Introduction to strings and sequence alignments

1 Sequences of biomolecules

Biology uses two main *languages* to store information in biomolecules, these are the languages of *nucleic acids* such as DNA and RNA and *proteins.* Thankfully for English speakers, the languages have fewer *letters* than English does, so we can map from biomolecules into English. We're going to start by examining DNA sequences which are adenine (`A`), thymine (`T`), guanine (`G`), and cytosine (`C`).

1.1 Is there a match?

In MATLAB, there are a few functions which can be helpful for dealing with strings. One of them, `strfind` takes two input strings and finds the location of one within the other. Before we move to DNA and biomolecules, let's start with an classical document in American history, the Declaration of Independence.

- Download file `independence.txt` which has the text of the Declaration as a single line.
- Use some method to get this text into MATLAB and search for the string `security`.
- How many uses of the word `the` are there?
- What about the word `Tyrant`
- (Bonus) What is the most abundant word in the Declaration (ignoring case)?

1.2 Find a specific sequence

Now we're going to move from simple string searching on a *short* text to something *long.* Bacterial genomes are typically a few mega basepairs (Mbp) in length, roughly 1000X smaller than the human genome. The circular genome codes for all of the proteins necessary for the cells to grow and reproduce. For this section, we're going to focus on the bacterial pathogen *Vibrio cholerae*, the causative agent of the human diarrheal disease cholera. In this section, we're going to search for specific sequence in its genome.

- Download the entire genome of *V. cholerae* from `V_cholerae.fna`. This file format is called a Fasta file and MATLAB's bioinformatics toolbox has a fast reader for it called `fastaread`. Once we have it loaded, we're going to explore the ideas of unique sequences as well as execution time.
- Using `fastaread`, import the genome into MATLAB.
- Develop a piece of code that will allow you to search for the specific sequence `ACATGAT`. *V. cholerae* has two circular chromsomes, so make sure you search for the sequence on each of them.
- How many times is `ACATGAT` present in the genome?
- Choose ten other 7-letter sequences and determine how many instances each has.
- Using the `subplot` function, make four plots stacked on top of each other. From top to bottom, plot a histogram of number of occurrences for random 5-letter, 7-letter, 10-letter and 15-letter sequences.

- Using `tic` and `toc`, keep track of the time it takes to search for sequences from length 3 to 60. Note: it may take a very long time to search for all 60^4 combinations, so you should make a good choice about how many to actually test. Keep track of the time per search and the number hits, you'll be using them to make a plot in the next step.
- Make a plot using two different colors of y-axes that shows the number of hits found for the measurements in the previous step as well as the time required to make the search. Do *log* or *linear* displays show the trend more clearly?
- (Bonus) Think through how you would adapt your algorithm if you wanted to match substrings that were only off by one mismatch. We'll see in a later section that fancy algorithms include penalties for gaps and mismatches.

1.3 Multiple sequence alignment: helicases

In the previous section, we looked for sequences that were exact matches. It turns out that the while the *in*exact match is harder to find, it is very important in biology. For example, many proteins vary in sequence only slightly, but still perform similar functions. While it would be possible to write your own algorithms to search for various types of mismatches, the bioinformatics community has already come up with a variety of methods for doing this matching. Here, we'll explore multiple sequence alignment using MATLAB's implementation `multialign`. In this section we'll look at this approach for finding the conserved active site of a specific class of helicases, enzymes that unwind DNA.

- Use the included script `retrieveFastaSeqs_helicases.m` to retrieve a collection of protein sequences from the **N**ational **C**enter for **B**io**I**nformatics. These sequences have been hand picked from a collection of enzymes all part of the same superfamily as defined by the database `PFam`.
- After collecting these sequences, run a multiple-sequence align using a built-in MATLAB tool whose name you can find in the bioinformatics toolbox help.
- Display the multiple-sequence alignment and browse around to look for regions that are conserved, that is, there are either the same or similar amino acids in the same order. Do you see any residues that are completely conserved? Can you see why this class is known as the `DEAD-box` family of helicases?

Note: This list is not all the members of the superfamily, we have cherry picked some examples to accentuate our particular point. To see the whole family, you can browse to `Pfam PF00270`.

These types of approaches can also be extended to compare enzymes between different types of organisms, for example, "Do proteases from bacteria that live at different temperatures share sequence similarity?" [1]

1.4 Further reading: VDJ recombination

During the process of creating antibodies, immune cells go through a process called VDJ recombination which generates unique protein sequences by selecting one **V** segment, one **D** segment, and one **J** segment. These antibodies are then used to recognize self from non-self protein molecules. By sequencing the antibodies from blood samples, researchers are studying how diverse an antibody repertoire can be created. There is also some chance of being able to detect if a person has been exposed to a certain pathogen based on the presence or absence of a certain antibody. The aspect of sequence matching that makes this difficult is that there is not a one-to-one match. That is, there are partial matches to each of the VDJ segments. If you are interested in reading more about the biology and bioinformatics behind this, see [2, 3].

1.5 Further reading: RNAseq

As you have seen in other parts of this "text search" section, MATLAB is not necessarily the state of the art system for analyzing the large amount of data that comes from sequencing experiments. For example, a common contemporary type of experiment is an RNAseq experiment. During such an experiment, different biological samples are compared based on the sequence and/or abundance of the mRNAs in the cells.

See, for example, the dataset from `https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE114065` “Transcriptomes and methylomes from naïve CD4+ T-cells from infants and children with and without food allergy”. In this study, the researchers generated more than 300 GB of sequencing reads which all needed to be mapped back to the human genome. In such situations, there are web based, distributed computing platforms such as `Galaxy` that run high-end tools like BWA-MEM [4] or Bowtie2 [5] behind the scenes. This separation of computing and user interfaces should allow easier access to end users who are not as familiar with algorithms or algorithm development.

References

[1] S. Tilak Raj, Nikhil Sharma and T. C. Bhalla, “Bacterial serine proteases: Computational and statistical approach to understand temperature adaptability,” *Journal of Proteomics & Bioinformatics*, vol. 10, no. 12, pp. 329–334, 2017.

[2] B. S. Wendel, C. He, M. Qu, D. Wu, S. M. Hernandez, K.-Y. Ma, E. W. Liu, J. Xiao, P. D. Crompton, S. K. Pierce, P. Ren, K. Chen, and N. Jiang, “Accurate immune repertoire sequencing reveals malaria infection driven antibody lineage diversification in young children,” *Nature Communications*, vol. 8, p. 531, Sept. 2017.

[3] Z. Sethna, Y. Elhanati, C. R. Dudgeon, C. G. Callan, A. J. Levine, T. Mora, and A. M. Walczak, “Insights into immune system development and function from mouse T-cell repertoires,” *PNAS*, vol. 114, pp. 2253–2258, Feb. 2017.

[4] H. Li, “Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM,” *ArXiv e-prints*, Mar. 2013.

[5] B. Langmead and S. L. Salzberg, “Fast gapped-read alignment with Bowtie 2,” *Nature Methods*, vol. 9, pp. 357–359, Apr. 2012.