

Note 21, 2019 Thursday

Notes on installing miniconda and related software packages (GATK, CSP, pysam, BS-seekers)
Previous installation notes: Install.pysam.for.BS-seeker.Apr12.2019.pdf

Honors course HON3398O team project trouble shooting
Team.Project/4.Trouble.Shooting/Install.pysam.for.BS-seeker.Apr12.2019.pdf

<https://docs.conda.io/en/latest/miniconda.html>

Miniconda

Miniconda is a free minimal installer for conda. It is a small, bootstrap version of Anaconda that includes only conda, Python, the packages they depend on, and a small number of other useful packages, including pip, zlib and a few others. Use the `conda install command` to install 720+ additional conda packages from the Anaconda repository.

Flaherty, Shane
Thu 6/28/2018 3:30 PM

- Johnson, Allison B;
- Sun, Shuying

📧

Hi Allison,

Since your project uses Python and has dependencies, the preferred method of meeting those requirements in a shared environment such as LEAP is to perform the install in your home directory. Fortunately, there is a great tool that handles much of the complexity of this and can provide the groundwork for much ease in incorporating other Python packages into your workflow as you find tools you need. Miniconda is the tool I'm describing, and here is a process to get you started using it and steps for installing ruby and pysam:

Install miniconda2 (python 2.7.X):

On LEAP, get the latest Miniconda with Python 2.7.X version:

```
[me@login2 ~]$ cd ~  
[me@login2 ~]$ wget https://repo.continuum.io/miniconda/Miniconda2-latest-Linux-x86_64.sh
```

Set execute permissions on the installer:

```
[me@login2 ~]$ chmod +x Miniconda2-latest-Linux-x86_64.sh
```

Run it:

```
[me@login2 ~]$ ./Miniconda2-latest-Linux-x86_64.sh
```

During the install...

Accept the license agreement typing “yes” and hitting enter,
then hit enter to install in the default location (/home/<you>/miniconda2),
type “yes” at the prompt whether to add miniconda2 installation to your .bashrc

...

After the installation completes, log out and then back in to “activate” the new **conda** installation in your session (via environment pathing in your .bashrc):

List packages installed by default with the Miniconda install:

```
[me@login2 ~]$ conda list
```

Note that one of those items is a complete install of python 2.7.15 evidenced from this line:

```
python                2.7.15                h1571d57_0
```

Note you now have a robust environment within which many python packages can be added by a simple command - “**conda** install <package>”. Miniconda seeks to handle dependencies and greatly simplifies the process, without potentially impacting a global install of Python.

Install dependencies:

Your CSP and Harsh application require Ruby and pysam (python module):

```
[me@login2 ~]$ conda install ruby
[me@login2 ~]$ conda install pysam
[me@login2 ~]$ conda list | egrep "ruby|pysam"
```

You can run the “which” command to see the path for your newly-installed Python and Ruby:

```
[me@login2 ~]$ which python
[me@login2 ~]$ which ruby
```

You won’t be able to run a “which” command to see whether pysam is installed as it is a python module, but python directives issued within python scripts or through running python interactively can provide that information.

Thank you,

Shane Flaherty

High Performance Computing

Texas State University

shane@txstate.edu | 512-245-7866

From: Johnson, Allison B
Sent: Wednesday, June 27, 2018 8:22 PM
To: Flaherty, Shane <shane@txstate.edu>
Cc: Sun, Shuying <ssun@txstate.edu>
Subject: Software Installation Issues

Hello Shane,

I hope you are doing well and looking forward to your upcoming R&R. I am reaching out to you in hopes that you can help with the installation of a few needed software packages. I have downloaded/installed both CSP and Harsh, however they both appear to need additional packages to fully run.

CSP:

Working Directory: /home/abj15/Research.Project/CSP
Website to manual: <https://sites.google.com/site/hmatsu1226/software/csp>

This software package runs off the ruby command, however when I run the example command I receive the below error:

```
[abj15@login2 CSP]$ ruby CSP1.rb example/genotype.txt example/fragment.txt  
out/csp1.txt phase/input.txt phase/output.out 11 5  
-bash: ruby: command not found
```

Looking into how to make ruby an executable command it seems I need to download either rbenv or RVM and then install a version of ruby. I used <https://github.com/rbenv/rbenv/blob/master/README.md#installing-ruby-versions> as a reference to install both rbenv and ruby, but didn't seem to have much success. Any suggestions/insight is extremely appreciated.

Harsh:

Working Directory: /home/abj15/Research.Project/Harsh
Website to download: <http://genetics.cs.ucla.edu/harsh/download.html>
Website to manual: <http://genetics.cs.ucla.edu/harsh/manual.html>
- <http://genetics.cs.ucla.edu/harsh/manual.html#convertor>: this link will send you to the bottom of the manual page.

The issue I am having here is that though I installed pysam under /home/abj15/Research.Project/Harsh/pysam-master, I receive the below error when running the following command:

```
[abj15@login2 Harsh]$ python convertor.py -o Output/harsh.single.1mil.826T_1.out -v
826T_1/single.1mil.826T_1.bwa.vcf -b 826T_1/single.1mil.826T_1.bwa.bam
Traceback (most recent call last):
  File "convertor.py", line 2, in <module>
    import pysam
ImportError: No module named pysam
```

With this I attempted to load pysam as a module with:

```
[abj15@login2 Harsh]$ module load pysam
ModuleCmd_Load.c(213):ERROR:105: Unable to locate a modulefile for 'pysam'
```

and it doesn't seem to be currently installed:

```
[abj15@login2 Harsh]$ which pysam
/usr/bin/which: no pysam in
(/group/hon/hon3398o/0.course.files/software/FastQC/;/opt/openmpi/bin:/usr/lib64/qt-
3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/opt/ganglia/bin:/o
pt/ganglia/sbin:/usr/java/latest/bin:/opt/pdsh/bin:/opt/rocks/bin:/opt/rocks/sbin:/ho
me/abj15/bin)
```

Again any suggestion/direction/insight is extremely appreciated.

Thank you!
Bertie Johnson

Flaherty, Shane
Mon 4/2/2018 4:02 PM

- Sun, Shuying;
- **IT Research Computing Admins**

Hi Dr. Sun,

It looks as though the recommended method for getting gatk up in an environment is using the miniconda package (simplifies install and handles dependencies well). Please follow the process

outlined at this link for either python 2.7 or python 3 (your choice)

-> https://itrcstats.itrc.txstate.edu/wiki/index.php/LEAP_Python_Home_Directory_Install

Note that the software installs at the end (numpy, cython, h5py, scipy) are examples and don't have to be installed unless you need them.

Next, after logging out and back in to activate your miniconda install, do the following:

Add channels (be sure to add them in the order shown):

```
[me@login1 ~]$ conda config --add channels r
[me@login1 ~]$ conda config --add channels defaults
[me@login1 ~]$ conda config --add channels conda-forge
[me@login1 ~]$ conda config --add channels bioconda
```

Check what's installed:

```
[me@login1 ~]$ conda list
```

Install gatk:

```
[me@login1 ~]$ conda install gatk
```

The version of GATK installed under miniconda requires download of a licensed copy of GATK from the Broad Institute as noted here -> <https://bioconda.github.io/recipes/gatk/README.html>

I have paced a copy that corresponds to the one miniconda installs in your home directory at /home/s_s355/gatk/GenomeAnalysisTK-3.8-0-ge9d806836.tar.bz2 You'll need to decompress this archive and point the registration script for gatk to the unarchived target as shown below:

Change directory to the gatk directory in your home and decompress/untar the archive:

```
[me@login1 ~]$ cd ~/gatk
[me@login1 ~]$ bunzip2 GenomeAnalysisTK-3.8-0-ge9d806836.tar.bz2
[me@login1 ~]$ tar xvf GenomeAnalysisTK-3.8-0-ge9d806836.tar
```

Now register gatk, pointing to the jar file from the archive (replacing "<me>" with s_s355):

```
[me@login1 ~]$ gatk-register /home/<me>/gatk/GenomeAnalysisTK-3.8-0-ge9d806836/GenomeAnalysisTK.jar
```

This will "install" the jar file to /gpfs/home/<me>/miniconda3/opt/gatk-3.8/GenomeAnalysisTK.jar

You'll then point to the jar file like so (this test confirms operation) (replace "<me>" with s_s355):

```
[me@login1 ~]$ java -jar /gpfs/home/<me>/miniconda3/opt/gatk-3.8/GenomeAnalysisTK.jar --help
```

Thank you,

Shane Flaherty

High Performance Computing

Texas State University
shane@txstate.edu | 512-245-7866

From: Sun, Shuying
Sent: Sunday, April 01, 2018 6:04 PM
To: Flaherty, Shane <shane@txstate.edu>; IT Research Computing Admins <itrcadmins@txstate.edu>
Cc: Sun, Shuying <ssun@txstate.edu>
Subject: 2nd email regarding GATK requirement // Re: Java version 1.8

Hello Shane,

After I sent you the last email, I checked GATK further, and I found that, in addition to Java 1.8, GATK also requires other packages as listed below, and I may not list ALL the required packages, for detailed information about installing and running GATK, please see the **attached file "RADME.md"**.

* To run GATK:

- * **Java 8**
- * **Python 2.6 or greater (required to run the `gatk` frontend script)**
- * **Python 3.6.2**, along with a set of additional Python packages, is required to run some tools and workflows.

GATK uses the **[Conda]** (<https://conda.io/docs/index.html>) package manager to establish and manage the environment and dependencies required by these tools.

- * **R 3.2.5 (needed for producing plots in certain tools)**
- * **To build GATK:**
- * **A Java 8 JDK**
 - * **Git 2.5 or greater**
 - * **[git-lfs](<https://git-lfs.github.com/>) 1.1.0 or greater**. Required to download the large files used to build GATK, and test files required to run the test suite. Run `git lfs install` after downloading, followed by `git lfs pull` from the root of your git clone to download all of the large files, including those required to run the test suite. The full download is approximately 2 gigabytes. Alternatively, if you are just building GATK and not running the test suite, you can skip this step since the build itself will use git-lfs to download the minimal

set of large `\lfs\`
resource files required to complete the build. The test
resources will not be downloaded, but
this greatly reduces
the size of the download.
* **Gradle 3.1 or greater. We recommend using the `./gradlew`
script which will
download and use an appropriate gradle version
automatically (see examples below).**
* **R 3.2.5 (needed for running the test suite)**

From: Sun, Shuying
Sent: Sunday, April 1, 2018 5:33 PM
To: Flaherty, Shane; IT Research Computing Admins
Subject: Java version 1.8

Hello Shane,

I tried to install the GATK software package, which requires Java 1.8 as shown below. The current java at LEAP is 1.7.0_51. Could you please upgrade java to 1.8? Thanks.


<https://software.broadinstitute.org/gatk/download/>

Requirements

All POSIX operating systems (Unix, Linux, MacOSX etc) are supported. Microsoft Windows is *not* supported. The current version requires Java 1.8. Oracle Java and OpenJDK are both officially supported.

Shuying Sun
Department of Mathematics
Texas State University
601 University Drive
San Marcos, TX 78666, USA
Phone: 512-245-3422
Sun, Shuying
Sun 4/1/2018 6:04 PM

- Flaherty, Shane;
- **IT Research Computing Admins;**
- Sun, Shuying

 README.md
37 KB
Hello Shane,

After I sent you the last email, I checked GATK further, and I found that, in addition to Java 1.8, GATK also requires other packages as listed below, and I may not list ALL the required packages, for detailed information about installing and running GATK, please see the **attached file "RADME.md"**.

* To run GATK:

- * **Java 8**

- * **Python 2.6 or greater (required to run the `gatk` frontend script)**

- * **Python 3.6.2**, along with a set of additional Python packages, is required to run some tools and workflows.

GATK uses the **[Conda]**(<https://conda.io/docs/index.html>) package manager to establish and manage the environment and dependencies required by these tools.

- * **R 3.2.5 (needed for producing plots in certain tools)**

* **To build GATK:**

- * **A Java 8 JDK**

- * **Git 2.5 or greater**

- * **[git-lfs](<https://git-lfs.github.com/>) 1.1.0 or greater**. Required to download the large files used to build GATK, and

test files required to run the test suite. Run `git lfs install` after downloading, followed by `git lfs pull` from

the root of your git clone to download all of the large files, including those required to run the test suite. The

full download is approximately 2 gigabytes. Alternatively, if you are just building GATK and

not running the test

suite, you can skip this step since the build itself will use git-lfs to download the minimal set of large `lfs`

resource files required to complete the build. The test resources will not be downloaded, but

this greatly reduces

the size of the download.

- * **Gradle 3.1 or greater. We recommend using the `./gradlew` script which will download and use an appropriate gradle version automatically (see examples below).**

- * **R 3.2.5 (needed for running the test suite)**