Brief overview of various code (exec.py, cnn.py, rnn,py, slide.py)

exec.py – runs multiple simulations of simulation.py

Argument	Flag	Description
--rep_rate	-r	Birth rate for asexual, non-recombining population.
--death_rate	-x	Death rate for asexual, non-recombining population.
--fitness	-w	The fitness of individuals carrying positively selected mutation. It scales the death rate by 1/fitness.
--mutRate	-u	Mutation rate
--probBen	-p	Probability of a mutation being beneficial.
--initSize	-i	Initial size of simulated population.
--genomeSize	-gs	Genome, or haplotype block, size in base pairs.
--gens	-g	Number of generations of evolution.
--maxPopSize	-ms	Maximum population size to terminate simulation.
--imageSize	-ims	Number of aligned haplotypes per image.
--numberSims	-n	Number of simulated haplotypes to be generated.
--output	-out	Path to output directory. Must end with a backslash e.g. /path/
--max_mutations	-max	Maximum number of mutations in a given aligned haplotype block.
--min_mutations	-min	Minimum number of mutations in a given aligned haplotype block.

Output description	Saves *n* aligned haplotype blocks in numpy array format with each image having a dimension defined by (number of aligned haplotypes, genome size). The number of aligned haplotypes is defined by the --imageSize argument and the genome size is defined by the --genomeSize argument. The final size of each numpy array is therefore (aligned haplotypes X genome size).

Example Usage	`python3 exec.py -r 2.02 -w 1 -x 1 -p 0 -u 1e-2 -i 110 -gs 1000 -g 250 -ms 1e5 -n 2000 -out /neutral/`

cnn.py – trains a binary CNN on the aligned haplotype data

Argument	**Flag**	**Description**
--positive	-pos	Path to directory storing positive simulated data.
--neutral	-neu	Path to directory storing neutral simulated data.
--prop	-p	Percentage of data assigned to test/validation set.
--out	-o	Path to output directory. Must end with a backslash e.g. /path/
--num	-n	Number of training samples from each evolutionary class. Total training size is therefore 2n.
--fitness	-w	Fitness of positive simulations. Optional parameter used to maintain standardized file labels.

Output description	Trains a binary CNN on the aligned haplotype data. The CNN is setup for 200 aligned haplotypes over 1000 base pairs. The input shape can be modified within the trainCNN function in model.py. Following training, two files are stored in the output directory: the CNN model in tensorflow (tf) format and the training and validation accuracy for each epoch in csv format.

Example Usage	`python3 cnn.py --positive /1e3_positive/ --neutral /1e3_neutral/ -p 0.1 --out /models_final/ --num 2000`

rnn.py – trains an LSTM on sliding window estimates from CNN sliding window analysis

Argument	**Flag**	**Description**
NA	NA	NA

Description	To train the RNN, you need to build a set of features (mean CNN estimates) and corresponding set of targets (1 if sliding window contains beneficial mutation or 0 if sliding window does not contain beneficial mutation). You can generate this information by running genome.py under different evolutionary scenarios and parsing the *_loci and *_pred files.

slide.py – scans aligned haplotypes for signatures of selection

Argument	**Flag**	**Description**
--genome_length	-gl	Size of genome or region to slide across.
--step_size	-ss	Sliding window step size.
--alignment_size	-as	Length of haplotype alignments CNN was trained on.
--buffer_size	-bs	Add a starting buffer (e.g. if early regions prone to sequencing error).
--number_subamples	-ns	Number of population subsamples to perform across each sliding window. Subsamples are used to build confidence intervals.
--group	-g	The name of the column to filter in geography data frame.
--samples	-s	The value to filter within the group column.
--months	-mon	The specific month to evaluate. Default is 0 which means use all data.
--geography	-geo	Path to data frame that contains meta information on samples.
--cnn_model	-cnn	The path to trained tensorflow CNN.
--rnn_model	-rnn	The path to trained tensorflow LSTM.
--haplotype_directory	-d	The directory containing the empirical binary encoded haplotypes. The haplotypes must be binary encoded (e.g. 01000100) with 0 meaning base is identical to reference and 1 meaning base is different.
--output	-out	Path to output directory. Must end with a backslash e.g. /path/
--backwards	-back	Runs the sliding window in the opposite direction. Combining forward and backward slides helps refine predictions.

Output description	Scans aligned haplotypes for signatures of selection and subsamples from the population to build confidence intervals. The output is a data frame with 5 columns. The columns index1 and index2 indicate the start and end positions for a given sliding window. Upper and lower represent the upper and lower bounds of the subsampled confidence. Mean is the mean probability across subsamples.

Example Usage

```
python3 slide.py -gl 29903 -ss 5 -as 1000 -bs 50 -ns 10 -g country -s
Canada -mon 0 -geo covid_ids.csv -m cnn.tf -d /haplotypes/ -out
/sliding_output/
```

Additional description of geography data frame (meta information) for slide.py:
The geography data frame is used to sample data from the directory containing the empirical haplotypes. The --group argument specifies which column to use when filtering haplotypes. The --samples argument specifies what values to filter from the --group column. For example, --group ‘country’ --samples ‘Canada’ indicates that filenames will be sampled from data in the ‘country’ column with the value ‘Canada’. All samples remaining in filtered data frame will be subsampled from the haplotype directory by their corresponding filename in the column ‘filename’.

filename	country	id	date	continent	region	month
Algeria+EPI_ISL_418241.txt	Algeria	EPI_ISL_418241	2020-03-02	Africa	Middle East & North Africa	3
Canada+EPI_ISL_418334.txt	Canada	EPI_ISL_418334	2020-03-07	Americas	North America	3