Author: Bernadette Johnson
Updated: 21 August 2019

How to Install and Run Trinity
(for RNA-Seq De novo Assembly)

About this Protocol

This protocol is for users who are interested in assembling transcriptome data that is available from the NCBI SRA library. It is also useful for users who would like to set-up and run Trinity for the first time.

Challenge Level: Requires some working knowledge of Linux, and determination. Once run, Trinity can take several days to assemble transcript sequences. To know if Trinity is the right choice for your research, please visit and learn about Trinity via their GitHub (https://github.com/trinityrnaseq/trinityrnaseq/wiki).

BLUF

Computer requirements and recommendations:

- A Linux subsystem, or a Linux virtual box
- Additional hard drive, other than your Local Disk (C:) drive, with ~5 TB of free space
- Available memory RAM of at least 50GB, but preferably more
- Available CPU of at least 4, but preferable more

*In Windows 10, CPU and RAM information is available by CRTL+SHIFT+ESC >More details > Performance.

Total programs downloaded for this protocol:

notepad ++, SRA toolkit, zlib, bowtie2, g++, salmon, java 8, SAMtools, wget, python, numpy, jellyfish, CMake, Trinity

Information for working example:

Context: For this project, I am interested in assembling and comparing the testes transcriptomes of four fish: croaker, knifejaw, puffer, and rockbream. Their sequence data is available through the NCBI SRA library.
System: I will be using a Linux virtual box installed on Windows 10. I have an additional hard drive with 5 TB of free space, 120(out of 128) GB of RAM and 6 (out of 12) CPU logical processors that I will use to run Trinity. In my home directory, I have a folder named 'shared' which I will work in for most of the process and examples.

Let's get started

Preparing your computer beforehand:

1. Download Notepad ++ for writing scripts on Windows.

Notepad++ is a text editor and source code editor for use with Windows. It is great for freely editing scripts outside of the Linux system, and is user-friendly. It will also be useful for copying and pasting information such as NCBI accession numbers from your internet browser.

2. Create a new folder, such as "shared", on your additional storage device (not your Local Disk (C:) drive).

<u>Now we will work through the Linux terminal:</u>

3. Now create a symbolic link to a newly created folder, so you can locate it in the home directory. This allows you to see this folder when working in the Linux terminal.

```
> ln -s /mnt/e/shared/
```

Please note: In this case, the path to my folder “shared” is through the (E:) drive. You can check your path by navigating to “This PC” on your Windows 10 system and checking under the “Devices and drives” section. If I had my folder in the (D:) drive for example, I would instead use “/mnt/d/shared/”. This step is important to ensure you do not fill up your (C:) drive.

4. Update your Linux system. This will update the available packages but does not upgrade any packages. In the terminal window enter the command:

```
> sudo apt-get update
```

Please note: Using the command sudo for the first time for your instance will require you enter your password.

5. Then upgrade all packages to install the newest version. Make sure to update before you upgrade, updating allows the package manager to access information on available upgrades. This step can take about 20-40 minutes.

```
> sudo apt-get upgrade
```

Installing SRA Toolkit:

Additional SRA Toolkit help can be found here: (https://ncbi.github.io/sra-tools/install_config.html)

1. Download the SRA Toolkit.

These instructions are for the latest version, but newer versions might be available, so please check.

```
> wget "http://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/current/sratoolkit.current-centos_linux64.tar.gz"
```

2. Navigate to where the downloaded file is located.

```
> cd path/to/the/downloaded/file
```

3. Unpack the downloaded zipped file:

```
> tar -xzf sratoolkit.current-centos_linux64.tar.gz
```

4. Now we need to add the fastq-dump program to your system path. This allows you to use the program from any directory, even ones outside the downloaded folder. To do so, navigate to your home directory, and then list all files:

```
> cd
> ls -a
```

5. Locate the bashrc file to edit.

```
>nano .bashrc
```

6. Specifically, we want to add the folder named 'bin' in the downloaded SRA toolkit folder, because it contains the files needed to run fastq-dump. ***Without deleting or editing other parts of the .bashrc file***, scroll all the way down to the bottom of the file and add the following:

```
export PATH=$PATH:~/shared/sratoolkit/bin/
```

Please note: $PATH: echos back the path already in the computer; *~/shared/sratoolkit/bin/*: adds the path of a folder named 'bin'. *This path is specific to where you have placed your sratoolkit folder and must start from your home directory.* In my case, starting from the home directory (~) I have a folder named 'shared', with a subfolder 'sratoolkit' and 'bin' where the program fastq-dump is located.

7. Now, exit (CTRL+X) and save (‘y’ then ENTER). Now, fastq-dump will run from any directory.

Running SRA Toolkit:

1. Create folders where the SRA files will download to.

In "shared", I am creating a folder "bernadette" with sub-folders of the species I am interested in downloading SRA files for. These folders are named "croaker," "knifejaw," "puffer," and "rockbream". Within each of the four species folders I have two more folders "testes" and "ovaries," since I am interested in downloading testes and ovaries transcriptomes. It is best to organize files ahead of time, as SRA names can be difficult to organize after downloaded.

2. Write a script for running fastq-dump using notepad ++.

Here is part of my script:

```
#!/bin/bash
#spotted_knifejaw_testes
fastq-dump --defline-seq '@$sn[_$rn]/$ri' --defline-qual '+$sn[_$rn]/$ri' --split-files -O
~/shared/bernadette/sra_download/knifejaw/testes SRR5666978 SRR5666989 SRR5667091
#spotted_knifejaw_ovaries
fastq-dump --defline-seq '@$sn[_$rn]/$ri' --defline-qual '+$sn[_$rn]/$ri' --split-files -O
~/shared/bernadette/sra_download/knifejaw/ovaries SRR5666719 SRR5666724 SRR5666739

#rockbream_testes
fastq-dump --defline-seq '@$sn[_$rn]/$ri' --defline-qual '+$sn[_$rn]/$ri' --split-files -O
~/shared/bernadette/sra_download/rockbream/testes SRR2886786
#rockbream_ovaries
fastq-dump --defline-seq '@$sn[_$rn]/$ri' --defline-qual '+$sn[_$rn]/$ri' --split-files -O
~/shared/bernadette/sra_download/rockbream/ovaries SRR2886787
```

Please note:

1. Instructions and options for running fastq-dump can be found through your Linux terminal:

> fastq-dump –h

Important options you should specify:

-O: the output folder, where do you want fastq-dump to download the files? This should be your work folder on your additional storage device. SRA files are large and can crash your computer if not enough space is available. For this reason, to protect your main (C:) *drive, you should consider using an external drive for your main working folders. Here I specify my output as "-O path/to/folder"*

where my home directory is set up on the external drive. I also have the reads deposited into labelled folders, making it easier for me to sort them out later.

--defline-seq '@$sn[_$rn]/$ri' --defline-qual '+$sn[_$rn]/$ri': used to reformat an SRA file header into one compatible with Trinity.

--split files: used to split pair reads into two files for fwd and reverse reads.

2. Saving your script:

*When using notepad ++, you will want to save with a .sh format extension. Once you save a script for the first time, it might not be in a format that Linux can read. To fix this, navigate to the file in the Linux terminal and edit it through the terminal using nano. (***nano myscript.sh***). Enter a space anywhere then delete it (we just want to prompt a new save). Exit (CTRL+ X) and 'y'. Then nano will ask, "File Name to Write: myscript.sh". We want to save it under a different format, holding down (ALT), hit the key 'm' to toggle between options until there is no specific format [DOS] or [Mac], then hit (ENTER).*

4. If you already have downloaded and saved the SRA files (.sra) separately:

You can use fastq-dump to convert the .sra files into .fasta files from the local computer, instead of downloading them again, which was relatively faster. Start by copying .sra files into the folders where you want the .fasta files to save to. Then, being sure to enter your own SRR# below, use the command:

```
> fastq-dump --defline-seq '@$sn[_$rn]/$ri' --defline-qual '+$sn[_$rn]/$ri' --split-files SRR#.sra
```

5. Move the stored cache files out of C: drive:

The files will now download onto the specified output file but, the original output folder will store cache files about ~2GB per SRR file downloaded. In my case, the original file is easily found by searching for "ncbi" within (C:). I recommend moving all the SRA files to the output folder. It is also possible to set the default output folder within fastq-dump, however the version I am working with presented an error that I was not able to circumvent.

6. Make note of progress and time requirements:

This process will take several hours (or even days) to run. The status of the script can be checked by opening an additional terminal window and using the command "top" which lets you see what is running, and for how long.

This is an example of what will be written on the terminal window as the process is working:

```
joneslab@DESKTOP:~/shared/bernadette/sra_download$ ./downloadSRA_forme.sh
Read 20378822 spots for SRR5666978
Written 20378822 spots for SRR5666978
Read 22039452 spots for SRR5666989
Written 22039452 spots for SRR5666989
Read 20160479 spots for SRR5667091
Written 20160479 spots for SRR5667091
Read 62578753 spots total
Written 62578753 spots total
```

7. Encountering errors*:*

The SRA Toolkit works very hard and is very stubborn and will retry most operations until they succeed or we eventually time out. The cause of some errors might include network connections, so ensuring a stable internet connection might prevent the formation of most errors. If the tool completes and generates a report of the number of reads, etc., then it was successful.

Installing Trinity:

Trinity requires the installation of several programs. Here is how to install them for the version 2.8.5 of Trinity, please also reference the official Trinity instructions as changes since this document has been written are likely to have occurred:

1. zlib

> sudo apt install zlib1g-dev

2. bowtie2

> sudo apt install bowtie2

3. g++

> sudo apt install g++

4. salmon

    1. Download salmon from their Github: (https://github.com/COMBINE-lab/salmon/releases).
    2. Navigate to the downloaded folder, and unzip using the command:

        > tar -xzf *.tar.gz

    Please note: The command *sudo apt install salmon* will download the old version. Do not do this! Trinity will not work with this old version. If you have already done this, change directory to ~/usr/bin and sudo rm ~/usr/bin/salmon.

5. Java 8

    1. Navigate to the java website and look for a Java installation that will work for your computer. Most likely, the Linux 64-bit installation instructions for Java will be what you need (https://java.com/en/download/help/linux_x64_install.xml#download).
    2. Follow the steps to download and install Java. You will need to click the ‘See all Java downloads’ button, and find and install the linux version or click https://www.java.com/en/download/manual.jsp.
    3. When saving, save it under the simple name ‘java’ and to you new folder we created, the equivalent to the one I have named on my computer "shared", which is located on your additional storage device (not your Local Disk (C:) drive).

6. SAMtools

```
> sudo apt install samtools
```

7. Python 2.7

```
> curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
> python get-pip.py
> python -m pip install --user numpy
```

Please note: Python 2.7 will become outdated. It is recommended that you install Python 3 if you are reading this script after January 2020.

8. Jellyfish

```
> sudo apt install jellyfish
```

9. CMake

```
> sudo apt install cmake
```

10. Download the source code (zip) of the latest Trinity from their GitHub: (https://github.com/trinityrnaseq/trinityrnaseq/releases)

Please note:

For some reason the tar.gz version of Trinity Release v2.8.5 has a bug that will not compile on this computer and caused me to completely uninstall and reinstall everything. The zip version does work though. So, choose at your own risk.

11. Unzip the downloaded file via right-click ‘Extract all’ on your Windows side.

12. Now we need to add these programs to your PATH. This will allow Trinity to locate the programs itself to run, otherwise it will be unable to search through your file structure to find everything.

    1. Move to your home directory:

    ```
    > cd
    ```

    2. Open up your .bashrc file to make appendments.

    ```
    > nano .bashrc
    ```

    3. Now, let’s add the programs we just downloaded to your path.

1. Open a separate terminal. We need to start by determining where all the programs have been downloaded to. This might get a bit sticky. Here are recommended folders that you should check (and where my files are located). If you cannot find the files try the command, ‘dpkg -L *package name’.*

    1. zlib: This is installed somewhere but it runs by itself and you won’t need to add it to your PATH.
    2. bowtie2: This is installed in the /usr/bin.
    3. g++: This is installed in the /usr/bin.
    4. salmon: This is installed where you specified it when downloading off their website. Most likely you will find it in your home directory.
    5. Java 8: This is installed where you specified it when downloading off their website. Most likely you will find it in your home directory.
    6. SAMtools: This is installed in the /usr/bin.
    7. Python 2.7/3: This is installed in the /usr/bin.
    8. Jellyfish: This is installed in the /usr/bin.
    9. CMake: This is installed in the /usr/bin.
    10. Trinity: This is installed where you specified it when downloading off their website. Most likely you will find it in your home directory.

Please note: The /usr/bin is two parent folders out of your home directory, and can be found by starting from your home directory ‘cd’ and then using ‘cd ..’ twice more.

2. Without editing any other part of the .bashrc file, please scroll all the way down to the bottom and add the following (making sure to change the directory names to what your directory names are!):

```
#Bunch of Programs
export PATH=$PATH:/usr/bin
#Java
export PATH=$PATH:~/shared/jre1.8.0_221/bin
#Salmon
 export PATH=$PATH:~/shared/salmon-latest_linux_x88_64/bin
#Trinity
export TRINITY_HOME=~/shared/trinityrnaseq-Trinity-v2.8.5
```

(CTRL+X) to exit script, ‘y’ to save changes

13. Restart your computer. For some anomalous reason, Trinity will have trouble in the next step unless this is done.

14. Compile Trinity:

```
> cd shared/trinityrnaseq-Trinity-v2.8.5
>make
```

Please note: This is part of what you should see when Trinity is successfully compiled:

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Performing Unit Tests of Build
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Inchworm:              has been Installed Properly
Chrysalis:             has been Installed Properly
QuantifyGraph:         has been Installed Properly
GraphFromFasta:        has been Installed Properly
ReadsToTranscripts:    has been Installed Properly
parafly:               has been Installed Properly
```

15. Upgrade again:

```
>sudo apt upgrade
```

16. Restart terminal! (Exit and reopen the window.)

17. Move back into the Trinity folder directory and run the sample data to make sure Trinity works:

```
> cd shared/trinityrnaseq-Trinity-v2.8.5/sample_data/test_Trinity_Assembly
> ./runMe.sh
```

Please note: This is part of what you should see when Trinity is successful:

```
##########################################################################
Trinity assemblies are written to /mnt/d/shared/trinityrnaseq-Trinity-v2.8.5/sample_data/test_Trinity_Assembly/trinity_out_dir/Trinity.fasta
##########################################################################
...
##### Done Running Trinity #####
```

Running Trinity:

Start by selecting the files you want to use for the assembly. These files would be from the SRA Toolkit. **However, I will only use the testes transcriptome for assemblage. Ovaries and testes can have different isoforms for the same genes possibly complicating the assemblage. In this case, select the testes for assemblage, then map the testes or ovaries to the assemblage.** Once started, a successful Trinity run will take days to complete. It is best to refrain from using your computer during this time. It is also recommended you pause all automatic updates for your computer.

Here is part of my script:

```
#knifejaw
./Trinity --trimmomatic --seqType fq --max_memory 120G \
--left
/home/joneslab/shared/bernadette/sra_download/knifejaw/testes/SRR5666978_1.fastq,/home/joneslab/shared/bernadette/sra_download/knifejaw/testes/SRR5666989_1.fastq,/home/joneslab/shared/bernadette/sra_download/knifejaw/testes/SRR5667091_1.fastq \
--right
/home/joneslab/shared/bernadette/sra_download/knifejaw/testes/SRR5666978_2.fastq,/home/joneslab/shared/bernadette/sra_download/knifejaw/testes/SRR5666989_2.fastq,/home/joneslab/shared/bernadette/sra_download/knifejaw/testes/SRR5667091_2.fastq \
--CPU 6
```

Important options you should specify:

--max_memory 120G: The max_memory set should be based on the available memory for your computer. Mine is 128 GB of RAM, I am leaving 8 GB of RAM as a safety precaution for my computer to run other processes. During this time however, I am limiting what I do on the computer.

--left /--right: The left and right reads must be paired up sequentially. Meaning, if SRR5666978_1.fastq is mentioned first for the left, then its matched pair SRR5666978_2.fastq must be first for the right. Also, the entire path must be written out, including '/home/' instead of '~/'. If you have more than one fastq pair, then they must be separated with only a comma- no space, no enter.

--CPU 6: The allotted CPU should be based on the number of cores your computer has. Mine has a total of 12. I set my CPU to 6, allowing another 6 as a safety precaution for my computer to run other processes (similar to why I set my max_memory below my total memory). Additionally, for reasons I cannot explain, the CPU needs to be an even number. Meaning, I could have set my CPU to 6 or 8, but not 7.

--trimmomatic: I am using the default parameters for trimmomatic. For additional options please visit their website: (http://www.usadellab.org/cms/?page=trimmomatic)

For other options, please read: (https://github.com/trinityrnaseq/trinityrnaseq/wiki/Running-Trinity)

--normalize_by_read_set: Can be used if you have memory/RAM limitations (not used in my example). It will normalize your data separately for each pair of fastq files, and then run an additional normalization that combines the individual normalized reads.

Dealing with Potential Errors:

Chances are while running TRINITY you will face many errors. Typically, specific solutions for these can be searched for on the internet. However, it is important to note, when TRINITY encounters an error and is cancelled, files in the output folder such as "trinity_out_dir" are not deleted. When attempting to fix any error, make sure you delete output files after an attempted correction so TRINITY will run based on new attempts and does not try to reuse old files.

Another common error is that the starting fastq or fasta read files are improperly labelled. Start by checking the headers on your read file with the “head -n50 *filename.fastq*”. If the headers of your read files are not labelled like “@1_1/2, @1_2/2” then Trinity will not run properly. If you are downloading data from the SRA website, make sure to use the **--defline-seq '@$sn[_$rn]/$ri'** option when using fastq-dump. If the data is not from SRA toolkit, you will need to change the header format of each of your reads before running Trinity. If this is the case, save a copy of the original data elsewhere, and attempt:

```
zcat input.fastq | awk '{{print (NR%4 == 1) ? "@1_" ++i "/2": $0}}' | gzip -c > output.renamed.fastq
```

If you see this, chances are your TRINITY run was a success:

```
################################################################################
Trinity assemblies are written to /mnt/e/shared/trinityrnaseq-Trinity-v2.8.4/trinity_out_dir/Trinity.fasta
################################################################################
```

Congratulations on your Trinity installation and run!