Introduction to computational biology

Assignment 2 report

Piotr Rutkowski, University of Warsaw 2020

Part 1: *Identify closest related protein-coding sequences in E.coli genome for each of the input protein sequences given in fasta file (protein_fragments.fa).*

Firstly, I created a BLAST database using given E.coli DNA genome sequences from genes_e_coli_new.fa. To make this data compatible with protein sequences, I translated DNA sequences into amino acid sequences and saved results as db.fa. Subsequently, I used Bio.Blast.Applications.NcbimakeblastdbCommandLine module to create a BLAST database. Having created the BLAST database, I performed local BLAST search against this database using Bio.Blast.Applications.NcbiblastpCommandLine module. The result was a blastp.xml file that I parsed and got a set of E.coli sequences that were identified as the closest related to all 98 protein sequences from protein_fragments.fa. For some proteins, there were multiple alignments. In such case, I chose the first one (which is also the best one).
BLAST output included e-values for all alignments. In each case, they were very small (the smaller, the better). Considering only best alignments, the largest e-value was 7.68558e-27. However, the majority of e-values were smaller than 1.0e-80.
Below you can see 2 of 98 BLAST alignment results and the code that printed them.

```
NEW RECORD 79
****Alignment****
sequence: gnl|BL_ORD_ID|223 prfC coding sequence
length: 529
e value: 6.72403e-99
DTYRTLTKVDCCLMVIDAAKGVEDRTRKLMEVTRLRVTPILTFMNKLCRD...
DTYRTLT VDCCLMVIDAAKGVEDRTRKLMEVTRLR TPILTFMNKL RD...
DTYRTLTAVDCCLMVIDAAKGVEDRTRKLMEVTRLRDTPILTFMNKLDRD...


NEW RECORD 80
****Alignment****
sequence: gnl|BL_ORD_ID|1793 rlmF coding sequence
length: 308
e value: 4.42475e-101
RFTGSWTSSQALSSAQAHISANPGLNRAIRLRRQKESGAIFNGIIHKNEQ...
RFTGS TSSQALSSAQA ISANPGLNRAIRLRRQKESGAIFNGIIHKNEQ...
RFTGSETSSQALSSAQAIISANPGLNRAIRLRRQKESGAIFNGIIHKNEQ...
```

```python
# print BLAST results
counter = 0
for blast_record in blast_records:
    counter += 1
    print('\n\nNEW RECORD', counter)
    alignment = blast_record.alignments[0]
    for hsp in alignment.hsps:
        print("****Alignment****")
        print("sequence:", alignment.title)
        print("length:", alignment.length)
        print("e value:", hsp.expect)
        print(hsp.query[0:50] + "...")
        print(hsp.match[0:50] + "...")
        print(hsp.sbjct[0:50] + "...")
```

I saved best alignments to matches.csv. In each line there is the id of the input (protein) sequence, the id of the best-matching E.coli gene and the corresponding alignment e-value.
Below you can see first 10 entries in this file. Considering very small e-values, these alignments are of very good quality.

protein_id	ecoli_id	e_value
groupA_0	queA	6.43504e-101
groupA_1	hupA	3.99289e-57
groupA_2	hupB	5.66038e-55
groupA_3	marR	2.14227e-95
groupA_4	nanA	3.37178e-100
groupA_5	acnB	3.30561e-89
groupA_6	proP	8.5255e-84
groupA_7	fadB	8.06442e-92
groupA_8	rplM	1.32952e-97
groupA_9	dmsA	5.80295e-94

Part 2: *For each of the identified E.coli genes, you should find the associated promoter DNA sequence in the file proms_e_coli_fixed.fa. Please note that the input file contained sequences from groups A and B, we speculate (based on the empirical evidence), that these groups of genes should have different regulatory mechanisms. We need to identify 10 sequence motifs present in the promoters associated with group A and 10 motifs associated with group B, independently of each other. All motifs should be of length 15 . The result of this step should be two sets of 10 different motif position-specific matrices in a .pfm format.*

I used sequence ids from matches.csv file to select appropriate promoters. In total, I received 98 promoters: 65 corresponding to group A and 33 corresponding to group B. These promoters were saved to proms_A.fa and proms_B.fa respectively.
Having chosen associated promoters, I ran MEME Suite (http://meme-suite.org) to find motifs for groups A and B individually. I specified that, for each group there should be exactly 10 promoters of length exactly 15. I downloaded two MEME Suite reports from the website specified above and saved them as meme_A.txt and meme_B.txt.
I parsed these files to extract motifs in the form of biopython Motif objects. Each Motif consists of a number of biopython Seq objects (sequences). In this case, these are sequences of length 15 that contribute to the motif. Below you can find the excerpt from meme_B.txt report that corresponds to the first motif found by MEME Suite for group B. In this example, the Seq objects were created for 32 15-mers marked as Site.

```
--------------------------------------------------------------------------------
        Motif ATATTGCCGCAATAT MEME-1 sites sorted by position p-value
--------------------------------------------------------------------------------
Sequence name            Strand  Start   P-value                 Site
-------------            ------  -----  ---------           ---------------
dcp                           +      4  1.82e-09        AAA ATATTGCCGCAATAT ATTTTCTTCT
feaR                          +     83  1.43e-08 TTTGTGTTGC ATATTGCCGCAATCT TGA
ackA                          +     50  2.17e-08 CTGACGTTTT ATATTGCCGCAATAC ATTATAGGTA
uspC                          +     16  5.39e-08 TTGGGATTGC ATACTGCCGCGATAT GGAGTAAAGC
gdhA                          -     57  6.30e-08 ATAGATATAA ATATTGCCGCACTCT ATACGATTGT
eamA                          +     44  9.56e-08 AATTTCATCT ATATTGTCGCAATAT CTTAGCTGAA
queD                          -     52  9.56e-08 AATTTCTCTA ATATTGCAGCAATAT GCCGTAGAGT
codB                          +     60  1.23e-07 GGGTTTCAAA ATATTGCCGCTATGT ATTTCGTGTC
wecH                          +     66  1.54e-07 AACCTGCGAG ATATTCCCGGAATAT TGGTGGTGAT
rsmF                          +     52  2.20e-07 AAACTGCGCG ATATGGCCGCAATAC GTGGTACATG
gabD                          +     67  3.03e-07 ACCTTTGAAA ATATTGCCGCTGTAT GAAACTTAAC
rlmF                          +      9  3.97e-07   GGGGAATG ATGTTGCCGGCATAT TATTTACCCT
ispB                          +     17  3.97e-07 TATTCCTAAC AAATTGCCGTAATAT GCCTTTGTTC
kduI                          +     84  4.54e-07 TGTTTCGTTT ATATTGCCGGAAGAT GA
abgR                          +     75  4.54e-07 GTTATCGGTG ATATTGCCGAGATAT AGGTAAAAAT
asnC                          -     53  6.31e-07 TTCATTAAAA ATATTACCCCAATAT GAATGAATCA
mdtG                          +      3  9.60e-07         TC ATACTGCCGCTATGT CATAGCAATC
nac                           +     72  1.84e-06 ACTTACAACC GTGTTCCCGCGATAT TGCCAGTTCT
pphB                          +     84  2.03e-06 AAAACCATGC AAATTGTCGCAATGT AA
rimM                          +     27  4.17e-06 TCATGACCAC TTATAGCCGGGATAT TGCTTTGTTT
ugd                           +     80  4.54e-06 CCTGATAAGA GTATTGGCGCGCTAT GCTACG
dld                           -     79  6.38e-06    AGTTGTT TTACTGCGGCAATGT TTCCACTCCT
kdgR                          +     59  6.38e-06 AAATCAGAAC ATGTTGCCACAATAC TTTCGCACCA
mtlD                          +     41  6.38e-06 AGCCTCACCC TTATGGCCCGAATAT AAAACATTGA
sdiA                          -     37  7.50e-06 ATTATAAATG GTTTTGCGGCACTAT GGCGTTGCGG
prfC                          +     14  7.50e-06 TTGATGGGTA AAATAGCCGCAATTT TTCGTTTTCA
serA                          +      6  9.51e-06      TCCAG GAATTTCCGCACTAT TTACCCAATC
fklB                          +     64  1.41e-05 TTTTATACCC ATATGGCCCGCCTAT GATGACCACC
greB                          -     38  3.18e-05 TCTGTTTGAT TAATTGCAGTAATAT TATAACGTGA
ddlA                          +     46  4.49e-05 CGTCTAACAC ATAATAACGCAATAT CCACGACAAA
rhtA                          -     19  9.85e-05 TCTAGATATT ATGTTACGGCGGTCT GCCTGGTTCA
ligB                          -     73  1.05e-04 CACAGGAAGA AATTTCCCGCCTTAT TGTGCCAAGA
```

Having created Motif objects, I saved them to .pfm files. To do so, I created text .pfm files and wrote there pfm matrices that I constructed using script Motif.format('pfm'). Each file stores a single matrix.
I saved all 20 motifs as A0.pfm,…, A9.pfm, B0.pfm,…, B9.pfm.

Part 3: *Given the two sets of motifs, we would like to select only the motifs that are specific to group A or group B.*

To do this, for each motif I needed to calculate a number of hits in sequences from groups A and B separately. By a hit we shall understand a situation when a motif aligned with a sequence's 15-mer has a positive log-odds score. To calculate log-odds scores, we must have log-odds matrices that are motif-specific. I parsed both MEME reports once more to extract log-odds matrices for all 20 motifs. Below you can find an excerpt from meme_B.txt that corresponds to the first motif from group B.

```
--------------------------------------------------------------------------------
    Motif ATATTGCCGCAATAT MEME-1 position-specific scoring matrix
--------------------------------------------------------------------------------
log-odds matrix: alength= 4 w= 15 n= 3010 bayes= 6.54013 E= 7.3e-051
   134  -1164    -71   -124
   -66  -1164  -1164    146
   146  -1164    -71   -224
  -324   -112  -1164    156
  -224  -1164   -112    151
  -166   -112    193   -324
  -324    210   -271   -224
  -224    205   -112  -1164
  -324   -112    210  -1164
  -324    181    -12   -224
   108   -112    -12   -166
   134    -39   -171   -324
 -1164  -1164   -271    171
   134   -112    -71   -324
 -1164   -112  -1164    162
```

Having extracted all 20 matrices, I defined two functions: log_odds_score(sequence, log_odds_matrix) and log_odds_hits(sequence, log_odds_matrix).
The first one receives on the input a sequence of length 15 (a 15-mer) and a log-odds matrix specific to a certain motif. The function returns a log-odds score for this sequence and this motif.
The second one receives on the input a full, long sequence and a motif-specific log-odds matrix. It returns a number of hits. For each 15-mer of the sequence it is checked whether log_odds_score returns a positive value. If so, a total number of hits is increased by one.
At this point, I had a list of 20 motifs and tools to calculate the numbers of hits in promoters from groups A and B. Now, the goal is to check the enrichment level of each motif in groups A and B.
To do this, I used the binomial test, where, for each 15-mer, a hit is considered a success and lack of it is a failure. Firstly, I iterated over motifs from group A. Having chosen one, I calculated the numbers of hits in groups A and B and ran two binomial tests:

- *number of successes*: the number hits in group A, *number of trials*: the number of 15-mers in all promoters from group A, *hypothesised probability of success*: the number of hits in grup A divided by the number of all 15-mers in group A

- *number of successes*: the number hits in group B, *number of trials*: the number of 15-mers in all promoters from group B, *hypothesised probability of success*: the number of hits in grup A divided by the number of all 15-mers in group A

Analogically, I did similar tests for motifs from group B (hypothesised probability of success was changed to the number of hits in grup B divided by the number of all 15-mers in group B). All tests returned p-values. I saved them to enrichments.csv. I present these results on the next page.
Binomial tests suggest that, assuming 5% p-value threshold, there is a single A-specific well-enriched motif (A0) and three B-specific well-enriched motifs (B0, B1 and B3). All other 16 motifs are of bad quality and do not differentiate well between groups A and B.

motif_id	hits_A	hits_B	evalue_A	evalue_B
A0	109	33	0.9999999999999446	0.00015516621789283104
A1	3	1	1.0	1.0
A2	4	0	1.0	0.18312831240462793
A3	2	0	1.0	0.6341699931406767
A4	2	1	1.0	0.999999999999945
A5	1	0	1.0	1.0
A6	2	0	1.0	0.6341699931406767
A7	3	0	1.0	0.42276871150603257
A8	2	0	1.0	0.6341699931406767
A9	2	0	1.0	0.6341699931406767
B0	19	31	1.7874546231274505e-08	0.999999999999913
B1	2	8	0.0001612231505480804	1.0
B2	0	1	0.4346194723565042	1.0
B3	1	5	0.0022713387247217725	1.0
B4	0	1	0.4346194723565042	1.0
B5	0	1	0.4346194723565042	1.0
B6	0	1	0.4346194723565042	1.0
B7	8	6	0.5394683237994014	0.9999999999999304
B8	0	1	0.4346194723565042	1.0
B9	0	1	0.4346194723565042	1.0