Installing and Running TuniCUT

March 02, 2016

Rational design and whole-genome predictions of sgRNAs for efficient CRISPR/Cas9-mediated mutagenesis in *Ciona*

Shashank Gandhi, Lionel Christiaen, and Alberto Stolfi (2016)

All files available on GitHub: https://github.com/shashank357/TuniCUT
For questions regarding TuniCUT, email: shashank.gandhi@caltech.edu

Please follow this step-by-step tutorial to install TuniCUT on any local machine to run predictions.

Step 1:
Installing Anaconda

Visit Anaconda's official webpage (https://www.continuum.io/downloads) and look for Mac OSX 64bit command line installer. Download the installer file for python 3.5. In Windows 10, command lines in "Anaconda Prompt" will work, but for other versions of Windows you may have to use the built-in Spyder terminal instead.

Open Terminal, type:
cd ~/Downloads/
Press enter. Then type:
bash Anaconda3-2.4.1-MacOSX-x86 64.sh
Press Enter.

Anaconda will start installing. Enter "yes" whenever prompted. Anaconda includes conda, conda-build, Python, and over 100 automatically installed scientific packages and their dependencies. Like Miniconda, over 200 scientific packages can be installed individually with the "conda install" command.

Now close your terminal (Command+Q) and open it again. In order to check whether anaconda was successfully installed or not, type:

conda --version
on the terminal window and press Enter.
This version number will appear if anaconda was successfully installed.

Step 2:
Installing supporting packages on the terminal

If the terminal window is still open, type:
conda update conda

and hit enter. Then type:
conda update --all

and hit enter.
Then run the following installation commands:
conda install biopython
conda install scikit-learn
conda install seaborn

Step 3:
Running TuniCUT

Make a new directory on your Desktop and name it “tunicut". Please make sure that the following files are present in this new directory:

1. tunicut.py
2. TrainingData.txt
3. CompleteData.csv

Open Terminal again. Type:
cd ~/Desktop/tunicut/

and press enter. This will take you to the folder where these files are present.
You can check that all the files are present in this folder by typing:

ls

and pressing enter.

Run TuniCUT using the following command:

python tunicut.py -f YourFastaFile.fa -o YourOutputFileName.txt -t TrainingData.txt -c CompleteData.csv

Here's an example of how the command should look like:

python tunicut.py -f LionelSequence.fa -o LionelPredictions.txt -t TrainingData.txt -c CompleteData.csv

YourFastaFile.fa: Name of your fasta file in fasta format (.fa). In order to run TuniCUT on your target sequences, all nucleotides must be indicated in ACGT or N (uppercase or lowercase). However, targets and PAM sites containing “N” (even if it is the “N” in “NGG”) will not be considered and will be ignored, and sgRNAs targeting these will not be listed in the output file. Other ambiguous nucleotide symbols (e.g. W, R, Y, etc.) will result in an error. It is also critical to provide enough flanking sequence surrounding your target (at least 11 bp surrounding each potential protospacer + PAM sequence), as the prediction takes flanking sequences into account.
YourOutputFilename.txt: This is the text file that will have your results. This .txt file is tab-delimited and can easily be imported in Microsoft Excel if needed. Please ensure that you add “.txt" extension to the file name as it is not added by default. In addition to this .txt file, TuniCUT will also automatically generate .gff3 and .bed files as complementary outputs.
CompleteData.csv: Contains entire dataset needed for TuniCUT to function. Provided with the package.
TrainingData.txt: Contains the training dataset needed for TuniCUT. Provided with the package.

Once you run the command, the output files with your results should be present in the folder. The .txt file should contain all identified sgRNAs (excluding those with stretches of more than 3 thymines in a row) in the sequence, along with predicted scores (given as % mutagenesis) and pre-designed oligonucleotide sequences ready to be ordered and used to construct the sgRNA expression cassette, by OSO-PCR. The .gff3 and .bed files can be used to visualize your results in an interactive manner in the IGV genome visualization tool (https://www.broadinstitute.org/igv/).