Assignment 6: Motif Finding
Bio5488
2/22/19
Assignment 6: Motif finding

• Input
  • Promoter sequences
  • PWMs of DNA-binding proteins

• Goal
  • Find putative binding sites in the sequences by scanning the sequences for matches to the PWM

• Output
  • List of the locations and scores of putative binding sites
Input files

• Promoter sequences
  • Just the sequence, i.e., not a fasta

• PWMs of DNA-binding proteins
  • *Whitespace*-delimited
  • $a_{ij} =$ score for base $i$ at position $j$
    • Rows correspond to A, C, G, & T
    • Columns correspond to positions
    • The *higher* the score, the *better* the score

<table>
<thead>
<tr>
<th>Base</th>
<th>Position 0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-5</td>
<td>-2</td>
<td>9</td>
<td>7</td>
<td>-6</td>
<td>10</td>
</tr>
<tr>
<td>C</td>
<td>-3</td>
<td>6</td>
<td>10</td>
<td>2</td>
<td>-5</td>
<td>-2</td>
</tr>
<tr>
<td>G</td>
<td>3</td>
<td>-6</td>
<td>-9</td>
<td>-1</td>
<td>5</td>
<td>-10</td>
</tr>
<tr>
<td>T</td>
<td>7</td>
<td>-6</td>
<td>-7</td>
<td>-3</td>
<td>10</td>
<td>-4</td>
</tr>
</tbody>
</table>

Example PWM file

-5 -9 4 5 -3 2
6 -5 10 -1 0 10
-10 -1 4 3 10 -4
6 0 -1 10 -3 1
Assignment TODOs

• Determine the highest affinity binding site for each PWM
  • Calculate by hand or write a script 😊

• Comment the starter script `scan_sequence.py`
  • Comment the existing code blocks
  • Comment the user-defined functions with function docstrings
Function docstrings

• **Purpose**: tells the reader how to use the function

• **Guidelines for what to include**
  • Describe what the function does
  • Describe the input argument(s)
  • Describe the output value(s)

• **Where to learn more:**
  • PEP 257: [https://www.python.org/dev/peps/pep-0257/](https://www.python.org/dev/peps/pep-0257/)
**Example of a function docstring**

```python
def calc_fishers_linear_discriminant(group1_values, group2_values):
    """Calculates the fisher's linear discriminant between two groups.

    The formula used to calculate fisher's linear discriminant is
    (mean1 - mean2)^2 / (stddev1^2 + stddev2^2)

    Args:
    - group1_values: list of floats for group 1
    - group2_values: list of floats for group 2

    Returns:
    - fisher's linear discriminant

    """

    # Rest of code goes here
```
Retrieving a function’s docstring

Call `help`

```python
>>> help(calc_fishers_linear_discriminant)
```

Function’s docstring is returned

```
Help on function calc_fishers_linear_discriminant in module __main__:

calc_fishers_linear_discriminant(group1_values, group2_values)
    Calculates the fisher's linear discriminant between two groups.

    The formula used to calculate fisher's linear discriminant is
    (mean1 - mean2)^2 / (stdev1^2 + stdev2^2)

    Args:
        group1_values: list of floats for group 1
        group2_values: list of floats for group 2

    Returns:
        fisher's linear discriminant
```

Docstrings are also used by third-party programs to create user-friendly documentation for your project.
Assignment TODOs (cont.)

• Determine the highest affinity binding site for each PWM
  • Calculate by hand or write a script 😊
• Comment the existing code
  • Comment the user-defined functions with function docstrings
• Modify the script to scan the reverse complement of the input sequence
• Modify the script to report only report hits that have scores above a given threshold
• Scan promoters (n = 2) to find putative binding sites for each DNA-binding protein (n = 2)
• Answer follow-up questions
Indexing

- Indexing is somewhat arbitrary; however it’s important to follow conventions:
  - The start position of a feature is smaller than the stop position
  - The coordinates are relative to the forward strand

Consensus sequence

```
A A G
```

Schematic of putative binding sites

```
5’-A A G C T T-3’
3’-T T C G A A-5’
```

Output table

<table>
<thead>
<tr>
<th>strand</th>
<th>sequence</th>
<th>position</th>
<th>score</th>
</tr>
</thead>
<tbody>
<tr>
<td>forward</td>
<td>AAG</td>
<td>0</td>
<td>12.8</td>
</tr>
<tr>
<td>reverse</td>
<td>AAG</td>
<td>3</td>
<td>15.6</td>
</tr>
</tbody>
</table>
Python list comprehensions

- Purpose: create lists in 1 line of code
  - There are also dictionary comprehensions that work similarly

<table>
<thead>
<tr>
<th>Code template</th>
<th>Example</th>
</tr>
</thead>
</table>
| As a for loop          | ```python
x = []
for i in range(5):
x.append(i**2)
``` |
| List comprehension      |                                                                         |
Python list comprehensions with filtering

<table>
<thead>
<tr>
<th>Code template</th>
<th>Example</th>
</tr>
</thead>
</table>
| As a for loop | x = []
for i in range(5):
    if i % 2 == 0: # if i is even
        x.append(i**2) |

<table>
<thead>
<tr>
<th>List comprehension</th>
<th></th>
</tr>
</thead>
</table>

• Where to learn more:
  • List comprehension PEP: [https://www.python.org/dev/peps/pep-0202/](https://www.python.org/dev/peps/pep-0202/)
  • Dict comprehension PEP: [https://www.python.org/dev/peps/pep-0274/](https://www.python.org/dev/peps/pep-0274/)
**Python’s `zip` function**

- **Purpose:** “zip” together lists
  - Returns a list* of tuples where the $i^{\text{th}}$ tuple contains the $i^{\text{th}}$ element from each of the input lists

<table>
<thead>
<tr>
<th>Code template</th>
<th>Example</th>
</tr>
</thead>
</table>
| `<zipped_list> = list(zip(<list1>, <list1>, ...))` | `x = [0, 1, 2]`  
  `y = [0, 1, 4]`  
  `coords = list(zip(x, y))`  
  `>>> coords`  
  `[(0, 0), (1, 1), (2, 4)]` |

- Zipped lists can be unzipped (`zip(*coords)`)  
- Where to learn more  
  - Python.org documentation:  
    [https://docs.python.org/3.4/library/functions.html#zip](https://docs.python.org/3.4/library/functions.html#zip)

*It’s really an iterator, one of list’s close cousins*
Printing formatted strings in Python with `format`

- **Purpose:** make your print statements print “pretty” output, e.g., tables

- **`format`** transforms a “template string” by substituting placeholders with formatted values
  - Placeholders are enclosed in `{}` and specify how the value should be formatted

<table>
<thead>
<tr>
<th>Not so pretty</th>
<th>Pretty</th>
</tr>
</thead>
</table>

- Where to learn more:
  - Python.org tutorial: [https://docs.python.org/3.4/tutorial/inputoutput.html#fancier-output-formatting](https://docs.python.org/3.4/tutorial/inputoutput.html#fancier-output-formatting)
  - Python.org documentation: [https://docs.python.org/3.4/library/string.html#formatstrings](https://docs.python.org/3.4/library/string.html#formatstrings)
Assignment 6: requirements

• Due in 1 week (3/1/19) at 10 AM

• Your submission directory should contain
  • A modified scan_sequence.py that is well commented and contains a docstring for each user-defined function
  • A README.txt with the answers to the questions and the commands/work you used to arrive at the answer