RBFOX2 RNA-binding specificity using RNA Bind-N-Seq (RBNS)

Fundamentals Assignment #4 – Covid-19 Computational Biology Workshop – May 14, 2020

Background

RNA-binding proteins (RBPs) regulate myriad aspects of RNA processing, including alternative splicing, RNA localization, and RNA stability. RBPs often recognize their target RNAs by short motif sequences, and the specificity of this RBP-RNA interaction is a biophysical/thermodynamic property of the RBP. Identification of the canonical motif(s) for a given RBP has enabled thorough study of the functions of specific binding events. However, RBPs often bind non-canonical motifs as well, albeit with weaker affinity, and these binding events contribute substantially to a nuanced RNA regulatory network across the transcriptome. Until recently, knowledge of these non-canonical binding events, including their binding affinities and dynamic behaviors, was very limited.

RNA Bind-N-Seq (RBNS) is an experimental technique to profile the binding specificities of a particular RBP *in vitro* using a library of random RNA sequences (Lambert et al. 2014). To perform RBNS, a 40-mer degenerate DNA sequence is transcribed by T7 polymerase, and the random RNA is incubated with the RBP of interest. The RBP-RNA complexes are precipitated with streptavidin, and the RNA that bound to the RBP is sequenced single-end on an Illumina HiSeq. As a control for quantification of enrichment, the input library is also sequenced using the same library preparation protocol. Enrichment of a motif in the RBNS library relative to the input indicates the binding affinity of the RBP for that motif. Using this approach, the authors characterized the binding specificies and thermodynamics of RBFOX2, CELF1, and MBNL1, all of which are important developmental alternative splicing factors.

Assignment

In this assignment, we will take a look at the RBNS data from the referenced publication, and we will use it to confirm the canonical binding motif of RBFOX2, which is `UGCAUG`. To accomplish this, we will be using two external libraries, `biopython` and `matplotlib`.

The single-end sequencing data from this paper is publically available on the Sequence Read Archive (SRA) at the ID `SRP041098`. For convenience, we have placed copies of the RBNS sequencing files on HiPerGator in the directory `/ufrc/ewang/cpkelley94/covidcompbio/RBNS`. You may copy these files into your own directory if you wish. There are two files in this directory:

1. **`RBNS_RBFOX2_365nM.fastq`**: the RBNS sequencing data from incubation of RBFOX2 with random RNA at 365 nM.
2. **`RBNS_RBFOX2_input.fastq`**: the sequencing data from the input (control) library.

We recommend that you complete this assignment on HiPerGator, as the sequencing files may be too large to store on your local machine, and analysis may require significant RAM usage (depending on how your code is structured).

Goals

1. Define a function called `count_motifs` that takes in two arguments: (1) a path to a sequencing FASTQ file (`fastq_path`) and (2) a list of motifs to search for (`motif_list`). The function should open the sequencing file, loop through the reads in the file, and count the number of reads in the file that contain each motif. Return a dictionary called `counts` whose keys are the motifs in `motif_list` and whose values are the number of reads containing the motif. Also return the total number of reads in the file.

2. Make a list that contains four motifs: (a) the canonical RBFOX2 motif `UGCAUG` and (b) three other motifs with the same base composition (`GAUUGC` , `UUGGCA` , `CUGAGU`). For each of the two sequencing FASTQ files, use your function `count_motifs` to get the total counts for each motif and the total number of reads in each file.

3. For each motif, calculate its enrichment in the RBNS data as

$$\text{enrichment} = \frac{\left(\text{motif counts in RBNS} \middle/ \text{RBNS total read count}\right)}{\left(\text{motif counts in input} \middle/ \text{input total read count}\right)}$$

4. Make a bar chart to show the enrichment of each motif in the RBFOX2 RBNS data. How does the enrichment of the canonical motif compare to the three randomized motifs?

Tips

- The first major step in writing `count_motifs` is to open up the sequencing file and loop through the reads. `biopython` provides a module called `SeqIO` for this purpose. In order to use it, you'll need to import it:

  ```
  from Bio import SeqIO
  ```

  Use the `parse()` function to create an iterator object from the FASTQ file:

  ```
  for i, record in enumerate(SeqIO.parse(fastq_path, 'fastq')):
  ```

  This will iterate over each sequencing read in the FASTQ file, handing you a `SeqRecord` object for each iteration. This object contains a lot of information about the sequencing read, but we only care about the sequence itself, which can be accessed as a `Seq` object using `record.seq` . Print out the first 10 read sequences to get an idea of the type of data we are working with.

- Illumina sequencers provide DNA sequences, not RNA. Thus, you will either need to convert the motifs to DNA sequences or the reads to RNA. To convert the `Seq` object to RNA, we can use the method `Seq.transcribe()` .

- Sometimes it is easier to ditch the BioPython object altogether and use the standard string type instead. You can do this by casting the `Seq` object to a string using `str()` .

- Iterating over millions of reads may take a substantial chunk of time. Our implementation of this script completed in ~15 minutes. Use `print()` to track the progress of your script periodically. While you are iterating through the sequencing files, you can print the progress every `n` reads using the modulo operator (`%`):

  ```
  if i % n == 0:
      print('Progress: read ' + str(i) + '...')
  ```

- You can make a bar graph (and many other kinds of plots) using the `pyplot` module from `matplotlib` . Import it with the following:

  ```
  import matplotlib
  matplotlib.use('Agg')

  from matplotlib import pyplot as plt
  ```

  The first two lines are required when working on HiPerGator, as `matplotlib` normally relies on a backend to display graphs on the screen, but HiPerGator cannot display plots graphically over the terminal. Instead, you can save your plots to image files for you to download and open on your local machine.

Make a bar chart using `plt.bar()`. You can find the documentation for this function and some examples here. You can change the axis labels using `plt.xlabel()` and `plt.ylabel()`, as well as the x-axis tick locations and labels using `plt.xticks()`. Save your figure to a PNG file using `plt.savefig()`.

Next level

1. **Trimming and filtering.** Sequencing datasets often contain a percentage of junk reads as a result of contamination during library preparation, incorrect index assignment after sequencing, and carryover of spike-in DNA. Additionally, reads often contain unwanted sequence data flanking the region of interest, such as adapter sequences required for library preparation. *Trimming* (cutting unwanted sequences off of reads) and *filtering* (excluding reads from analysis that do not fit certain criteria) are common quality control (QC) techniques for improving the robustness of sequencing analysis.

   In this experiment, the 40 nt randomized region is flanked by a 5’ adapter sequence `CGACGAUC` and a 3’ adapter sequence of `UGGAAUUC`. We found that the 3’ adapter sequence remains present in most of the reads. For each read, find the position of `UGGAAUUC` in the sequence using the string method `find()`, and only consider sequence upstream of that adapter handle when searching for motifs.

   What is returned by `find()` when the 3’ adapter is not found in the read? Handle this case by skipping these reads using `continue`, and count the occurrences of skipped reads.

2. **Unbiased search.** Rather than searching for a small set of motifs, we can perform an unbiased search for enriched motifs, allowing us to profile the binding preferences of RBFOX2. Write a function that generates all possible RNA motifs 6 nt in length (4096 motifs total), and find the enrichment of each motif in the RBNS experiment. Can you identify non-canonical RBFOX2 motifs with substantial enrichment?

   What are the flaws of this approach, and how would you fix them?