**December 2nd, 2019**

**Revisit my python learning process**

<https://support.bioconductor.org/p/77021/>

I have a place to hold leaning note at: /ddn/gs1/home/li11/learningPython.It turns out that I have lost track!!

I would like to teat my “titanic” learning experience first /ddn/gs1/home/li11/learningPython/python4Titanic, which is a symbolic link to /ddn/gs1/home/li11/myGit/workingWithYicheng/Titanic/. The project is under version control with github: <https://github.com/2waybene/workingWithYicheng/tree/master/Titanic>

Well, stuck on python csv.reader(), and got it taken care of via pythondoc: <https://docs.python.org/3/library/csv.html>. And, while working on this, I found out that “pip install” stops supporting python2.7 starting January 1st, 2020.

Moving quite slow with the python programming!

Need to find a solid teaching/learning portal.

**Working with Salmon quantification results**

A good edgeR sample code: <https://github.com/griffithlab/rnaseq_tutorial/blob/master/scripts/Tutorial_edgeR.R>

I am looking for packages for normalized data, which is TPM from Salmon quantification process.

**December 3rd, 2019**

**Getting homolog genes human and mouse**

I used Ensembl for such a conversion to get the most current version of such look up table.

<http://www.ensembl.info/2009/01/21/how-to-get-all-the-orthologous-genes-between-two-species/>

Gene annotation

I used this link: <https://bioconductor.org/help/course-materials/2014/SeattleOct2014/B02.4_Annotation.html>

**December 4th, 2019**

**Shared a scripture with Jac**

This is what the Sovereign Lord, the Holy One of Israel, says: “In repentance and rest is your salvation, in quietness and trust is your strength, but you would have none of it.” (Isiah 30:15)

**December 5th, 2019**

**Learning tximport R package**

This package’s document is here: <https://bioconductor.org/packages/release/bioc/vignettes/tximport/inst/doc/tximport.html>

Encountered road block, will come back in some other time

**December 9th, 2019**

**Learning GPU basic**

It is announced that Apple MacBookPro 16 comes with AMD Radeon GPU. I will choose the highest end AMD Radeon Pro 5500M with 24 compute units. It is designed for 3D and game development, but I am going to use it for GPU computing and deep learning programming.

<https://www.notebookcheck.net/AMD-Radeon-Pro-5500M-GPU-Benchmarks-and-Specs.442754.0.html>

Since the Mac is equipped with AMD graphical memory, one has to learn something different from Nvidia CUDA code. The Georgia Tech has this: <http://gpuocelot.gatech.edu/>, but it seems discontinued already.

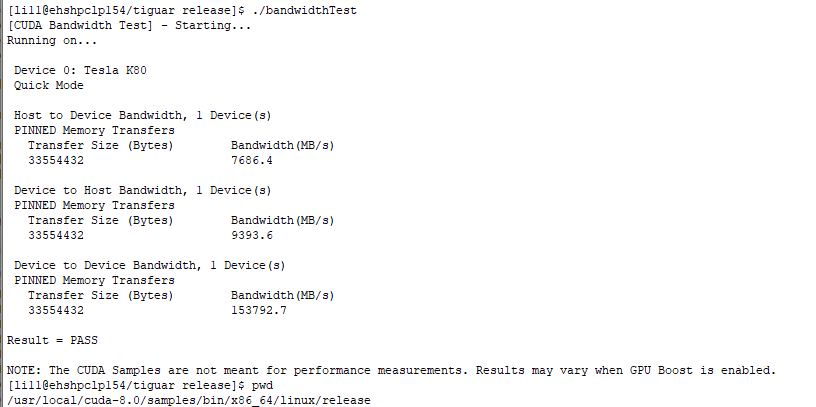
It seems that Dell 7910 comes with Nvidia memory and the newest driver is Quadro K5200 (but not the one I am using)

<https://www.nvidia.com/en-us/geforce/forums/game-ready-drivers/13/197070/dell-tower-precision-7910-cuda-65/>

Neural network on AMD GPU: <https://towardsdatascience.com/train-neural-networks-using-amd-gpus-and-keras-37189c453878>, but it only support Linux OS. No Mac is supported.

So much knowledge out there and it makes people hard to keep up: <https://www.fossmint.com/benchmark-apps-to-measure-mac-performance/>

At least, I can test CUDA GPU installation on the Tigura: <https://devtalk.nvidia.com/default/topic/1027653/how-do-i-check-if-i-install-cuda-and-cudnn-successfully-/>



I am working on putting a new tab for IndexCalculation

It involves “directory input” capbability.

One suggests to use shinyFiles package here: <https://community.rstudio.com/t/shiny-directory-input/29160/4>

**December 10th, 2019**

**Keep on with GPU basic**

To find out the video card on the linux CentOS7: <https://www.cyberciti.biz/faq/linux-tell-which-graphics-vga-card-installed/> , this is a very good post. Well, it turns out that I may need to purchase a Nvidia video card. It will be installed on the Dell Precision 7910

**December 11th, 2019**

**Verifying CUDA capability on a Linux server**

There is one server Tiguar equipped with CUDA core, and it can be a good test case.

Here is a few very useful help website: [https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#post-installation-actions](https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html" \l "post-installation-actions)

**Validate that we have nvidia cores:**

[li11@ehshpclp154/tiguar ~]$ **lspci | grep -i nvidia**

06:00.0 3D controller: NVIDIA Corporation GK210GL [Tesla K80] (rev a1)

07:00.0 3D controller: NVIDIA Corporation GK210GL [Tesla K80] (rev a1)

0a:00.0 3D controller: NVIDIA Corporation GK210GL [Tesla K80] (rev a1)

0b:00.0 3D controller: NVIDIA Corporation GK210GL [Tesla K80] (rev a1)

0e:00.0 3D controller: NVIDIA Corporation GK210GL [Tesla K80] (rev a1)

0f:00.0 3D controller: NVIDIA Corporation GK210GL [Tesla K80] (rev a1)

12:00.0 3D controller: NVIDIA Corporation GK210GL [Tesla K80] (rev a1)

13:00.0 3D controller: NVIDIA Corporation GK210GL [Tesla K80] (rev a1)

**There are some post-installation actions:**

[li11@ehshpclp154/tiguar ~]$ **nvcc --version**

nvcc: NVIDIA (R) Cuda compiler driver

Copyright (c) 2005-2015 NVIDIA Corporation

Built on Tue\_Aug\_11\_14:27:32\_CDT\_2015

Cuda compilation tools, release 7.5, V7.5.17

**Here are my current ome post-installation actions:**

[li11@ehshpclp154/tiguar ~]$ **nvcc --version**

nvcc: NVIDIA (R) Cuda compiler driver

Copyright (c) 2005-2015 NVIDIA Corporation

**December 12th, 2019**

**Refreshing Linux admin related information**

**Get to know os version**

cat /etc/os-release

<https://www.cyberciti.biz/faq/how-to-check-os-version-in-linux-command-line/>

**Installing PyTorch on a Linux server (home directory)**

There is one server Tiguar equipped with CUDA core, and it can be a good test case. I requested help from our systemadmin,

<https://developer.nvidia.com/cuda-downloads?target_os=Linux&target_arch=x86_64&target_distro=CentOS&target_version=7&target_type=rpmlocal>

wget <http://developer.download.nvidia.com/compute/cuda/10.2/Prod/local_installers/cuda-repo-rhel7-10-2-local-10.2.89-440.33.01-1.0-1.x86_64.rpm>

sudo rpm -i cuda-repo-rhel7-10-2-local-10.2.89-440.33.01-1.0-1.x86\_64.rpm

sudo yum clean all

sudo yum -y install nvidia-driver-latest-dkms cuda

sudo yum -y install cuda-drivers

After the installation, there are a few test protocol to ensure proper/complete installation:

<https://docs.nvidia.com/cuda/cuda-installation-guide-linux/index.html#post-installation-actions>