BI4

October 8, 2024

[1]:
```
!pip install genomic-benchmarks
```

```
Collecting genomic-benchmarks
  Downloading genomic_benchmarks-0.0.9.tar.gz (21 kB)
  Preparing metadata (setup.py) … done
Collecting biopython>=1.79 (from genomic-benchmarks)
  Downloading
biopython-1.84-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(12 kB)
Requirement already satisfied: requests>=2.23.0 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from genomic-
benchmarks) (2.32.3)
Requirement already satisfied: pip>=20.0.1 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from genomic-
benchmarks) (24.2)
Requirement already satisfied: numpy>=1.17.0 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from genomic-
benchmarks) (1.23.5)
Requirement already satisfied: pandas>=1.1.4 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from genomic-
benchmarks) (2.2.2)
Collecting tqdm>=4.41.1 (from genomic-benchmarks)
  Using cached tqdm-4.66.5-py3-none-any.whl.metadata (57 kB)
Requirement already satisfied: pyyaml>=5.3.1 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from genomic-
benchmarks) (6.0.1)
Collecting gdown>=4.2.0 (from genomic-benchmarks)
  Downloading gdown-5.2.0-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: yarl in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from genomic-
benchmarks) (1.11.0)
Requirement already satisfied: beautifulsoup4 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from
gdown>=4.2.0->genomic-benchmarks) (4.12.3)
Collecting filelock (from gdown>=4.2.0->genomic-benchmarks)
  Using cached filelock-3.16.1-py3-none-any.whl.metadata (2.9 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from
```

```
pandas>=1.1.4->genomic-benchmarks) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from
pandas>=1.1.4->genomic-benchmarks) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from
pandas>=1.1.4->genomic-benchmarks) (2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from
requests>=2.23.0->genomic-benchmarks) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from
requests>=2.23.0->genomic-benchmarks) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from
requests>=2.23.0->genomic-benchmarks) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from
requests>=2.23.0->genomic-benchmarks) (2024.8.30)
Requirement already satisfied: multidict>=4.0 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from yarl->genomic-
benchmarks) (6.0.4)
Requirement already satisfied: six>=1.5 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from python-
dateutil>=2.8.2->pandas>=1.1.4->genomic-benchmarks) (1.16.0)
Requirement already satisfied: soupsieve>1.2 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from
beautifulsoup4->gdown>=4.2.0->genomic-benchmarks) (2.5)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-packages (from
requests[socks]->gdown>=4.2.0->genomic-benchmarks) (1.7.1)
Downloading
biopython-1.84-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.2 MB)

3.2/3.2 MB 140.2 kB/s eta 0:00:00 kB/s eta
0:00:03:04
Downloading gdown-5.2.0-py3-none-any.whl (18 kB)
Downloading tqdm-4.66.5-py3-none-any.whl (78 kB)
Downloading filelock-3.16.1-py3-none-any.whl (16 kB)
Building wheels for collected packages: genomic-benchmarks
  Building wheel for genomic-benchmarks (setup.py) … done
  Created wheel for genomic-benchmarks:
filename=genomic_benchmarks-0.0.9-py3-none-any.whl size=22506
sha256=6333931349711fe3d2216360a7a75d1951a3ce0749e6c4a93e59a14a10e16246
  Stored in directory: /home/aryan/.cache/pip/wheels/f8/72/7e/96228b1cf2d3d1b1f8
31a351712d86316c61c511e25e471120
Successfully built genomic-benchmarks
Installing collected packages: tqdm, filelock, biopython, gdown, genomic-
```

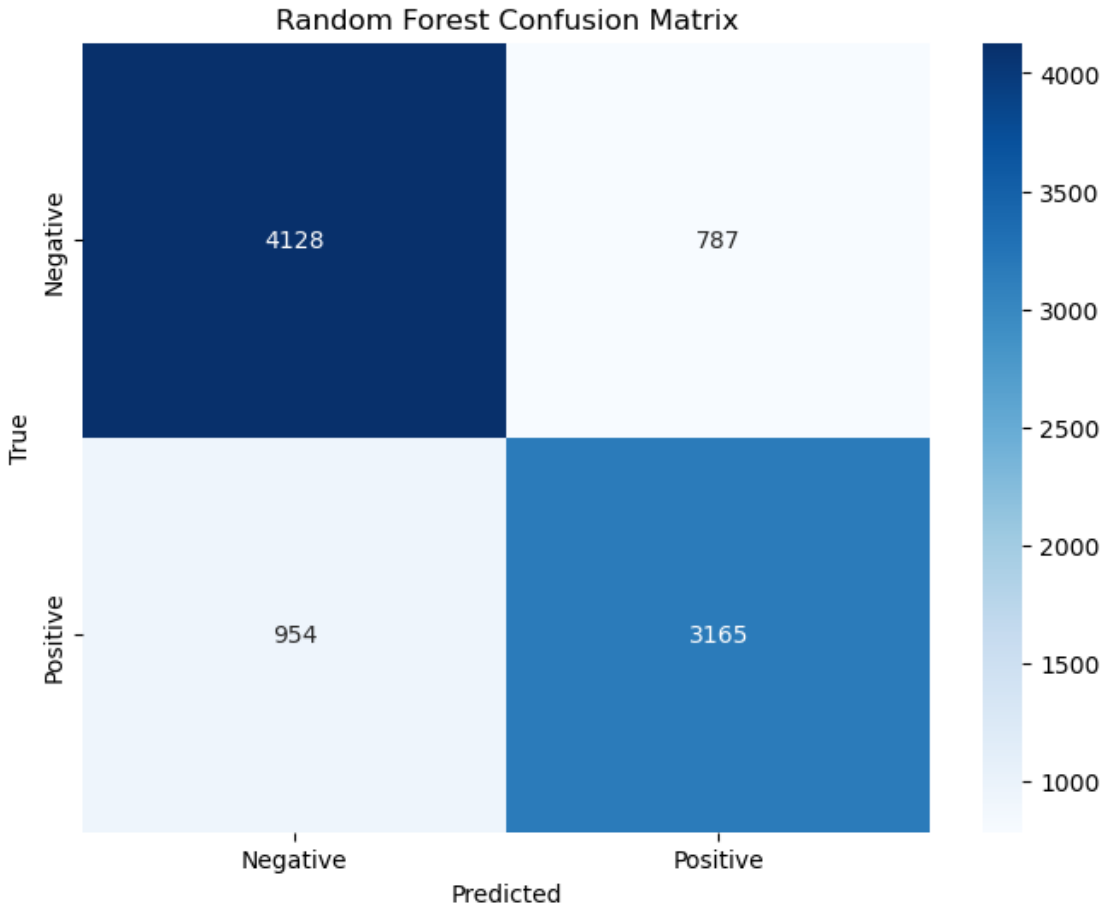
```
benchmarks
Successfully installed biopython-1.84 filelock-3.16.1 gdown-5.2.0 genomic-
benchmarks-0.0.9 tqdm-4.66.5
```

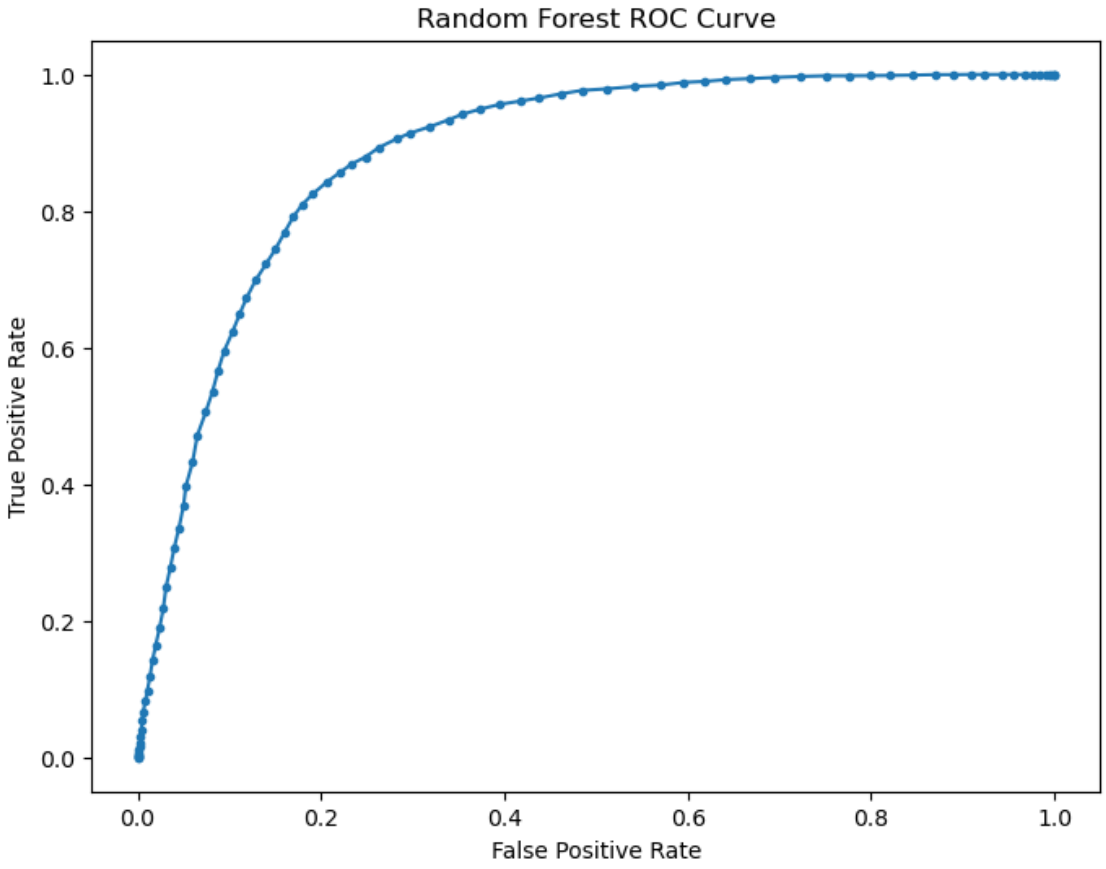
[1]:
```python
from genomic_benchmarks.data_check import list_datasets

list_datasets()

# Display information about the "human_nontata_promoters" dataset with version 0
from genomic_benchmarks.data_check import info

info("human_nontata_promoters", version=0)

# Load the "human_nontata_promoters" dataset for training and testing
from genomic_benchmarks.dataset_getters.pytorch_datasets import␣
 ↪HumanNontataPromoters

train = HumanNontataPromoters(split='train', version=0)
test = HumanNontataPromoters(split='test', version=0)

# Access a specific example from the training dataset (e.g., the 3000th sample)
train[3000]

import numpy as np

# Define a mapping of DNA bases to one-hot encoding
base_to_index = {'A': 0, 'C': 1, 'G': 2, 'T': 3}

# Function to one-hot encode a DNA sequence, treating 'N' as missing data
def one_hot_encode(sequence, seq_length):
    encoded_sequence = np.zeros((seq_length, 4), dtype=int)
    for i, base in enumerate(sequence):
        if base in base_to_index:
            # Set the corresponding index to 1 for valid bases (A, C, G, T)
            encoded_sequence[i, base_to_index[base]] = 1
        else:
            # Treat 'N' as missing data (all zeros)
            encoded_sequence[i, :] = 0
    return encoded_sequence

# Apply one-hot encoding to the entire training and testing datasets
train_encoded = [one_hot_encode(item[0], len(item[0])) for item in train]
test_encoded = [one_hot_encode(item[0], len(item[0])) for item in test]

# Access the one-hot encoded sequence of the first sample in the training␣
 ↪dataset
train_encoded[0]
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report

# Extract labels from the training and testing datasets
train_labels = [item[1] for item in train]
test_labels = [item[1] for item in test]

# Reshape the one-hot encoded sequences into a two-dimensional format
train_encoded = np.array(train_encoded).reshape(len(train_encoded), -1)
test_encoded = np.array(test_encoded).reshape(len(test_encoded), -1)

# 2. Choose an Algorithm (Random Forest)
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)

# 3. Training the Model
rf_classifier.fit(train_encoded, train_labels)

# 4. Model Evaluation
predictions = rf_classifier.predict(test_encoded)
accuracy = accuracy_score(test_labels, predictions)
report = classification_report(test_labels, predictions)

print(f"Accuracy: {accuracy}")
print("Classification Report:\n", report)

import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
import seaborn as sns

# Create a confusion matrix for the Random Forest classifier
rf_cm = confusion_matrix(test_labels, predictions)

# Plot the confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(rf_cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Negative',␣
 ↪'Positive'], yticklabels=['Negative', 'Positive'])
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Random Forest Confusion Matrix')
plt.show()


from sklearn.metrics import roc_curve, roc_auc_score

# Calculate ROC curve for Random Forest
rf_probs = rf_classifier.predict_proba(test_encoded)[:, 1]
```

```
rf_fpr, rf_tpr, _ = roc_curve(test_labels, rf_probs)

# Plot ROC curve
plt.figure(figsize=(8, 6))
plt.plot(rf_fpr, rf_tpr, marker='.')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Random Forest ROC Curve')
plt.show()
```

```
/home/aryan/anaconda3/envs/labs/lib/python3.9/site-
packages/genomic_benchmarks/utils/datasets.py:11: TqdmExperimentalWarning: Using
`tqdm.autonotebook.tqdm` in notebook mode. Use `tqdm.tqdm` instead to force
console mode (e.g. in jupyter console)
  from tqdm.autonotebook import tqdm

Dataset `human_nontata_promoters` has 2 classes: negative, positive.

All lengths of genomic intervals equals 251.

Totally 36131 sequences have been found, 27097 for training and 9034 for
testing.

Downloading…
From (original):
https://drive.google.com/uc?id=1VdUg0Zu8yfLS6QesBXwGz1PIQrTW3Ze4
From (redirected): https://drive.google.com/uc?id=1VdUg0Zu8yfLS6QesBXwGz1PIQrTW3
Ze4&confirm=t&uuid=a1b5a1a3-0521-40c8-8ebb-a0d216d5190d
To: /home/aryan/.genomic_benchmarks/human_nontata_promoters.zip
100%|                    | 11.8M/11.8M [00:08<00:00, 1.32MB/s]

Accuracy: 0.8072835953066194
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.84      0.83      4915
           1       0.80      0.77      0.78      4119

    accuracy                           0.81      9034
   macro avg       0.81      0.80      0.81      9034
weighted avg       0.81      0.81      0.81      9034
```


Random Forest Confusion Matrix
4128
787
954
3165
Negative
Positive
True
Negative
Positive
Predicted
4000
3500
3000
2500
2000
1500
1000


Random Forest ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

[ ]: