Project: GeneFlow

Prerequisites

1. Python 3.8 or above.
2. Required Python libraries:
    a. Flask
    b. pandas
    c. numpy
    d. seaborn
    e. matplotlib
    f. sklearn
    g. joblib
    h. Io

3. Datasets:
    a. enhanced_aquatic_population_data.csv
    b. environmental_dataset.csv

Installation

1. Clone the repository.
2. Navigate to the project directory and install dependencies:

pip install -r requirements.txt

3. Ensure the datasets are placed in the paths specified in the app.py and training.py scripts.

Usage

Running the Application

1. Train the models (if not pre-trained):

python training.py

2. Start the Flask server:

python app.py

3. The server will be available at http://127.0.0.1:5000.

API Endpoints

1. Predict Trait Frequency

- **Endpoint**: /predict_trait_frequency
- **Method**: POST
- **Description**: Predicts the frequency of Trait A.
- **Input JSON Example**:

```
{
  "Population_Size": 1000,
  "Migration_Rate": 0.05,
  "Temperature_C": 25.0,
  "Selection_Pressure": 0.8
}
```

- **Response Example**:

```
{
  "Predicted_Trait_Frequency_A": 0.56
}
```

2. Predict Selection Pressure

- **Endpoint**: /predict_selection_pressure
- **Method**: POST
- **Description**: Predicts the selection pressure.
- **Input JSON Example**:

```
{
  "Temperature_C": 25.0,
  "pH_Level": 7.5,
  "Environment": 1,
  "Dissolved_Oxygen": 8.2
}
```

- **Response Example**:

```
{
  "Predicted_Selection_Pressure": 0.85
}
```

3. Predict Trait Frequency B

- **Endpoint**: /predict_trait_frequency_b
- **Method**: POST
- **Description**: Predicts the frequency of Trait B.
- **Input JSON Example**:

```
{
  "Population_Size": 1000,
  "Migration_Rate": 0.05,
  "Temperature_C": 25.0,
  "Selection_Pressure": 0.8
}
```

- **Response Example**:

```
{
  "Predicted_Trait_Frequency_B": 0.65
}
```

4. Simulate Scenario

- **Endpoint**: /simulate_scenario
- **Method**: POST
- **Description**: Simulates trait frequency changes under different scenarios.

- **Input JSON Example**:

```
{
  "variable": "Temperature_C",
  "values": [20.0, 25.0, 30.0],
  "fixed_features": {
    "Population_Size": 1000,
    "Migration_Rate": 0.05,
    "Selection_Pressure": 0.8
  }
}
```

- **Response Example**:

```
[
  {
    "Population_Size": 1000,
    "Migration_Rate": 0.05,
    "Selection_Pressure": 0.8,
    "Temperature_C": 20.0,
    "Predicted_Trait_Frequency_A": 0.50
  },
  {
    "Population_Size": 1000,
    "Migration_Rate": 0.05,
    "Selection_Pressure": 0.8,
    "Temperature_C": 25.0,
    "Predicted_Trait_Frequency_A": 0.55
  },
  {
    "Population_Size": 1000,
    "Migration_Rate": 0.05,
    "Selection_Pressure": 0.8,
    "Temperature_C": 30.0,
    "Predicted_Trait_Frequency_A": 0.60
  }
]
```


5. Recommend Breeding Program

- **Endpoint**: /recommend_breeding_program
- **Method**: GET
- **Description**: Recommends optimal conditions for population fitness.
- **Response Example**:


```
[
  {
    "Population_ID": 1,
    "Temperature_C": 25.0,
    "pH_Level": 7.5,
    "Fitness_A": 0.90,
    "Fitness_B": 0.85
  }
]
```


6. Heatmap Trait Distribution

- **Endpoint**: /heatmap_trait_distribution

- **Method**: GET
- **Description**: Returns heatmaps for trait frequency distribution.
- **Response**: PNG file download.

7. Lineplot Simulation

- **Endpoint**: /lineplot_simulation
- **Method**: GET
- **Description**: Returns line plots of simulated trait frequencies.
- **Response**: PNG file download.

8. Barplot Population Fitness

- **Endpoint**: /barplot_population_fitness
- **Method**: GET
- **Description**: Returns bar plots comparing fitness levels.
- **Response**: PNG file download.