# Lab 16: Motif Finding

**BACKGROUND:** You identified a human gene called **JUNB** that is relevant to your research. You now want to see how the gene is regulated by identifying transcription factors (TFs) that regulate the gene. You also want to identify any domains in its protein sequence.

## TASK 1 Objectives:

*Retrieve the promoter sequence and identify potential TF binding sites.*

**A.** Extract the promoter sequence from the gene.

1. Go to the UCSC Table Browser
2. In the row that starts with the word "**clade**", choose "**human**" from the "**genome**" field
3. In the row starting with the word "**region**", enter **JUNB** in the *position* field, and click on the *Lookup* button
4. Click on the top link that references **JUNB**: `JUNB (uc002mvc.4) at chr19:12791496-12793315 - Homo sapiens jun B proto-oncogene (JUNB), mRNA. (from RefSeq NM_002229)` -- this will bring you back to the Table Browser page.
5. When you are back in the Table Browser page, in the row "**region**" select "position" radio button, and in the row starting with "**output format**", select *sequence* from the dropdown
6. Click the **get output** button, choose *genomic*, then press the **submit** button
7. Check the **Promoter/Upstream** checkbox, and enter 500 in the text box
8. Uncheck all other boxes and press the **get sequence** button
9. Copy and paste the FASTA result into a file for later use.

**B.** Look for transcription factor binding site (TFBS) motifs.

1. Go to AME web site: https://meme-suite.org/meme/tools/ame
2. Upload the sequences that you downloaded from UCSC
3. Keep the defaults for motif database, sequence scoring method, and motif enrichment test.
4. Click the 'Submit' button.
5. Once the results are ready (it might take several minutes), click the 'AME HTML output' link, and attach a screeshot of the obtained motifs to this notebook

**Questions:**

> AME is part of which suite?
> What is Meme suite used for?
> List the motif databases that were used in your search (hint! Look at AME's results)

Provide your answers here. 1. AME is part of which suite? MEME suite 2. What is Meme suite used for? The MEME Suite is used to discover novel motifs in collections of unaligned nucleotide or protein sequences, and to perform a wide variety of other motif-based analyses.(from the website overview: https://meme-suite.org/meme/doc/overview.html) 3. List the motif databases that were used in your search (hint! Look at AME's results) Vertebrates (In vivo and in silico)

## TASK 2 Objectives:

*Find motifs in protein sequences without looking at the promoter.*

**A.** Retrieve the protein sequence for **JUNB_HUMAN**. Search for it at: http://www.uniprot.org/

**B.** Save the sequence in a file; you will need it for **Task 3**. *Hint:* You can create new text files in jupyter by choosing `New -> Text File` in the file browser.

**C.** Search for domains:

1. Go to the HMMER web page
2. Paste the JUNB protein sequence from above into the first box
3. Click '**Submit**' to search for protein motifs in your protein sequence.
4. Hmmer will display the protein sequence and the locations of significant protein motif hits.

**Question:**

> Are the results surprising?

In statistical way it is suprising because the e-value for Jun is so low, meaning it would be very unlikely to find this motif due to random chance, whereas with other motifs maybe it could be more common

> Are there multiple protein motifs?

Yes, there are Jun，bZIP_1，bZIP_2,and bZIP_Maf.

junb.fasta### TASK 3 Objectives:

**BACKGROUND**: A fast way to search for leucine zipper domains in a protein sequence is to check if there is a small pattern containing an “L” every 7 amino acids (A leucine zipper pattern is an L with 6 other amino acids, and then another L with 6 other amino acids, and then another L, so on and so forth.). Use your python skills to take a sequence and identify whether that pattern exists.

**Note**: Regular expressions are very powerful and can get complicated. This is a very basic tutorial. For more in-depth knowledge, go to http://docs.python.org/library/re.html. You will use the regular expressions library for this exercise. But first you will practice basic expression and syntax. A regular expression is a consensus sequence-like pattern. Here are two example regular expressions:

| Regular Expression | Description | Matching sequences |
|---|---|---|
| `motif = 'A[CT]G[AG]'` | Position 1: A<br>Position 2: C or T<br>Position 3: G<br>Position 4: A or G | ACGA<br>ATGA<br>ACGG<br>ATGG |
| `motif = 'A.G.'` | Position 1: A<br>Position 2: A or C or T or G<br>Position 3: G<br>Position 4: A or C or T or G | AAGA, AAGT, AAGC AAGG,<br>ATGA, ATGT ATGC, ATGG,<br>ACGA ACGT, ACGC, ACGG<br>AGGA, AGGT, AGGC AGGG |

A dot (.) is a wildcard that stands for any character at a single position. Another useful wildcard often used with a dot is the asterisk (*). A * refers to repeats of any character. [ACGT]* refers to repeats of (A, C, G, or T). The syntax in a Python script is:

```
import re
sequence = 'TACACGTATAC'
motif = 'A.G.'
result = re.search(motif, sequence)
```

If the pattern is not found, result will be 'None'. Search for 'XYZ' in the above sequence and print the result to see what happens when a pattern is not found.

You can get the start and end positions of the motif match:

```
start = result.start()
end = result.end()
```

To extract the exact motif in the sequence, you can slice it out:

```
sequence[start:stop]
```

## TASK 3 Objectives:

1. Write python code below which operates on the sequence file `junb.fasta`.
2. Use regular expressions. To do this, you must first `import re`. This needs to be done at the very beginning of your code.
3. Read the sequence from the file and save it in a variable, e.g. sequence.
4. Create your motif variable for leucine zippers using dots (.). Hint: a motif with a lysine (K) repeated twice with two amino acids in the middle would be: `motif = 'K..K'`. Remember that the leucine zipper pattern is an L with 6 other amino acids, and then an L with 6 other amino acids, and then another L, so on and so forth. Note: K is for lysine, L is for leucine!
5. Use `re.search` to check if the leucine zipper pattern. See syntax on previous page.
6. Print the location and motif within the sequence. How does your Python script result compare with the online tool result?
7. Upload your jupyter notebook to Blackboard for lab credit.

In [21]:
```
#write your python code here
import re
#Read the sequence from the file and save it in a variable
sequence=''
```

```
file = open('junb.fasta','r')
s=file.readlines()

# Loop through lines
for l in s:
    if l[0]!='>': #delete the header line
        sequence+=l.strip('\n') #add sequence w/t newline characters
#Create motif variable for leucine zippers using dots (.)
motif = '(L.{6}){3,}L'
result = re.search(motif, sequence)
start = result.start()
end = result.end()
print("Location:",result.span(),'\n'"Motif:",sequence[start:end])
```

```
Location: (295, 324)
Motif: LEDKVKTLKAENAGLSSTAGLLREQVAQL
```

The motif's location in python script result is more accurate than that in online tool result.

```
In [ ]:
```