Contents

GeoGenIE User Manual

Introduction

GeoGenIE (Geographic-Genetic Inference Engine) is a comprehensive software tool designed to predict geographic coordinates (longitude and latitude) from genetic SNP data. GeoGenIE utilizes deep learning models and offers several advanced features such as outlier detection to remove translocated individuals, handling of imbalanced sampling using a custom oversampling algorithm adapted from SMOTE, and weighting the loss function by inverse sampling densities. The software is user-friendly and provides extensive visualizations and metrics for evaluating predictions, making it a robust and accurate solution for geographic-genetic inference.

Installation

To install GeoGenIE, it is recommended to use a virtual environment or conda environment. From the root project directory, enter the following command:

```
pip install .
```

GeoGenIE will be hosted on PyPI in the future for easier installation.

Figure 1: GeoGenIE Logo

Dependencies

The following packages will be installed when running `pip install .`:

- python >= 3.11
- geopandas
- geopy
- imblearn
- jenkspy
- kaleido
- kneed
- matplotlib
- numba
- numpy
- optuna
- pandas
- pickle
- plotly
- pynndescent
- pysam
- requests
- scikit-learn
- scipy
- seaborn
- statsmodels
- torch
- xgboost
- pyyaml

Usage

Running GeoGenIE

GeoGenIE can be run with individual command-line arguments, but using a YAML config file is recommended. See `config_files/config.yaml` for an example YAML file. Assuming GeoGenIE is installed in your environment, you can run it like this:

```
geogenie --config config_files/config.yaml
```

Command-line Options

You can see all the command-line options by running the help flag:

```
geogenie -h
```

Note: If you don't want to use the configuration file, you can specify each argument individually on the command line. For example:

```
geogenie --output_dir "my_analysis" --prefix "my_prefix" <other arguments>
```

Configuration File

You can set all the options for input files, model parameters, etc., in the `config_files/config.yaml` file. Using a configuration file allows tracking of parameters across multiple runs and ensures better reproducibility.

- Python `None` values are represented by `null` (without quotes).
- Python `True` values are represented by `true` (all lowercase, no quotes).
- Python `False` values are represented by `false` (all lowercase, no quotes).

You can also leave comments with `# my_comment`. The arguments can be in any order in the `config_files/config.yaml` file.

Running the Software

```
geogenie --config config_files/config.yaml
```

Required Input Files

- VCF file
  - Specified with `--vcf <path/to/vcf_file.vcf.gz>`.
- GTSeq file
  - Specified with `--gtseq <path/to/gtseq_file.csv>`.
- Coordinates file
  - Specified with `--sample_data <path/to/coordinates_file.tsv>`.
  - Tab-delimited file with three columns:
    - "sampleID"
    - "x" (longitude)
    - "y" (latitude)
  - Coordinates ("x" and "y") should be in decimal degree format.
  - Unknown coordinates for which predictions should be made should contain the string "nan" in the "x" and "y" columns.
- Known Coordinates file
  - Specified with `--known_sample_data <path/to/coordinates_file.tsv>`.
  - Tab-delimited file with three columns:
    - "sampleID"
    - "x" (longitude)
    - "y" (latitude)
  - Known coordinates file provides recorded localities for per-sample bootstrapped output plots.

GeoGenIE Features and Settings

Data Input and Preprocessing

GeoGenIE supports various options for data input and preprocessing:

- `--min_mac`: Minimum minor allele count (MAC) to retain SNPs. Default: 2.
  - **Tip**: Increasing the MAC can help to filter out rare variants and reduce noise in the data.
- `--max_SNPs`: Maximum number of SNPs to randomly subset. Default: None (Use all SNPs).
  - **Tip**: Use this option to reduce computational load by limiting the number of SNPs used in the analysis.

Model Configuration

Configure the model with the following options:

- `--nlayers`: Number of hidden layers to include in the neural network. Default: 10.
  - **Tip**: Increase the number of layers for more complex models but beware of overfitting.
- `--width`: Number of neurons per layer. Default: 256.
  - **Tip**: Higher width values allow the model to learn more complex features but increase computation time.
- `--dropout_prop`: Dropout rate to prevent overfitting. Default: 0.25.
  - **Tip**: Adjust this rate to balance model complexity and generalization.
- `--criterion`: Model loss criterion. Options: 'rmse', 'huber', 'drms'. Default: 'rmse'.
  - **Tip**: Use 'huber' for a combination of mean squared error and mean absolute error to handle outliers.
- `--load_best_params`: Load best parameters from previous Optuna search. Default: None.
  - **Tip**: Use this option to save time by reusing optimized parameters.
- `--use_gradient_boosting`: Use Gradient Boosting model instead of deep learning model. Default: False.
  - **Tip**: Gradient Boosting can be effective for smaller datasets or when deep learning models overfit.
- `--dtype`: PyTorch data type. Options: 'float32', 'float64'. Default: 'float32'.
  - **Tip**: Use 'float64' for higher precision at the cost of increased memory usage.

Training Parameters

Define training parameters:

- `--batch_size`: Training batch size. Default: 32.
  - **Tip**: Larger batch sizes can lead to more stable training but require more memory.
- `--max_epochs`: Maximum training epochs. Default: 5000.
  - **Tip**: Set this high and rely on early stopping to prevent overfitting.

- `--learning_rate`: Learning rate for optimizer. Default: 0.001.
  - **Tip**: Adjust this parameter to control the speed of learning; lower values provide finer adjustments.
- `--l2_reg`: L2 regularization weight. Default: 0.0.
  - **Tip**: Use L2 regularization to penalize large weights and reduce overfitting.
- `--early_stop_patience`: Epochs to wait after no improvement before stopping. Default: 48.
  - **Tip**: Decrease this value to stop training earlier if no improvement is observed.
- `--train_split`: Training data proportion. Default: 0.8.
  - **Tip**: Ensure the sum of `--train_split` and `--val_split` is 1.0.
- `--val_split`: Validation data proportion. Default: 0.2.
  - **Tip**: Use a larger validation split for more robust model evaluation.
- `--do_bootstrap`: Enable bootstrap replicates. Default: False.
  - **Tip**: Enable bootstrapping to estimate confidence intervals for predictions.
- `--nboots`: Number of bootstrap replicates. Default: 100.
  - **Tip**: Increase this value for more accurate confidence intervals.
- `--feature_prop`: Proportion of features for bootstrapping. Default: 0.8.
  - **Tip**: Adjust to include a larger or smaller subset of features in each bootstrap replicate.
- `--important_feature_prop`: Proportion of most important features to ensure are used for all bootstrap replicates. Default: 0.2.

Tip**: Ensure key features are always included in bootstrapping. - `--n_importance_estimators`: Number of estimators for feature importance. Default: 100. -** Tip**: Increase for more accurate feature importance estimates. - `--do_gridsearch`: Perform Optuna parameter search. Default: False. -** Tip**: Enable to optimize model parameters for better performance. - `--n_iter`: Iterations for parameter optimization. Default: 100. -** Tip**: Increase for more thorough parameter searches. - `--lr_scheduler_patience`: Learning rate scheduler patience. Default: 16. -** Tip**: Reduce learning rate when no improvement is seen for specified epochs. - `--lr_scheduler_factor`: Factor to reduce learning rate. Default: 0.5. -** Tip**: Lower values make finer adjustments to the learning rate. - `--factor`: Scale factor for neural network widths. Default: 1.0. -** Tip**: Adjust to control the width reduction of hidden layers. - `--grad_clip`: Enable gradient clipping. Default: False. -** Tip**: Use to prevent gradient explosion in deep networks.

Geographic Density Sampler

Configure the geographic density sampler:

- `--use_weighted`: Use inverse-weighted probability sampling. Options: ‘loss’, ‘none’. Default: ‘none’.

  - **Tip**: Apply weighting to address imbalanced sampling densities.
- `--oversample_method`: Synthetic oversampling/undersampling method. Options: 'kmeans', 'none'. Default: 'none'.
  - **Tip**: Use 'kmeans' to create synthetic samples for underrepresented regions.
- `--oversample_neighbors`: Number of nearest neighbors for oversampling. Default: 5.
  - **Tip**: Adjust to control the number of synthetic samples generated.
- `--n_bins`: Number of bins for synthetic resampling. Default: 8.
  - **Tip**: Increase for finer-grained sampling density adjustments.
- `--use_kmeans`: Use KMeans clustering in the sampler. Default: True.
  - **Tip**: Essential for generating synthetic samples based on clusters.
- `--w_power`: Exponential power for inverse density weighting. Default: 1.0.
  - **Tip**: Adjust for more aggressive weighting of sampling density.
- `--max_clusters`: Maximum number of clusters for KMeans. Default: 10.
  - **Tip**: Increase for more detailed clustering.
- `--focus_regions`: Geographic regions of interest for sampling density weights.
  - **Tip**: Use to prioritize specific geographic areas in sampling. Should be a list of tuples with focus region bounding areas (min_lon, min_lat, max_lon, max_lat). Multiple regions can be specified in the list.
- `--normalize_sample_weights`: Normalize sample weights between 0 and 1. Default: False.
  - **Tip**: Ensure all sample weights are on a comparable scale.

Outlier Detection

GeoGenIE can remove outliers flagged as distant from nearby samples in spatial and genetic contexts:

- `--detect_outliers`: Enable outlier detection. Default: False.
  - **Tip**: Enable to remove samples that deviate significantly from expected geographic and/ or genetic patterns.
- `--min_nn_dist`: Minimum distance between nearest neighbors for outlier detection. Default: 1000 meters.
  - **Tip**: Increase to detect only very distant outliers.
- `--scale_factor`: Scale factor for geographic distance. Default: 100.
  - **Tip**: Adjust to control the scaling of distances in outlier detection.
- `--significance_level`: Significance level for p-values to determine outliers. Default: 0.05.
  - **Tip**: Lower values increase the stringency of outlier detection.
- `--maxk`: Maximum number of nearest neighbors for outlier detection. Default: 50.
  - **Tip**: Adjust to control the scope of neighbor comparisons.
  - **CAUTION**: Raising too high can increase computation time and resources.

Bootstrapping for Error Estimates

To obtain confidence intervals for locality predictions, enable bootstrapping with the `--do_bootstrap` boolean option. Bootstrapping is parallelized, and you can set the number of CPU threads with `--n_jobs <n_cpus>` or `--n_jobs -1` to use all available CPU threads.

Using bootstrapping generates additional plots showing confidence intervals for each sample, saved in `<output_dir>/plots/bootstrapped_sample_ci/<prefix>_*.<filetype>`.

The file type for output plots can be specified with `--filetype "pdf"`, `--filetype "png"`, or `--filetype "jpg"`. The number of bootstrap replicates can be changed with `--nboots <integer>`.

Output and Miscellaneous

Configure output and other miscellaneous settings:

- `--prefix`: Output file prefix. Default: 'output'.
  - **Tip**: Use meaningful prefixes to organize output files from different GeoGenIE runs.
- `--sqldb`: SQLite3 database directory for Optuna optimization. Default: None.
  - **Tip**: Specify to save optimization results for future use and resuming Optuna optimization.
- `--output_dir`: Directory to store output files. Default: './output'.
  - **Tip**: Organize outputs by specifying a unique directory and/ or unique prefix for each run.
- `--seed`: Random seed for reproducibility. Default: None.
  - **Tip**: Set a seed to ensure reproducible results.
- `--gpu_number`: GPU number for computation. Default: None (use CPU only).
  - **Tip**: Use a GPU to speed up training for large datasets.
- `--n_jobs`: Number of CPU jobs to use. Default: -1 (use all CPUs).
  - **Tip**: Adjust to limit CPU usage if running multiple processes.
- `--verbose`: Verbosity level for logging. Default: 1.
  - **Tip**: Increase to 2 get more detailed logging information, or decrease to 0 to reduce logging information.

Embedding Settings

GeoGenIE offers several embedding options for input features:

- `--embedding_type`: Embedding to use with input SNP dataset. Supported options: 'pca', 'kernelpca', 'nmf', 'lle', 'mca', 'mds', 'polynomial', 'tsne', and 'none'. Default: 'none' (no embedding).
  - **Tip**: Use 'pca' for principal component analysis to reduce dimensionality, or 'tsne' for visualizing high-dimensional data.

- `--n_components`: Number of components for ‘pca’ or ‘tsne’ embeddings. Default: Search for optimal components.
  - **Tip**: Specifying a higher number of components can capture more variance but may also increase computational complexity.
- `--embedding_sensitivity`: Sensitivity setting for selecting the optimal number of components with ‘mca’ and ‘pca’. Default: 1.0.
  - **Tip**: Adjust this parameter to fine-tune the balance between overfitting and underfitting.
- `--tsne_perplexity`: Perplexity setting for T-SNE embedding. Default: 30.
  - **Tip**: Lower perplexity values can capture finer details but may lead to more noise.
- `--polynomial_degree`: Polynomial degree for ‘polynomial’ embedding. Default: 2.
  - **Tip**: Higher degrees add complexity and computational overhead; use with caution.
- `--n_init`: Number of initialization runs for Multi-Dimensional Scaling embedding. Default: 4.
  - **Tip**: More initialization runs can provide more stable results but increase computation time.

Plot Settings

Set plotting parameters to customize the visualizations:

- `--show_plots`: Show in-line plots. Default: False.
  - **Tip**: Enable for interactive environments like Jupyter notebooks.
- `--fontsize`: Font size for plot labels, ticks, and titles. Default: 24.
  - **Tip**: Adjust to improve readability of plots.
- `--filetype`: File type for saving plots. Default: ‘png’.
  - **Tip**: Use ‘pdf’ for high-quality vector graphics, or ‘png’ or ‘jpg’ for an image format.
- `--plot_dpi`: DPI for image format plots. Default: 300.
  - **Tip**: Higher DPI produces clearer images but increases file size.
- `--remove_splines`: Remove bottom and left axis splines from map plots. Default: False.
  - **Tip**: Use for cleaner map visualizations.
- `--shapefile`: URL or file path for shapefile used in plotting. Default: Continental USA County Lines basemap.
  - **Tip**: Specify a custom shapefile for regions outside the USA.
- `--basemap_fips`: FIPS code for basemap. Default: None.
  - **Tip**: Use to focus plots on specific states or regions. E.g., the Arkansas FIPS code is “05”.
- `--highlight_basemap_counties`: Highlight specified counties on the base map in gray. Default: None.
  - **Tip**: Highlight areas of interest for better visualization.

- `--samples_to_plot`: Comma-separated sample IDs to plot when using bootstrapping. Default: None.
  - **Tip**: Plot specific samples to focus on particular cases.
- `--n_contour_levels`: Number of contour levels for the Kriging plot. Default: 20.
  - **Tip**: Adjust for more detailed or simplified contour plots.
- `--min_colorscale`: Minimum colorbar value for the Kriging plot. Default: 0.
  - **Tip**: Set to match the range of your data.
- `--max_colorscale`: Maximum colorbar value for the Kriging plot. Default: 300.
  - **Tip**: Increase if your data range exceeds the default.
- `--sample_point_scale`: Scale factor for sample point size on Kriging plot. Default: 2.
  - **Tip**: Adjust to ensure points are visible but not overwhelming.
- `--bbox_buffer`: Buffer for the sampling bounding box on map visualizations. Default: 0.1.
  - **Tip**: Increase to add more context around the sampling area.

Output Files and File Structure

Outputs are saved to the directory specified by `--output_dir <my_output_dir>/<prefix_>_*`. The prefix is specified with `--prefix <prefix>`. The directory structure of `<output_dir>` includes:

- `benchmarking`: Execution times for model training and prediction, with one line per bootstrap replicate if using bootstrapping.
- `bootstrapped_sample_ci`: One plot per sample showing confidence intervals on a map.
- `bootstrap_metrics`: JSON files with statistical metrics for each bootstrap replicate, in `test` and `val` subdirectories.
- `bootstrap_predictions`: CSV files containing predictions for each bootstrap replicate, in `test`, `val`, and `unknown` subdirectories.
- `bootstrap_summaries`: Mean, median, and standard deviation representations of bootstrap replicates for the test, val, and unknown ("pred") datasets.
- `data`: Text files with sample IDs detected as outliers if `--detect_outliers` is enabled.
- `logfiles`: Logs with INFO, WARNING, and ERROR messages, including timestamps and GeoGenIE modules.
- `models`: Trained PyTorch models saved as ".pt" files, one per bootstrap if `--do_bootstrap` is enabled.
- `optimize`: Optuna results, including the best-found parameters as a JSON file.
- `plots`: All plots and visualizations, including model prediction error visualizations and the basemap shapefile specified with `--shapefile <url>`.

Per-sample plots

visualizing bootstrapped prediction error are saved in the `plots/bootstrapped_sample_ci` subdirectory.

Warning: Re-running GeoGenIE with the same `output_dir` and `prefix` will overwrite all outputs except the Optuna SQL database.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. In Proceedings of the 25th International Conference on Knowledge Discovery and Data Mining.
2. Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research, 18(17), 1-5. http://jmlr.org/papers/v18/16-365
3. Chang et al., (2023). GGoutlieR: an R package to identify and visualize unusual geo-genetic patterns of biological samples. Journal of Open Source Software, 8(91), 5687. https://doi.org/10.21105/joss.05687 “‘