06/03/2024

Pyxis

A command-line tool for detecting enriched motifs in genomic peak regions.

EVELYN QUAN, CSE 185

HTTPS://GITHUB.COM/EVELYNSQ/PYXIS

Table of Contents

Basic Usage

Installation:

```
git clone https://github.com/evelynsq/pyxis.git
cd pyxis
python setup.py install [--user]
```

Dependencies:

- pyfaidx
- numpy
- pandas
- scipy
- seqlogo

General Command:

```
pyxis peaks.bed ref.fa motif.pwms [--background bg.bed] [other options]
```

Options for pyxis

-h / --help	**Help message**
-o FILE / --output FILE	Direct **output stream** to specified file
-b FILE / --background FILE	**Background peaks** in motif-finding
-p PVAL / --pval PVAL	**Threshold p-value** for enrichment significance
-s / --seqlogo	Generate **sequence logo** for motifs
--pseudo VAL	**Pseudocount** for offsetting log2-odds PWM scores to PPM for generating sequence logos
--version	Print **version**

Example Output

pyxis_enrichments.tsv

motif_name	pval	log_pval	num_peak_motif	pct_peak_motif	num_bg_motif	pct_bg_motif	enriched_status
BACH2_HUMAN.H11MO.0.A	1.0	0.0	3	0.3	2	0.2	Non-sig
ALX3_HUMAN.H11MO.0.D	1.0	0.0	2	0.2	3	0.3	Non-sig
ELK1_HUMAN.H11MO.0.B	1.0	0.0	0	0.0	1	0.1	Non-sig
KAISO_HUMAN.H11MO.1.A	1.0	0.0	2	0.2	1	0.1	Non-sig
MLX_HUMAN.H11MO.0.D	1.0	0.0	1	0.1	1	0.1	Non-sig

- Most enriched motifs will be at top of list.

Columns:

1. Name of motif
2. P-value of enrichment significance
3. Log10 p-value
4. Number of peaks matching motif
5. Fraction of peaks matching motif
6. Number of background sequences matching motif
7. Fraction of background sequences matching motif
8. Was motif found to be significantly enriched?

Benchmarking

- Runtime of pyxis **myutils.py functions**
- Runtime for executing pyxis **basic usage commands** on example data + real dataset
- Comparisons to HOMER's **findMotifsGenome.pl** runtime and sequence logos
- **Peak memory usage** comparisons

Pyxis Myutils.py Functions

Function	Mean Runtime Per Loop ± Std. Dev. of 7 Runs (10000 loops each)
ReadBED	183 µs ± 9.18 µs
ReadPWMS	247 µs ± 6.29 µs
WriteFastaSeq	8.87 µs ± 139 ns
ComputeNucFreqs	23.9 µs ± 709 ns
RandomBkSequence	88.2 µs ± 1.15 µs
ScoreSeq	1.48 µs ± 14.5 ns
ReverseComplement	6.29 µs ± 46.8 ns
FindMaxScore	141 µs ± 2 µs
RandomSequence	23.1 µs ± 2.67 µs
GetThreshold	637 µs ± 3.14 µs
ComputeEnrichment	2.15 ms ± 280 µs
pwm_to_ppm	29.5 µs ± 496 ns

Fastest!

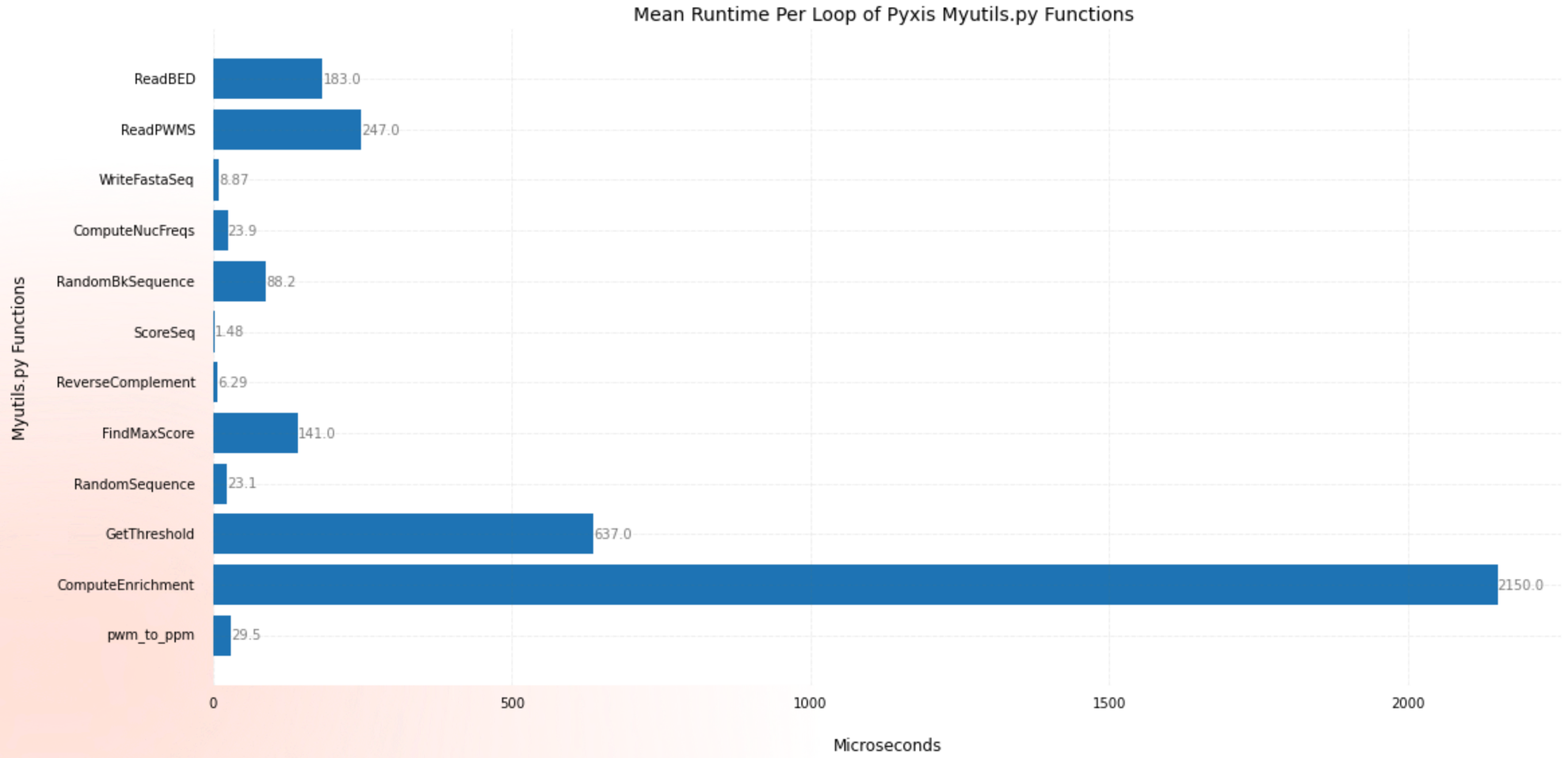
O(*n*) runtime: *n* is the length of the PWM being used for scoring

Slowest

Fisher's Exact Test can be slow - especially for large values in your contingency table

HTTPS://GITHUB.COM/EVELYNSQ/PYXIS

Pyxis Myutils.py Functions



HTTPS://GITHUB.COM/EVELYNSQ/PYXIS

Pyxis Options

Command	Mean Runtime Per Loop (10 loops, best of 5)
Basic Usage	1.97 sec
Basic Usage + Background Provided	2.02 sec
Basic Usage + P-value Provided	2.03 sec
Basic Usage + Seqlogo	2.82 sec
Basic Usage + Seqlogo + Pseudovalue	2.78 sec

About the same!

Including *-s / --seqlogo* option can make the command run around **1.4x slower** than the basic usage command

Results on Real Data

Tool	Dataset	Runtime Per Loop (1 loop, best of 5)
Pyxis	Example Data	1.81 sec
Pyxis	ISL1	341 sec
HOMER	Example Data	186 sec
HOMER	ISL1	3.12e+03 sec

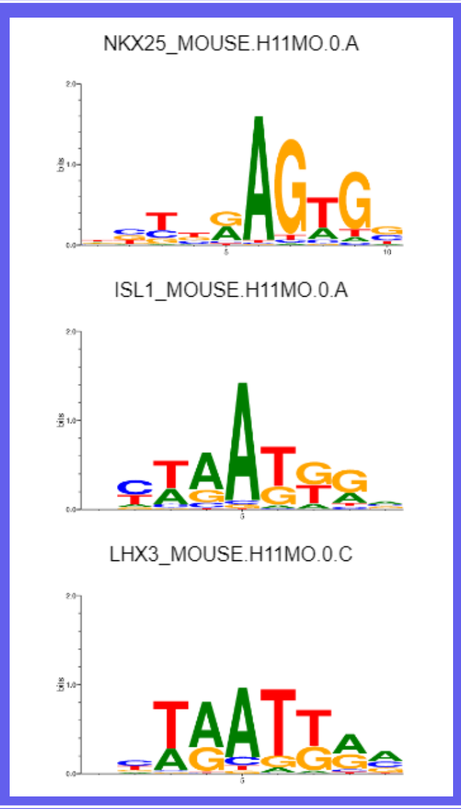
motif_name	pval	log_pval	num_peak_motif	pct_peak_motif	num_bg_motif	pct_bg_motif	enriched_status
ISL1_MOUSE.H11MO.0.A	3.8681244406275597e-295	-294.4125	4835	0.781856403622251	2879	0.4655562742561449	Sig
NKX25_MOUSE.H11MO.0.A	6.859704709383046e-281	-280.16369	4801	0.7763583441138422	2890	0.46733505821474774	Sig
LHX3_MOUSE.H11MO.0.C	1.0008831183450168e-267	-266.99962	4212	0.6811125485122898	2286	0.36966364812419145	Sig
NKX22_MOUSE.H11MO.0.A	4.799427573329636e-218	-217.31881	5292	0.8557567917205692	3761	0.6081824062095731	Sig
BACH2_HUMAN.H11MO.0.A	4.856501062484705e-217	-216.31368	4781	0.773124191461837	3115	0.503719275549806	Sig
KAISO_HUMAN.H11MO.1.A	1.840030567469271e-86	-85.73517	2888	0.4670116429495472	1826	0.29527813712807244	Sig
MLX_HUMAN.H11MO.0.D	8.11901614846211e-82	-81.0905	4744	0.7671410090556274	5535	0.8950517464424321	Sig
ALX3_HUMAN.H11MO.0.D	3.2897559701277345e-16	-15.48284	6100	0.9864165588615783	5960	0.963777490297542	Sig
GATA4_MOUSE.H11MO.0.A	0.0009722257333249527	-3.01223	6184	1.0	6173	0.9982212160413971	Non-sig
ELK1_HUMAN.H11MO.0.B	0.03549233860694681	-1.44987	6176	0.9987063389391979	6164	0.9967658473479948	Non-sig

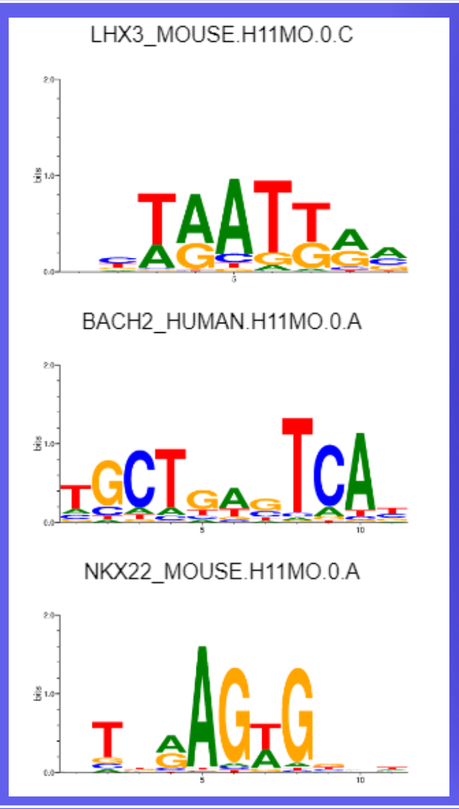
Dataset: Peaks for the ISLET1 (ISL1) transcription factor, which influences early cardiac cell fate. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10679653/

- We find the ISL1 motif and several other motifs mentioned in the paper (NKX25, LHX3) to be significantly enriched with pyxis!

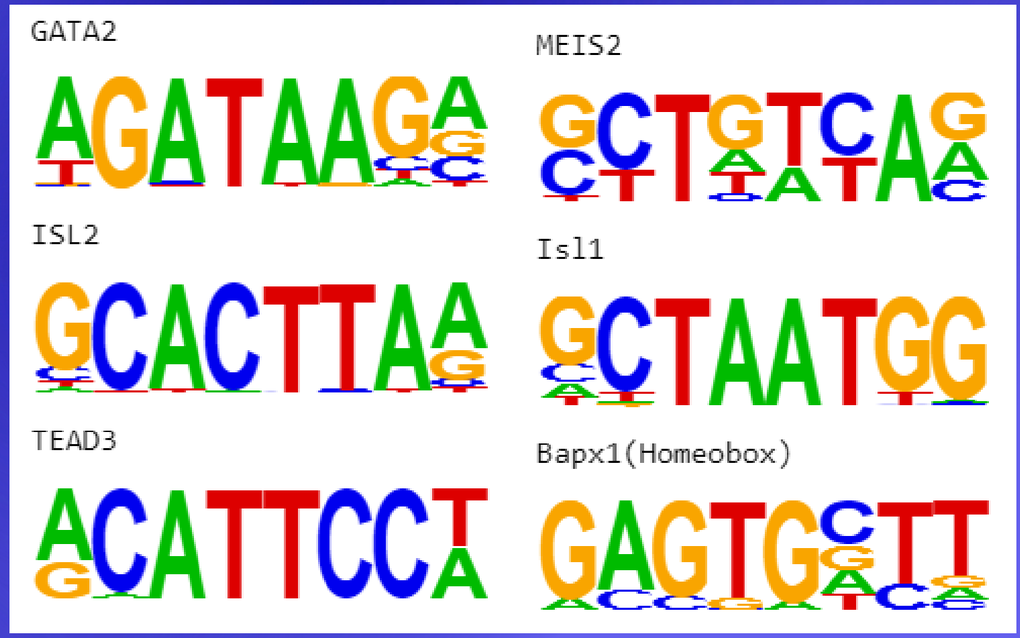
SeqLogo Comparisons

pyxis



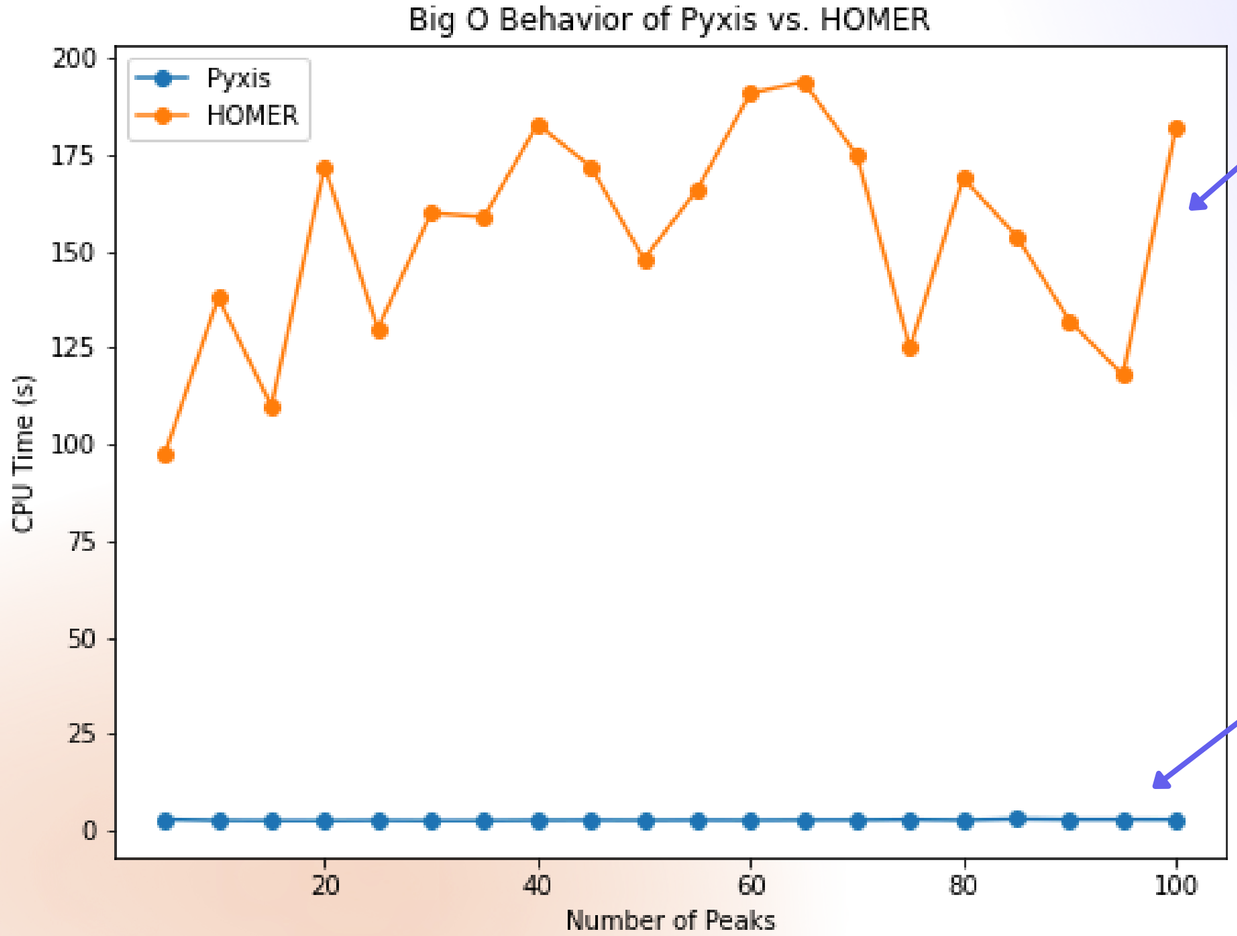


findMotifsGenome.pl



HTTPS://GITHUB.COM/EVELYNSQ/PYXIS

Big O Behavior

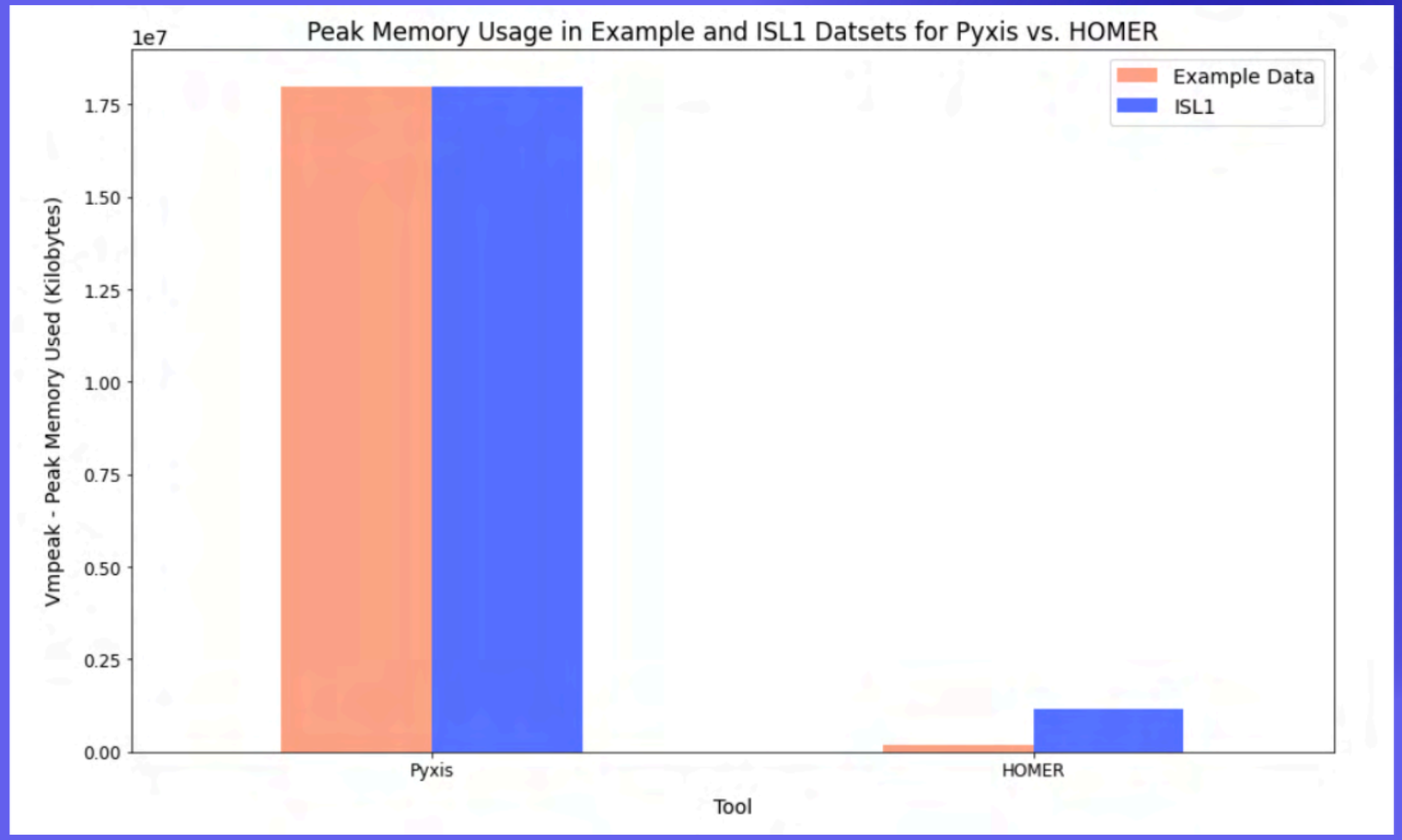


High variability due to DataHub?

- Could run a greater number of loops with *timeit* to be sure (would take a very long time)

Simpler + faster **but** less accurate and intricate

Memory Usage

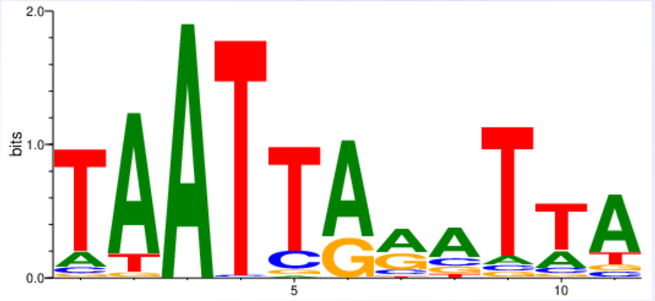


HOMER has a drastically lower peak memory usage.

Challenges Encountered

- **Long wait times** when running *timeit* on larger datasets. Spent a lot of time on benchmarking just waiting for commands to finish running on DataHub.

- Trying to **get HOMER to work with example test files**– Figuring out which options to use were confusing.



- Figuring out how to create my own **implementation of *pwm_to_ppm*** to plot sequence logos

Thank You!

Feel free to try pyxis out here:

https://github.com/evelynsq/pyxis

Follow installation instructions listed on the README to get started!

Questions?