and the jobStore location, aws:<region>:<jobStoreName>.

--nodeTypes option lets you specify the list of instances you want as well as the price you're willing to pay for spot instances. A value like r4.8xlarge indicates that we also want on-demand r4.8xlarge instances (for which we pay exactly the on-demand price, which is usually substantially higher). –consCores is the number of threads, recommendation is one per core.

Next we will start our cactus run:

```
# start tmux so we can exit the session while leaving cactus running
tmux
source venv/bin/activate

# start run
cactus --consCores 2 --nodeTypes c4.8xlarge,r4.8xlarge --minNodes 0,0 --maxNodes 20,1
--nodeStorage 250 --provisioner aws --batchSystem mesos --metrics
aws:us-east-1:xtremo-devo-cactus --logFile cactus.log seqFile.txt output.hal
```

You can exit the tmux session and the leader node. To detach (meaning exit the window to come back to later) from the tmux session, use CTRL + b then d (hold ctrl, press b, let go of both of the keys, and press d). Whatever program(s) you are running in the tmux session will continue going without you. To re-attach back to a session you've become detached from, simply type in: tmux attach

If that runs successfully, there will be a output.hal file.You can download that output.hal file using toil:
toil rsync-cluster -p aws -z us-east-1 cactus-xtremodevo-bucket :output.hal

When logging out and stopping using the environment, make sure to exit toil, deactivate the virtual environment and to stop the instance from running on Amazon dashboard to avoid any costs:

```
$ conda deactivate
```

Important Notes:

Debugging:

- conscores required for non single machine batch systems (solution: add conscores argument when running cactus)
- mkdir_p error (solution: run from cactus manual example, check python version)
- an error occurred (InsufficientInstanceCapacity) when calling the RunInstances operation (reached max retries: 4). We currently do not have sufficient capacity in the Availability Zone you requested." (solution: retry at a different time of the day)
- JobStore already exists error (solution: rerun using –restart)
- FASTA headers error (solution: check versions of Toil and Cactus. Remake AWS bucket)

To check Billing in AWS:
Check it routinely, because even if you think you are not getting charged, you might be storing hidden files and paying for them. Delete instances and buckets if not using them anymore. Keep things clean and neat to avoid spending extra money.

If you get an error related to the headers, try removing spaces and special characters in them:

```
sed '/^>/s/ .*//'  Ami_cal.fna > my-mod-Ami_cal.fna
sed '/^>/s/ .*//'  Hom_sap.fna > my-mod-Hom_sap.fna
sed '/^>/s/ .*//'  Kry_mar.fna > my-mod-Kry_mar.fna
sed '/^>/s/ .*//'  Lep_ocu.fna > my-mod-Lep_ocu.fna
sed '/^>/s/ .*//'  Mus_mus.fna > my-mod-Mus_mus.fna
sed '/^>/s/ .*//'  Nem_whi.fna > my-mod-Nem_whi.fna
sed '/^>/s/ .*//'  Not_fur.fna > my-mod-Not_fur.fna
sed '/^>/s/ .*//'  Ori_lat.fna > my-mod-Ori_lat.fna
```

Save Log File: toil rsync-cluster -p aws -z us-east-1b xtremodevolab-cactus :/cactus.log .

References

Running Cactus — faircloth-lab e4a8853 documentation. (n.d.). https://protocols.faircloth-lab.org/en/latest/protocols-computer/analysis/analysis-cactus.html

ComparativeGenomicsToolkit. (n.d.). *cactus/doc/running-in-aws.md at master · ComparativeGenomicsToolkit/cactus*. GitHub. https://github.com/ComparativeGenomicsToolkit/cactus/blob/master/doc/running-in-aws.md

Running in AWS — Toil 5.13.0a1 documentation. (n.d.). https://toil.readthedocs.io/en/latest/running/cloud/amazon.html

Introduction — Toil 5.13.0a1 documentation. (n.d.). https://toil.readthedocs.io/en/latest/running/introduction.html#jobstoreoverview