Deep learning Augmented RNA-seq analysis of Transcript Splicing

Demo 1: Installation & Basic Usage

Zijun Zhang <zj.z@ucla.edu>

UCLA

March 17, 2018

1. Installation

- Download the Darts software
  > git clone git@github.com:zj-zhang/DARTS.git

- Install Darts_BHT and Darts_DNN
  > cd Darts_BHT
  > make install
  > cd ../Darts_DNN
  > make install

- Install Keras and Theano (recommend Anaconda); you will also need to change Keras backend to Theano, see here.
  > conda install -c conda-forge keras
  > pip install theano

- Add Darts to your environment variables

  > vim ~/.bash_profile

- .. paste the following code to the end of the file:

  export PATH=$HOME/.darts:$PATH

- Then source it

  > source ~/.bash_profile

2. Using Darts_BHT

- In a new directory (say, demo_1), download the demo data to local folder (11M) and unzip it

  > mkdir demo_1; cd demo_1

  > wget https://master.dl.sourceforge.net/project/rna-darts/demo/darts_demo_1.tar.gz

  > tar -xvzf darts_demo_1.tar.gz

- Run Darts-flat inference on the read count file `input.read_count.txt`; for the sake of time, let's only run the first 200 events:

  > head -n 201 input.read_count.txt > input.read_count.200.txt

  > mkdir darts_out

  > Darts_BHT -i input.read_count.200.txt -o darts_out/ -k 1 -v

- This will prompt a progress bar for showing the inference progress. Once finished, the output file will be generated in folder `darts_out`; for now, let’s use the pre-computed file you just downloaded, which you can check the first 200 rows are identical:

  > less darts_flat.out.txt

3. Using Darts_DNN

- Build the feature set for our target exon events:

```
> Darts_DNN build_feature -i darts_flat.out.txt \
  -c cisFeature_absmax_normalized.h5 \
  -e kallisto/PC3E/ kallisto/GS689/ \
  -o data.h5
```

- Note: the backslash “\” separates different arguments for readability; in practice it can be omitted.
- This step will generate the output feature set “data.h5”, using the cis-feature stored in “cisFeature_absmax_normalized.h5” and the Kallisto abundance results stored under “kallisto” directory.

- Now run the prediction:

  > Darts_DNN predict -i data.h5 -o pred.txt -m model_param.h5

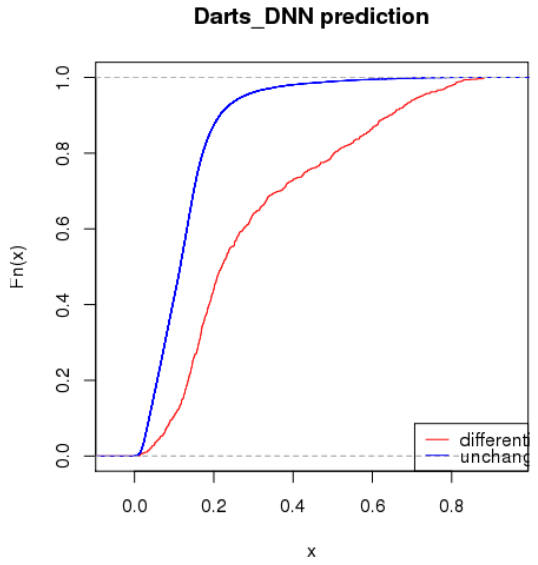
- If configured correctly, you should see:

  2018-03-17 16:43:09,112 - Darts_DNN.predict - INFO - pos=810
  2018-03-17 16:43:09,114 - Darts_DNN.predict - INFO - neg=38963
  2018-03-17 16:43:09,114 - Darts_DNN.predict - INFO - AUROC=0.799907525436
  2018-03-17 16:43:09,114 - Darts_DNN.predict - INFO - AUPR=0.159964942402

4. Visualizing Darts_DNN prediction

- We can do a quick check on the prediction in R. We should see a clear separation between differential splicing events vs. unchanged events:

```
%%R
> data=read.table('pred.txt', header=T)
> plot(ecdf(data$Y_pred[data$Y_true>0.9]), col='red', do.points=F, main='Darts_DNN prediction', ylim=c(0,1))
> lines(ecdf(data$Y_pred[data$Y_true<0.1]), col='blue')
> legend('bottomright', col=c('red','blue'), legend=c('differential', 'unchanged'), lty=1)
```



5. Run Darts_BHT with informative prior

- Since the prediction AUROC for this data is 0.80, this prediction can be utilized as an informative prior in Darts_BHT to boost the biological discovery in lowly expressed genes.

- Here is how to do it. We will use the first 200 events again as an example:

  > Darts_BHT -i input.read_count.200.txt -o darts_out/ -r pred.txt -k 1 -v

- This time, the output file name will be “Sp_out.prior.txt”. You can check the difference by incorporation of informative prior to the Darts-flat generated “Sp_out.txt” previously.