gpuPanda_tutorial

August 5, 2020

1 Running Panda and Lioness using the Graphical Processing Unit (GPU)

1.0.1 Author:

Daniel Morgan*.

*Channing division of network medicine, Brigham's and Women hospital and Harvard Medical School, Boston, MA.

1.1 Introduction

PANDA [1] estimates gene regulatory networks using information from Transcription Factor (TF) Protein-Protein Interaction (PPI), gene coexpression, and TF DNA binding motifs. At its core, PANDA computes similarites between the three sources of information to infer the weight of regulatory edges, using matrix mulitplication. To accelerate the network inference, Graphical Processing Units (GPUs) can be used to perform matrix multiplication in an efficient manner.

1.2 1. Setup Environment: download, install & import latest packages

```
[0]: import os
     import s3fs
     import pandas as pd
     import numpy as np
     from psutil import *
```

```
[0]: !nvcc --version
```

1.2.1 1.1 Configure cupy for installed Cuda version

1. use cuda version based on above release (*e.g.* release 10.1= cuda101)
2. install cupy version 7.4.0 or above

```
[2]: !pip install cupy-cuda101==7.4.0
     import cupy as cp
     # print(cp.__version__)
     !pip freeze | grep cupy
```

```
Collecting cupy-cuda101==7.4.0
  Downloading https://files.pythonhosted.org/packages/e3/ea/bef39342236e3b
89587443842f0fdee48a468e9420ed6aad8a481bca02a9/cupy_cuda101-7.4.0-cp36-cp36m-man
ylinux1_x86_64.whl (361.9MB)
    || 361.9MB 43kB/s
Requirement already satisfied: fastrlock>=0.3 in
/usr/local/lib/python3.6/dist-packages (from cupy-cuda101==7.4.0) (0.4)
Requirement already satisfied: numpy>=1.9.0 in /usr/local/lib/python3.6/dist-
packages (from cupy-cuda101==7.4.0) (1.18.3)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.6/dist-
packages (from cupy-cuda101==7.4.0) (1.12.0)
Installing collected packages: cupy-cuda101
  Found existing installation: cupy-cuda101 6.5.0
    Uninstalling cupy-cuda101-6.5.0:
      Successfully uninstalled cupy-cuda101-6.5.0
Successfully installed cupy-cuda101-7.4.0
cupy-cuda101==7.4.0
```

##1.2 GPU and CPU info

```
[ ]: !nvidia-smi
```

```
[ ]: !cat /proc/cpuinfo
```

1.2.2   1.3 Clone netZoo

```
[0]: !rm -rf netZooPy
```

```
[0]: !git clone --single-branch --branch devel https://github.com/netZoo/netZooPy.git
     os.chdir('netZooPy')
     !pip install -e .
     os.chdir('..')
```

```
[0]: import netZooPy
     from netZooPy.panda import Panda
     from netZooPy.lioness import Lioness
```

1.3   2. Load test data

First, we get the data from the GRAND database. This can be done directly through pandas.

```
[0]: LCL_ppi=pd.read_csv('https://granddb.s3.amazonaws.com/cells/ppi/LCL_ppi.
     ↪txt',sep='\t',header=None)
     LCL_expression=pd.read_csv('https://granddb.s3.amazonaws.com/optPANDA/
     ↪expression/Hugo_exp1_lcl.txt',sep='\t',header=None)
     LCL_motif=pd.read_csv('https://granddb.s3.amazonaws.com/gpuPANDA/
     ↪Hugo_motifCellLine_reduced.txt',sep='\t',header=None)
     LCL_ppi.to_csv('LCL_ppi.txt',sep='\t',index=False)
     LCL_expression.to_csv('Hugo_exp1_lcl.txt',sep='\t',index=False)
     LCL_motif.to_csv('Hugo_motifCellLine_reduced.txt',sep='\t',index=False)
```

```
[0]:  # os.getcwd()
      expression_data='Hugo_exp1_lcl.txt'
      motif_data='Hugo_motifCellLine_reduced.txt'
      ppi_data='LCL_ppi.txt'
```

1.4   3. Run Panda with GPU & precision flags

The precision flags allow to compute the network in sinlgle (7 decimal digits) or double precision (15 decimal digits). Although double precision networks are more accurate, single precision can be useful to free memory on devices and to go faster. It is a tradeoff that depends on the application

```
[0]:  panda_obj = Panda(expression_data, motif_data,␣
      ↪ppi_data,computing='gpu',precision='single',save_tmp=False,save_memory =␣
      ↪False, remove_missing=True, keep_expression_matrix = True,modeProcess =␣
      ↪'intersection')
```

```
[0]:  panda_obj.save_panda_results('single_cpu_panda.txt')
```

1.5   4. Run LIONESS with GPU

LIONESS [2] calls PANDA to estimate regulatory networks for each sample. We can use the GPU acceleration capabilities to estimate sample-specific networks through setting the `computing` flag to gpu.

```
[0]:  lioness_obj = Lioness(panda_obj,computing='gpu',start=1, end=5,␣
      ↪save_dir='lioness_output', save_fmt='txt')
```

```
[ ]:  lioness_obj.panda_network.shape
```

1.5.1   References

[1] Glass, Kimberly, et al. "Passing messages between biological networks to refine predicted interactions." PloS one 8.5 (2013).

[2] Kuijjer, Marieke Lydia, et al. "Estimating sample-specific regulatory networks." iScience 14 (2019): 226-240.