**Python Scripts – READ ME**

================

This repository contains the source code for running the analysis described in the paper. Not all of the scripts in this repository will be relevant to the complete replication of the study - if you wish to do this, please send me an e-mail and we can sort something out (a.t.ho@bath.ac.uk).

###Instructions:

- All files run from the root directory.

- Python scripts are found in 'Python\_scripts' (all scripts are run using Python3)

- R scripts are found in 'R\_scripts'

- UTR sequences are found in ‘FASTA files’

###The folders found in my working directory are labelled 0-9, all available on request:

- 0 - Scripts

- 1 - Genome files

- 2 – Files for D. discoideum intraspecies analysis

- 3 - PaxDB raw data and appropriate EMBL genomes

- 4 - Outputs (includes CSVs)

###Description of python scripts:

+4T\_simulations.py – Simulates dinucleotide-controlled random sequences for +4T comparison

Fourth\_site\_freqs.py – Calculate fourth site nucleotide frequencies

0\_get\_UTRs\_paxdb\_gtf.py – Extract UTR sequences from PaxDb genomes using GTF annotations

0\_get\_UTRs\_paxdb\_gff.py – Extract UTR sequences from PaxDb genomes using GFF annotations

0\_get\_UTRs.py – Extract UTR sequences from genomes using GTF annotations

1\_Chi\_results\_with\_pos1.py – Calculate no. of genomes with ASC enrichment (includes position +1)

1\_Chi\_results\_without\_pos1.py – Calculate no. of genomes with ASC enrichment (excludes position +1)

1\_A+C\_chi.py – Calculate positional enrichment against A+C null

1\_sims\_chi.py – Calculate positional enrichment against dinucleotide-controlled simulations

2\_dicty\_get\_fold\_change.py – Calculate and extract fold-change values for each D. discoideum gene

2\_dicty\_FC\_analysis.py – Fold-change analysis for D. discoideum

2\_dicty\_AC\_null.py – Comparison of D. discoideum single and social growth genes to A+C null

2\_dicty\_sims\_null.py – Comparison of D. discoideum single and social growth genes to dinucleotide sims

2\_dicty\_fourth\_t.py – Calculate +4T frequencies for TAA, TGA, TAG-terminating single and social genes

2\_dicty\_ASC\_fourth\_t\_proportions.py – Calculate proportion of +4T-genes that also have a proximal ASC

2\_dicty\_stop\_usage.py – Count the number of TAA, TGA, TAG-terminating genes in single and social genes

2\_dicty\_tstarting\_codons.py – Calculate T-starting codon frequencies at each 3’ UTR codon position

2\_dicty\_ASC\_comparison.py – Calculate ASC frequencies in single and social genes

3\_thaliana\_AC\_null.py – Comparison of A. thaliana pollen-selective and pollen-depleted genes to A+C null

3\_thaliana\_sims\_null.py – Comparison of A. thaliana pollen-selective and pollen-depleted genes to sims

3\_thaliana\_stop\_usage.py – Count the number of TAA, TGA, TAG-terminating genes in the two gene groups

3\_thaliana\_tstarting\_codons.py – Calculate T-starting codon frequencies at each 3’ UTR codon position

3\_thaliana\_ASC\_comparison.py – Calculate ASC frequencies in pollen-specific and pollen-depleted genes

3\_thaliana\_FC\_analysis.py – Fold-change analysis for A. thaliana

3\_thaliana\_fourth\_t.py – Calculate +4T frequencies for TAA, TGA, TAG-terminating genes

3\_thaliana\_primary\_FC.py – Calculate stop codon usage for genes above and below a F/C threshold

3\_thaliana\_ASC\_fourth\_t\_proportions.py – Calculate proportion of +4T-genes that also have a proximal ASC

4\_aspergillus\_stop\_usage.py – Count usage of each stop in Aspergillus genes

4\_aspergillus\_FC\_analysis.py – Fold-change analysis for Aspergillus species

4\_aspergillus\_ASC\_comparison.py – Calculate ASC frequencies in CAGs and non-CAGs

4\_aspergillus\_niger\_AC\_null.py – Comparison of CAGs and non-CAGs to A+C null

4\_aspergillus\_niger\_sims\_null.py – Comparison of CAGs and non-CAGs to simulations

4\_aspergillus\_oryzae\_AC\_null.py – Comparison of CAGs and non-CAGs to A+C null

4\_aspergillus\_oryzae\_sims\_null.py – Comparison of CAGs and non-CAGs to simulations

5\_enrichment\_scores.py – Calculate enrichment scores (mean of positional enrichment)

5\_variables.py – Calculate median IGR, median gene length, and GC content for genomes in model

5\_high\_scores.py – Calculate high scores (highest positional enrichment)

6\_+4C.py – Compare C-starting codon frequency at position +1 to downstream

6\_+4T.py – Compare T-starting codon frequency at position +1 to downstream

7\_Expression\_ASCs.py – Calculate ASC and +4T frequencies for HEGs and LEGs

7\_Expression\_stop\_preference.py – Count usage of each stop codon within HEGs and LEGs for each genome

7\_Expression\_pooled.py – Count +4T and ASC frequencies for pooled HEGs and LEGs genes

7\_Expression\_whole\_UTR.py – Calculate ASC frequencies for whole UTR and by position for HEGs and LEGs

7\_Expression\_write.py – Identify HEGs and LEGs and write to two new FASTA files