Project 4. Tardigrades: from genestealers to space marines

New bioinformatics skills covered: gene prediction, functional annotation, protein localization

Tardigrades (also known as water bears, pudgy wudgies, or moss piglets) are microscopic animals capable of withstanding some of the most severe environmental conditions. These water-dwelling, eight-legged creatures can be found throughout the world, from the Himalayas (above 20,000 ft), to the deep sea (below 13,000 ft). Classified as "extremophiles", they can survive freezing (up to 1°K), total dehydration, pressure (more than 1,200 atmospheres) and radiation (1,000 times more radiation than other animals).

Tardigrades are the first known animal to survive in space. In September 2007, dehydrated tardigrades were taken into low Earth orbit, and exposed to the hard vacuum of outer space and solar UV radiation (the project was called "Tardigrades In Space", or TARDIS).

For a long time researchers tried to understand how these creatures can be so stress-tolerant. They tried to explain radiation withstand with the fact that they are able to enter the dehydrated state (and survive in up to five years), and this state provides fewer reactants for ionizing radiation. But later it was shown that when hydrated, they still remain highly resistant to shortwave UV radiation in comparison to other animals.

It means that they can efficiently repair damage to their DNA. But how?! It was a mystery until 2016, when we were able to sequence the genome of *Ramazzottius varieornatus*, one of the most stress-tolerant species of Tardigrades. In this project we will try to analyze this genome and understand this secret.

1. Obtaining data. Genome sequence.

For this project we will be using a sequence of the *Ramazzottius varieornatus*, the YOKOZUNA-1 strain (sequenced in the University of Tokyo and named after the highest rank in professional sumo).

We will not assemble the genome itself because the raw data are too large (three Illumina libraries, about 10 gigabytes of reads, and additional Sanger sequencing data), and the assembly process is not so different for prokaryotes and eukaryotes. Eukaryotic-focused assemblers (like SOAPdenovo, Platanus or DISCOVAR) are usually tailored for multithreading and take into account possible heterozygosity, but generally they build on the same de Bruijn graph approach.

You can download assembled genome here:
ftp.ncbi.nlm.nih.gov/genomes/all/GCA/001/949/185/GCA_001949185.1_Rvar_4.0/GCA_001949185.1_Rvar_4.0_genomic.fna.gz

NOTE: First steps in subsequent analysis - repeat masking and gene prediction - are computationally-intense, so I made them both strictly optional - you may find it in the very end of the document.

2. Functional annotation.

We worked hard to obtain regions of interest in the genome of tardigrades, corresponding to genes and proteins. But it's still just a string of letters and now we will try to make functional annotations. That is, we're looking for homologous proteins and conserved domains to assign potential functions to the genes and proteins we find.
You can download AUGUSTUS results here.

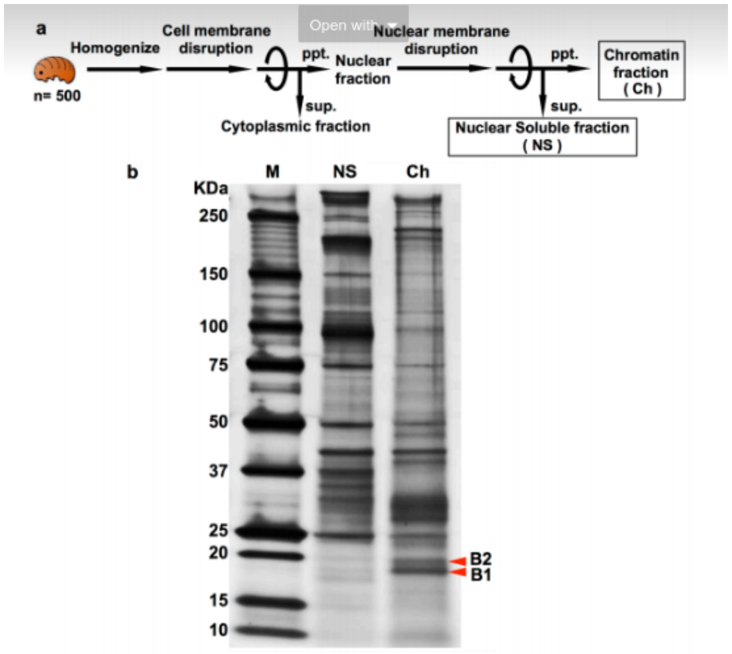
First, we need to extract protein sequences (fasta) from the prediction output (usually GFF/GTF). If you are using AUGUSTUS results, you can do it with the getAnnoFasta.pl script, but there are plenty of other options.

Count the number of obtained proteins (e.g. *grep* all the strings with ">" symbol and count lines with *wc -l*). It should be a pretty large number, common for multicellular organisms (usually 15-20k). How can we narrow it down to a list of potential candidates that we can then verify by experimentation?

3. Physical localization

Considering that DNA is a major target of radiation damage, we may hypothesize that tardigrades might have unique proteins associated with their DNA to protect and/or effectively repair it. To explore this possibility, we will combine genomic and proteomic data.

Using centrifugation and various techniques that disrupt membranes, our colleagues were able to extract chromatin fraction (that is, just the DNA and the proteins associated with it). They then analyzed the extracted proteins using tandem mass spectrometry (technology that allows you to obtain short protein fragments called peptides). This gave them (and you!) a list of peptides that were associated with the DNA (you can download that list here).
Each of the peptides is a piece of one of the proteins that was bound to our DNA (or was pulled out as a false positive), so now our goal is to figure out which proteins from the *R. varieornatus* genome these peptides correspond to, and find the relevant ones.



You can do this in any of the following ways:

1) Ctrl+F (not recommended for obvious reasons)
2) Write a short script in the language of your choice. Note that the sequence in a FASTA file can be written in multiple lines, so you need to take into account all nucleotides (or amino acid residues) between two headers “>”. Specific libraries with implemented parsers, like BioPython, BioPerl or BioJava, can be handy.
3) **(!recommended!)** Do a local alignment-based search: create a local database from your protein fasta file and look it up using your peptide sequence file as a query.
   There are few options:
   a) Classic BLAST+. I hope you already have it installed at this point, but if not I suggest you to install it via conda (e.g. `mamba install -c bioconda blast`)
      First, you need to create a database:
      ```
      makeblastdb -in <reference.fa> -dbtype prot  -out <database_name>
      ```
      Then you can perform the search:
      ```
      blastp -db <database_name> -query <query_proteins.fa> -outfmt 6
      -out <outfile_name>
      ```

Outfmt parameter defines the output format, 6 means tabular format without headers, which is easy to parse. You can play with different types of output, and also define set of desired fields in the tabular format - e.g. -outfmt "6 qseqid sseqid evalue qcovs pident stitle". See the fields description, for example, here.

Then you will need to extract proteins of interest from the initial file. Again, there are a lot of ways to do it, including parsing, but for a quick solution I recommend `samtools faidx` utility -

see the discussion on Biostars. Also, you may consider `seqtk subseq` (seqtk is actually a pretty useful toolkit, developed by Heng Li, great and terrible).

b) Diamond, or 1000x accelerated BLAST. It can be installed as easy as `conda install -c bioconda diamond`. It works with proteins only, and can be a good replacement for blastp and blastx. Its command syntax is very similar to blast:

```
diamond makedb --in <reference.fa>  --db <database_name>
diamond blastp -d <database_name.dmnd> -q <query_proteins.fa>  -f 6 -o
<outfile_name> --very-sensitive
```

(again, you can tune your output format: `-f 6 qseqid sseqid evalue qcovs pident stitle` or something else).

4) Any other way of your choice (for example, mmseqs2 or minimap2). Feel free to share your method with your classmates!

4. Localization prediction

There are still too many proteins to verify each of them experimentally. Now, we will try to predict where these proteins are found in the cell based on their sequences.

4a. WoLF PSORT (https://wolfpsort.hgc.jp/) predicts the subcellular localization of proteins. The signal peptides are short sequences found in proteins that are exported outside of the cytoplasmic membrane. Most secretory proteins are synthesized from a precursor with an N-terminal extension of ~20 amino acid residues called a signal peptide. A signal peptide contains a single transmembrane helix. These signal peptides are removed during translocation of the secretory proteins across the membrane via a special enzyme, called signal peptidase (usually enzymes that end in "ase" tend to break up the thing that is the first part of the wo{rd). PSORT predicts cellular localization of proteins based on the presence of a signal peptide on their N-terminus.

Submit your set of proteins to the WoLF PSORT server, making sure to choose the right organism type. Save the results in your lab journal.

4b. TargetP 1.1 Server

TargetP 1.1 (https://services.healthtech.dtu.dk/service.php?TargetP-2.0) also predicts the subcellular localization of eukaryotic proteins. The location assignment is based on the predicted presence of any of the N-terminal presequences: chloroplast transit peptide (cTP), mitochondrial targeting peptide (mTP) or secretory pathway signal peptide (SP).

Submit your set of genes to the TargetP server, making sure to choose the right organism type. Save the results in your lab journal.

5. BLAST search

BLAST protein sequences against the “UniProtKB/Swiss-Prot” database. Then, download and explore the results. For each best hit, you can save in your journal the Accession Number, E-value, % Ident, % Query coverage, and annotation.

6. Pfam prediction

We can predict the function of the proteins even if we can not find orthologous sequences in databases with Blast. We will use HMMER (web-version, https://www.ebi.ac.uk/Tools/hmmer/) to search our protein sequences against a collection of profile-HMMs for different protein domains and motifs.
Select “Search” and then select the suitable tool (in our case it is hmmscan), select the Pfam database, and submit your proteins.

7. Integrate your various pieces of evidence

Create a table where for each protein, you provide the following information (if any is available):
a) best blast hit (annotation and e-value)
b) predicted Pfam domains
c) probable localization(s) according to WoLF PSORT
d) localization according to TargetP

What entries can you recommend for experimental verification as novel tardigrade-specific DNA-binding proteins? For all database hits pay special attention to e-value, and describe reliability of your findings.

For your lab report

For the abstract, as usual, briefly introduce the topic, briefly discuss how you tackle the problem, and briefly mention the key results and their impact.

For the introduction, describe tardigrades and why we are interested in them. Also provide some background on tardigrade genome studies. Briefly introduce the computational methods we're using: gene prediction, sequence homology.

In your methods section, you should mention all the tools and approaches you've used in the search of DNA-repair related proteins. Be sure to describe the parameters you used if they were not the default options (or say "We ran __ with default parameters").

For the discussion section, explore your candidate genes and provide assumptions about their possible role in DNA repair. Provide ideas of other approaches to verify your findings. Read a related paper (or just abstract and part "Identification of a tardigrade-unique DNA-associated protein.") and compare your findings with relevant part of the results from the paper. You can suggest further experiments to explore other genetic mechanisms of tardigrade stress resistance.

***Optional Extra-Credit Challenge Question**

For 1 bonus point on this lab report each.

1. ***Repeat masking***

The first step in the computation phase of genome annotation is repeat identification. Eukaryotic genomes can be very repeat rich (up to 50% of the human genome is thought to consist of repeats). Repeats can be inserted within other repeats, and often only fragments within fragments are present.

The term 'masking' simply means transforming every nucleotide identified as a repeat to an 'N' ("hard masking") or to a lower case a, t, g or c ("soft masking"). Left unmasked, repeats can seed millions of spurious BLAST alignments, producing false evidence for gene annotations. Worse still, many transposon open reading frames (ORFs) look like true host genes to gene predictors, causing portions of transposon ORFs to be added as additional exons to gene predictions, completely corrupting the final gene annotations. That is why good repeat masking is thus crucial for the accurate annotation of protein-coding genes.

Usually we mask repeats in the sequenced genome using RepeatMasker (http://www.repeatmasker.org/). There is also web-version, but it is not suitable for the genomes that big. Note that RepeatMasker is based on reference repeat database, so you should select repeat libraries from closest reference (e.g. C.elegans).
But for novel organisms, like a tardigrade, it's better to use RepeatModeller, which can predict novel repeat families. Best way to install it is via Anaconda.
https://anaconda.org/bioconda/repeatmodeler

2. Gene prediction

There are lots of different ways to predict coding regions in the genome. The first gene predictors (from back in the 1990s) provided a fast and easy means to identify genes in *assembled DNA sequences for those times. They are often referred to as ab initio gene predictors because they use mathematical models rather than external evidence (such as EST and protein alignments) to identify genes and to determine their intron–exon structures. You can try GeneMark-ES that works completely ab initio, without curated training sets, performing several iterations of a Viterbi-type algorithm, finding the most likely HMM trajectory given the DNA sequence.*

Another method of the gene prediction is based on evidence hints (e.g. transcriptome). There are multiple of these programs, one of the most popular is called Augustus (http://bioinf.uni-greifswald.de/augustus/). To predict genes, it requires models, i.e., files with probabilities of specific 2-, 3-, 4- and 5-mers in the introns, exons, upstream and downstream untranslated regions and on the boundaries.

AUGUSTUS has currently been trained on species-specific training sets to predict genes in the limited list of species. The good news is that, for closely-related species, we usually only need one training set. For example, the human version is good for all mammals.

For tardigrades you may want to pick someone closer. In the lower part of the title page, there is a list of all currently supported provided models. It's pretty hard to pick a desired model based on just latin names, so you can find closest relative using the phyloT service (http://phylot.biobyte.de/) - it can draw a phylogenetic tree based on the current taxonomy status of the species in the NCBI Taxonomy database.

You may try it locally, or use the web version (detailed tutorials are provided!).

3. Model training

Our species of interest is pretty far from any pre-calculated models, so we can try to create a model specifically for our species given additional evidence, such as RNA-seq data. A full-length cDNA dataset was created for R. varieornatus using Sanger sequencing. You may download it from the NCBI EST database: open the R.varieornatus page in NCBI Taxonomy, select "Nucleotide EST", download it in the GenBank format, and train it according to the author's tutorial (somewhat unclear) or the web version (time-consuming). Use the obtained model to reannotate the genome and compare with your previous results.