pytfmpval Documentation

Release 0.2.0

Jared Andrews

Sep 01, 2017

Contents

This Python package serves as a wrapper for the incredibly useful TFM-Pvalue C++ program. It allows users to determine score thresholds for a given transcription factor position frequency matrix associated with a specific p-value. Naturally, it can also perform the reverse, quickly calculating an accurate p-value from a score for a given motif matrix.

`pytfmpval` allows this functionality to be easily utilized within a Python script, module, or interactive session.

CHAPTER 1

Installation

pytfmpval is on PyPI, so you can install via `pip` easily:

```
pip install pytfmpval
```

CHAPTER 2

What Works and What Doesn't

Currently, the `TFMpvalue-pv2sc` and `TFMpvalue-sc2pv` programs are implemented, though there are also plans to implement the remaining `TFMPvalue-fastpvalue`, `TFMpvalue-distrib`, and `TFMpvalue-lazydistrib` programs.

CHAPTER 3

A Simple Example

JASPAR is a very highly-touted transcription factor motif database from which motif count matrices can be downloaded for a large variety of organisms and transcription factors. There exist numerous other motif databases as well (TRANSFAC, CIS-BP, MEME, HOMER, WORMBASE, etc), most of which use a relatively similar format for their motifs. Typically, a motif file consists of four rows or columns with each position in a given row or column corresponding to a base within the motif. Sometimes there is an comment line started with >. The row or column order is always `A, C, G, T`. In this example, the motif consists of four rows corresponding to the 16 positions of the motif with counts for each base at each position.

```
>>> from pytfmpval import tfmp
>>> m = tfmp.create_matrix("MA0045.pfm")
>>> tfmp.score2pval(m, 8.7737)
9.992625564336777e-06
>>> tfmp.pval2score(m, 0.00001)
8.773708000000001
```

This could also be done by creating a string for the matrix by concatenating the rows (or columns) and using the `read_matrix()` function. This method is usually easier, as it allows the user to parse the motif file as necessary to ensure a proper input. It's also more fitting for high-throughput use.

```
>>> from pytfmpval import tfmp
>>> mat = (" 3  7  9  3 11 11 11  3  4  3  8  8  9  9 11  2"
...         " 5  0  1  6  0  0  0  3  1  4  5  1  0  5  0  7"
...         " 4  3  1  4  3  2  2  2  8  6  1  4  2  0  3  0"
...         " 2  4  3  1  0  1  1  6  1  1  0  1  3  0  0  5"
...        )
>>> m = tfmp.read_matrix(mat)
>>> tfmp.pval2score(m, 0.00001)
8.773708000000001
>>> tfmp.score2pval(m, 8.7737)
9.992625564336777e-06
```

CHAPTER 4

Full tfmp Module Reference

`tfmp.`**`create_matrix`**(*matrix_file, bg=[0.25, 0.25, 0.25, 0.25], mat_type='counts', log_type='nat'*)

From a JASPAR formatted motif matrix count file, create a Matrix object.

This function also converts it to a log-odds (position weight) matrix if necessary.

Parameters

- **`matrix_file`** (*`str`*) – White-space delimited string of row-concatenated motif matrix.
- **`bg`** (*`list of floats`*) – Background nucleotide frequencies for [A, C, G, T].
- **`mat_type`** (*`str`*) – Type of motif matrix provided. Options are: "counts", "pfm", "pwm". "counts" is for raw count matrices for each base at each position. "pfm" is for position frequency matrices (frequencies already calculated. "pwm" is for position weight matrices (also referred to as position-specific scoring matrices.)
- **`log_type`** (*`str`*) – Base to use for log. Default is to use the natural log. "log2" is the other option. This will affect the scores and p-values.

Returns Matrix in pwm format.

Return type m (pytfmpval Matrix)

`tfmp.`**`read_matrix`**(*matrix, bg=[0.25, 0.25, 0.25, 0.25], mat_type='counts', log_type='nat'*)

From a string of space-delimited counts create a Matrix object.

Break the string into 4 rows corresponding to A, C, G, and T. This function also converts it to a log-odds (position weight) matrix if necessary.

Parameters

- **`matrix_file`** (*`str`*) – White-space delimited string of row-concatenated motif matrix.
- **`bg`** (*`list of floats`*) – Background nucleotide frequencies for [A, C, G, T].
- **`mat_type`** (*`str`*) – Type of motif matrix provided. Options are: "counts", "pfm", "pwm". "counts" is for raw count matrices for each base at each position. "pfm" is for position frequency matrices (frequencies already calculated). "pwm" is for position weight matrices (also referred to as position-specific scoring matrices.)

- **log_type** (*str*) – Base to use for log. Default is to use the natural log. “log2” is the other option. This will affect the scores and p-values.

Returns Matrix in pwm format.

Return type m (pytfmpval Matrix)

`tfmp.`**`score2pval`**(*matrix*, *req_score*, *mem_thresh=2.0*)

Determine the p-value for a given score for a specific motif PWM.

Parameters

- **matrix** (*pytfmpval Matrix*) – Matrix in pwm format.
- **req_score** (*float*) – Requested score for which to determine the p-value.
- **mem_thresh** (*float*) – Memory in GBs to remain free to system. Once passed, the closest p-val approximation will be returned instead of the exact p-val. Should only occur rarely with very long and degenerate motifs. Used to help ensure the system won’t run out of memory due to these outliers. This is only calculated after each pass, each of which is more time and memory intensive than the last, so changing this value isn’t recommended unless accuracy out to the 8th decimal place is really necessary.

Returns The calculated p-value corresponding to the score.

Return type pv (float)

`tfmp.`**`pval2score`**(*matrix*, *pval*, *mem_thresh=2.0*)

Determine the score for a given p-value for a specific motif PWM.

Parameters

- **matrix** (*pytfmpval Matrix*) – Matrix in pwm format.
- **pval** (*float*) – p-value for which to determine the score.
- **mem_thresh** (*float*) – Memory in GBs to remain free to system. Once passed, the closest p-val approximation will be returned instead of the exact p-val. Should only occur rarely with very long and degenerate motifs. Used to help ensure the system won’t run out of memory due to these outliers. This is only calculated after each pass, each of which is more time and memory intensive than the last, so changing this value isn’t recommended unless accuracy out to the 8th decimal place is really necessary.

Returns The calculated score corresponding to the p-value.

Return type score (float)

CHAPTER 5

Contribute

Any and all contributions are welcome. Bug reporting via the Issue Tracker is much appeciated. Here's how to contribute:

1. Fork the pytfmpval repository on github (see forking help).
2. Make your changes/fixes/improvements locally.
3. Optional, but much-appreciated: write some tests for your changes. (Don't worry about integrating your tests into the test framework - writing some in your commit comments or providing a test script is fine. I will integrate them later.)
4. Send a pull request (see pull request help).

CHAPTER 6

Reference

Efficient and accurate P-value computation for Position Weight Matrices
H. Touzet and J.S. Varré
Algorithms for Molecular Biology 2007, 2:15

CHAPTER 7

License

This project is licensed under the GPL3 license. You are free to use, modify, and distribute it as you see fit. The program is provided as is, with no guarantees.

Index