Laboratory 10: getting GO

In this laboratory exercise, you will attempt to produce preliminary functional annotations for the open reading frames in a portion of the *Selaginella moellendorffii* genome sequence. Inferring gene function in *S. moellendorffii* is challenging because none of the well–annotated model plant species are closely related, as a result similarity searches often have few, to no, useful model species hits.

Tasks

(1) Retrieve a fragment of the *Selaginella moellendorffii* genome (Banks et al. 2011) by typing `esearch -db nuccore -query 'NW_003314451[Accession]' | efetch -format fasta > NW_003314451.fasta` in the terminal.

(2) Reformat the download for GeneMark by typing `GenBank2fasta.py -a -f NW_003314451.fasta > Selaginella.fasta` in the terminal.

(3) Install GeneMark (Ter-Hovhannisyan et al. 2008):

  (a) Open a web browser (e.g. Firefox).

  (b) Go to http://opal.biology.gatech.edu/GeneMark/license_download.cgi in the browser.

  (c) Select 'GeneMark-ES/ET/EP+ ver 4.71_lic' and 'LINUX 64 kernel 3.10 - 5' and complete the rest of the form.

  (d) Agree to the terms and download the program and the 64-bit version of the key.

  (e) Decompress the program by typing `tar xvzf gmes_linux_64_4.tar.gz` in the terminal.

  (f) Move the program and its files to the scripts folder by typing `mv gmes_linux_64_4/ scripts/` in the terminal.

  (g) Add the program to your $PATH by typing `echo 'export PATH=$PATH:$HOME/scripts/gmes_linux_64_4/' >> .bashrc` in the terminal.

  (h) Install the Perl dependencies for GeneMark by typing `sudo cpan install YAML Hash::Merge Logger::Simple Parallel::ForkManager MCE::Mutex` in the terminal. Type your password when prompted. CPAN will require configuration the first time it is used. Attempt to configure automatically.

  (i) Decompress the downloaded key by typing `gzip -d gm_key_64.gz` in the terminal.

  (j) Rename the key by typing `mv gm_key_64 $HOME/.gm_key` in the terminal.

  (k) Close the terminal window and open a new one to make sure your changes to PATH take effect.

(4) Find the open reading frames in the *Selaginella moellendorffii* genome fragment by typing `gmes_petap.pl --ES --cores $(nproc) --sequence Selaginella.fasta` in the terminal. After a bit, this will produce a file called 'genemark.gtf'.

(5) Extract the open reading frames by typing `get_sequence_from_GTF.pl genemark.gtf Selaginella.fasta` in the terminal. This will produce files called 'nuc_seq.fna' and 'prot_seq.faa'. Answer question (1).

(6) Install DIAMOND (Buchfink et al. 2015) by typing `sudo apt install diamond-aligner` in the terminal. Type your password when prompted and agree to the install. You will be asked to choose which users DIAMOND is configured for—be sure your user is among those selected.

(7) Download and format *Arabidopsis thaliana* reference data:

   (a) Download all peptide sequences by typing `wget https://www.arabidopsis.org/download_files/Genes/TAIR10_genome_release/TAIR10_blastsets/TAIR10_pep_20101214_updated` in the terminal.

   (b) Create a DIAMOND database by typing `diamond-aligner makedb --in TAIR10_pep_20101214_updated -d Arabidopsis` in the terminal.

   (c) Download GO data by typing `wget https://www.arabidopsis.org/download_files/GO_and_PO_Annotations/Gene_Ontology_Annotations/ATH_GO_GOSLIM.txt.gz` in the terminal.

   (d) Decompress the file and extract only the functional GO annotations that are supported by experimental evidence by typing `gzip -cdk ATH_GO_GOSLIM.txt.gz | perl -F'\t' -lane '{if(($F[7] eq "F")&&($F[9]=~m/EXP|HDA|HEP|HGI|HMP|HTP|IDA|IEP|IGI|IMP|IPI/)) {print(join("\t",@F[0,4,5,8]))}}' | sort -u > Arabidopsis.go` in the terminal. Answer question (2).

(8) Query the *Arabidopsis thaliana* database with the inferred proteins from *Selaginella moellendorffii* by typing `diamond-aligner blastp -p $(nproc) -d Arabidopsis --more-sensitive -f 6 qseqid sseqid bitscore evalue length pident -q prot_seq.faa -o matches.tsv` in the terminal. Answer question (3).

(9) Install DIAMOND2GO.py:

   (a) Move to your scripts directory by typing `cd scrips` in the terminal.

   (b) Download the script by typing `wget https://raw.githubusercontent.com/dpl10/phytoinformatics2023/master/DIAMOND2GO.py` in the terminal.

   (c) Make the script executable by typing `chmod 0755 DIAMOND2GO.py` in the terminal.

   (d) Return to your home directory by typing `cd` in the terminal.

(10) Extract the *Arabidopsis thaliana* GO annotations from the *Selaginella moellendorffii*/*Arabidopsis thaliana* DIAMOND hits by typing `DIAMOND2GO.py -d matches.tsv -g Arabidopsis.go > diamond.tsv` in the terminal. Answer question (4).

(11) Install DeepGoPlus (Kulmanov & Hoehndorf 2020).

   (a) Download the DeepGoPlus code by typing `wget https://github.com/bio-ontology-research-group/deepgoplus/archive/refs/tags/v1.0.1.tar.gz` in the terminal.

   (b) Extract the compressed tar ball by typing `tar xvzf v1.0.1.tar.gz` in the terminal.

   (c) Enter the code directory by typing `cd deepgoplus-1.0.1/` in the terminal.

   (d) Download the exact version of DIAMOND used by DeepGoPlus by typing `wget http://github.com/bbuchfink/diamond/releases/download/v2.0.2/diamond-linux64.tar.gz` in the terminal. Answer question (5).

   (e) Type `tar xvzf diamond-linux64.tar.gz` in the terminal to decompress the tar ball.

   (f) To build a Docker instance compatible with DeepGoPlus, create a Docker build file:

      (1) To start the Docker file, type `echo 'FROM tensorflow/tensorflow:2.3.1' > Dockerfile` in the terminal.

(2) To add to the build file, type `echo 'ADD diamond /usr/bin/' >> Dockerfile` in the terminal.
(3) Add more to the build file by typing `echo 'RUN python3 -m pip install --upgrade pip' >> Dockerfile` in the terminal.
(4) And add more by typing `echo 'RUN python3 -m pip install --upgrade setuptools' >> Dockerfile` in the terminal.
(5) And add more by typing `echo 'RUN python3 -m pip install --upgrade Click==7.1.2' >> Dockerfile` in the terminal.
(6) Finally, type `echo 'RUN python3 -m pip install --upgrade pandas==1.1.2' >> Dockerfile` in the terminal to complete the build file.

(g) To build the Docker instance type `docker build -t deepgoplus .` in the terminal. This command may take a bit to run.
(h) Download the trained DeepGoPlus model and associated data by typing `wget http://deepgoplus.bio2vec.net/data/data.tar.gz` in the terminal.
(i) Decompress the DeepGoPlus data by typing `tar xvzf data.tar.gz` in the terminal.

(12) Make a local copy of the protein sequences to be annotated by typing `cp ../prot_seq.faa .` in the terminal.

(13) Start the DeepGoPlus compatible Docker instance by typing `docker run -u $(id -u):$(id -g) --rm -it -v "$PWD:/tmp" -w /tmp deepgoplus` in the terminal. Answer question (6).

(14) Conduct a DIAMOND search for DeepGoPlus input by typing `diamond blastp -p $(nproc) -d data/train_data.dmnd --more-sensitive -q prot_seq.faa --outfmt 6 qseqid sseqid bitscore | gzip > prot_seq-diamond.tsv.gz` in the terminal.

(15) Compress the input sequences for DeepGoPlus input by typing `gzip prot_seq.faa` in the terminal.

(16) Run the DeepGoPlus model prediction by typing `python predict.py -if prot_seq.faa.gz -of deepgoplus.tsv.gz -df prot_seq-diamond.tsv.gz` in the terminal. Answer question (7).

(17) Type `exit` to stop the DeepGoPlus compatible Docker instance.

Questions (https://forms.gle/WKhzVrNoo9JkUu5u6)

(1) For task (5), how many sequences were identified?

(2) For task (7)(d), what does each of the steps in the command do?

(3) For task (8):

(a) What does each of the diamond-aligner options do?
(b) How many *Selaginella moellendorffii* sequences matched an *Arabidopsis thaliana* sequence?
(c) How could you change the DIAMOND command to retrieve fewer matches?

(4) For task (10):

(a) How many putative *Selaginella moellendorffii* proteins had *Arabidopsis thaliana* GO terms associated with them?
(b) How confident are you in the GO functional annotations assigned? Why?
(c) Would adding annotations form additional model species improve the quality of the GO functional annotations?

(5) For task (11)(d):

(a) Which version of DIAMOND is being installed?
(b) Which version did you install in task (6)?
(c) Do the two versions take the same options?

(6) For task (13):

(a) What do each of the command options do?
(b) Why did you have to make a copy of 'prot_seq.faa' rather than using the existing one?

(7) For task (16):

(a) Compare and contrast the DeepGoPlus annotations to the DIAMOND annotations.
(b) If you were writing a scientific manuscript, which would you present? Why?

Literature cited

Banks, J. A., T. Nishiyama, M. Hasebe, J. L. Bowman, M. Gribskov, C. de Pamphilis, V. A. Albert, N. Aono, T. Aoyama, B. A. Ambrose, N. W. Ashton, M. J. Axtell, E. Barker, M. S. Barker, J. L. Bennetzen, N. D. Bonawitz, C. Chapple, C. Cheng, L. G. G. Correa, M. Dacre, J. DeBarry, I. Dreyer, M. Elias, E. M. Engstrom, M. Estelle, L. Feng, C. Finet, S. K. Floyd, W. B. Frommer, T. Fujita, L. Gramzow, M. Gutensohn, J. Harholt, M. Hattori, A. Heyl, T. Hirai, Y. Hiwatashi, M. Ishikawa, M. Iwata, K. G. Karol, B. Koehler, U. Kolukisaoglu, M. Kubo, T. Kurata, S. Lalonde, K. Li, Y. Li, A. Litt, E. Lyons, G. Manning, T. Maruyama, T. P. Michael, K. Mikami, S. Miyazaki, S. Morinaga, T. Murata, B. Mueller-Roeber, D. R. Nelson, M. Obara, Y. Oguri, R. G. Olmstead, N. Onodera, B. L. Petersen, B. Pils, M. Prigge, S. A. Rensing, D. M. Riaño-Pachón, A. W. Roberts, Y. Sato, H. V. Scheller, B. Schulz, C. Schulz, E. V. Shakirov, N. Shibagaki, N. Shinohara, D. E. Shippen, I. Sørensen, R. Sotooka, N. Sugimoto, M. Sugita, N. Sumikawa, M. Tanurdzic, G. Theißen, P. Ulvskov, S. Wakazuki, J.-K. Weng, W. W. G. T. Willats, D. Wipf, P. G. Wolf, L. Yang, A. D. Zimmer, Q. Zhu, T. Mitros, U. Hellsten, D. Loqué, R. Otillar, A. Salamov, J. Schmutz, H. Shapiro, E. Lindquist, S. Lucas, D. Rokhsar & I. V. Grigoriev. 2011. The *Selaginella* genome identifies genetic changes associated with the evolution of vascular plants. Science 332: 960–963.

Buchfink, B., C. Xie & D. H. Huson. 2015. Fast and sensitive protein alignment using DIAMOND. Nature Methods 12: 59–60.

Kulmanov, M. & R. Hoehndorf. 2020. DeepGOPlus: improved protein function prediction from sequence. Bioinformatics 36: 422–429.

Ter-Hovhannisyan, V., A. Lomsadze, Y. O. Chernoff & M. Borodovsky. 2008. Gene prediction in novel fungal genomes using an *ab initio* algorithm with unsupervised training. Genome Research 18: 1979–1990.

Due at the start of class April 4.