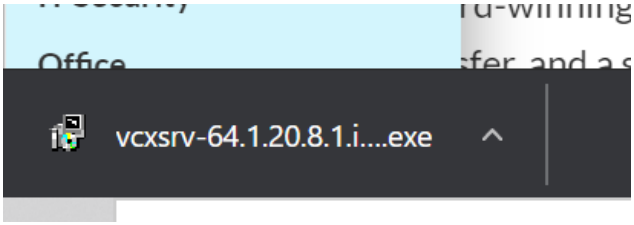
TRIUMF - ISSP:
Cedar Walkthrough
Run EDA Script on ComputeCanada

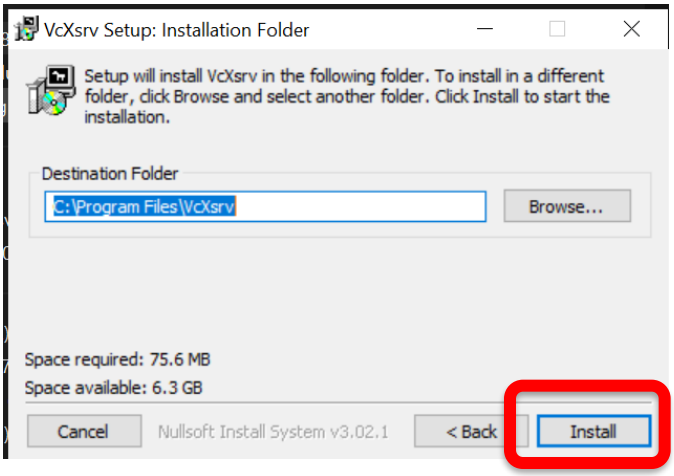
Task 1 – Download and Configure VCXsrv

1. **Download** VCXsrv from
   https://sourceforge.net/projects/vcxsrv/

2. **Run** .exe file and click **Yes**



3. **Click** through GUI and **Install** (Keep All Default Settings)



4. Open **Ubuntu LTS (Not WSL)**

5. **Run** the Following Command:

```
sudo apt-get install xorg openbox -y
```

Task 2 – Start Interactive Job on Cedar

1. **SSH** into Cedar Compute Canada using the following command:

```
ssh -Y -X <username>@cedar.computecanada.ca
```

2. **Type** your Cedar Password

```
abdul@DESKTOP-KQDKFDB:~$ ssh -Y -X aa16@cedar.computecanada.ca
aa16@cedar.computecanada.ca's password:
Warning: No xauth data; using fake authentication data for X11 forwarding.
Last login: Thu Nov 12 22:25:19 2020 from 207.23.215.218
================================================================================
Welcome to Cedar!

For information see: https://docs.computecanada.ca/wiki/Cedar
Email support@computecanada.ca for assistance and/or to report problems.
================================================================================

[aa16@cedar5 ~]$
```

3. **cd** to your working directory

ex. `cd projects/rpp-blairt2k/aa16/`

4. **Clone** EDA repository to your directory using “git clone”

```
[aa16@cedar5 aa16]$ git clone https://github.com/Alif-B/TriumfCNN.git
```

5. **Run** the following command to Start Interactive Job

```
salloc --time=5:00:0 --ntasks=4 --mem-per-cpu=125G --account=rpp-blairt2k --x11
```

```
[aa16@cedar5 TriumfCNN]$ salloc --time=5:00:0 --ntasks=4 --mem-per-cpu=125G --account=rpp-blairt2k --x11
salloc: Granted job allocation 55246339
salloc: Waiting for resource configuration
salloc: Nodes cdr[747,762,769,792] are ready for job
[aa16@cdr747 TriumfCNN]$
```

Task 3 – Singularity Container

1. **Run** setup.sh file (provided) to setup container

```
aa16@cdr833 aa16]$ bash setup.sh
```

Note: Done correctly your environment should look like as follows:

```
Singularity> cd Triumf2/
Singularity> ls
TriumfCNN
Singularity> cd ..
Singularity> ls
EventDisplay.py  Triumf2  event998.npz  mpmt_full_geo.npz  setup.sh  test
Singularity> cd Triumf2/
Singularity> cd TriumfCNN/
```

Task 4 – Create Python Environment

Note: Confirm you are in an Interactive Job before following these steps

2. **Purge** modules by running the following command:

```
module purge
```

3. **Load** all modules necessary to run script

```
module load intel/2018.3
module load openmpi/3.1.2
module load fftw-mpi/3.3.8
module load python/3.6.3
module load scipy-stack
module load mpi4py/3.0.3
module load hdf5-mpi/1.10.3
```

```
[aa16@cdr747 TriumfCNN]$ module purge
The following modules were not unloaded:
  (Use "module --force purge" to unload all):

  1) StdEnv/2016.4   3) imkl/11.3.4.258   5) icc/.2016.4.258     7) intel/2016.4
  2) nixpkgs/16.09   4) gcccore/.5.4.0    6) ifort/.2016.4.258   8) openmpi/2.1.1
[aa16@cdr747 TriumfCNN]$ module load intel/2018.3

Inactive Modules:
  1) openmpi/2.1.1

The following have been reloaded with a version change:
  1) gcccore/.5.4.0 => gcccore/.7.3.0       3) ifort/.2016.4.258 => ifort/.2018.3.222     5) intel/2016.4 => intel/2018.
3
  2) icc/.2016.4.258 => icc/.2018.3.222     4) imkl/11.3.4.258 => imkl/2018.3.222

[aa16@cdr747 TriumfCNN]$ module load openmpi/3.1.2

Activating Modules:
  1) openmpi/3.1.2

[aa16@cdr747 TriumfCNN]$ module load fftw-mpi/3.3.8
[aa16@cdr747 TriumfCNN]$ module load python/3.6.3
[aa16@cdr747 TriumfCNN]$ module load scipy-stack
[aa16@cdr747 TriumfCNN]$ module load mpi4py/3.0.3
[aa16@cdr747 TriumfCNN]$ module load hdf5-mpi/1.10.3
```

4. **Create** Virtual Environment

```
virtualenv --no-download python_env
```

```
[aa16@cdr747 TriumfCNN]$ virtualenv --no-download python_env
Using base prefix '/cvmfs/soft.computecanada.ca/easybuild/software/2017/Core/python/3.6.3'
New python executable in /project/6008045/aa16/Triumf2/TriumfCNN/python_env/bin/python
Installing setuptools, pip, wheel...done.
```

5. **Source** into Virtual Environment

```
source python_env/bin/activate
```

```
TriumfCNN]$ source python_env/bin/activate
(python_env) [aa16@cdr747 TriumfCNN]$
```

6. **Upgrade** pip install

```
pip install --no-index --upgrade pip
```

7. **Install** all dependencies required for EDA script *(any additional you may require can be installed as well)*

```
pip install h5py
pip install mercurial
pip install pytest
```

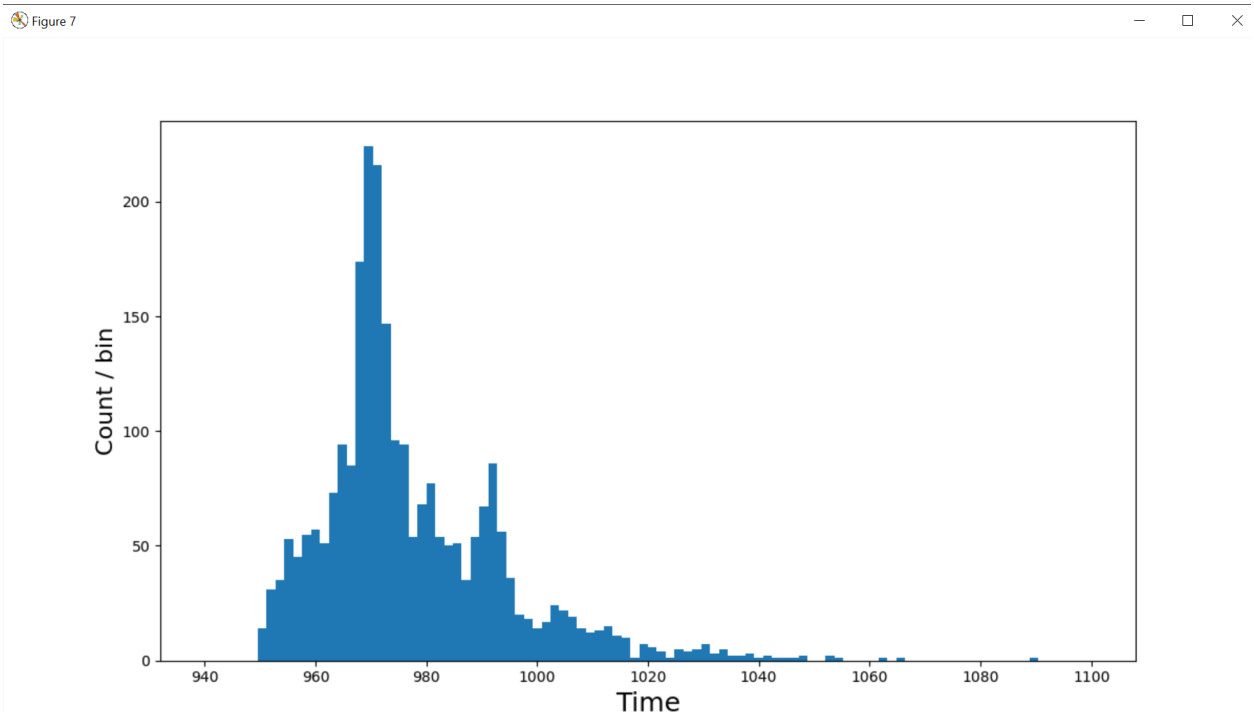
8. **Run** EventDisplay.py Script Successfully

```
python EventDisplay.py
```

```
(python_env) [aa16@cdr747 TriumfCNN]$ python EventDisplay.py
R= 370.0958251953125 H= 1034.8980712890625
min_x= -370.0958251953125 max_x= 370.0958251953125 diameter= 740.191650390625
min_z= -370.0958251953125 max_z= 370.0958251953125 diameter= 740.191650390625
min_y= -517.4490356445312 max_y= 517.4490356445312 height= 1034.8980712890625
Fontconfig warning: ignoring C.UTF-8: not a valid language tag
```

Note: x11 server is used so you are able to see your EDA graphs. VCXsrv will open and show all your figures when the script is run.

Ex.



GitHub Repository

https://github.com/Alif-B/TriumfCNN.git

Contact Information

Email: a.abdullah9350@gmail.com