Melaku Garsamo

Generating Unique DNA Barcodes

This code snippet generates a list of unique DNA barcodes. A DNA barcode is a specific sequence of nucleotides that serves as a unique identifier for a biological entity. Here's a step-by-step explanation of the process:

1. **Generate All Possible DNA Sequences:** Initially, a list of all possible 10-base DNA sequences is created by combining the four DNA bases - "A" (Adenine), "C" (Cytosine), "G" (Guanine), and "T" (Thymine) in a nested loop.

2. **Shuffle the List:** The list of sequences is then shuffled to obtain a random order.

3. **Generate Unique Barcodes:** An empty set named 'barcodes' is initialized to store unique barcodes. The following steps are performed repeatedly until 10,000 unique barcodes are obtained:

   - A random sequence is selected from the list of shuffled sequences.
   - Three random indices are chosen from 0 to 9 to perform mutations.
   - At the selected indices, the nucleotides are changed to other bases, ensuring that each barcode differs by exactly three nucleotides from its original sequence.
   - The mutated barcode is added to the 'barcodes' set.
4. **Halfway Checkpoint:** After generating 5,000 unique barcodes, a comment "Halfway done!" is printed to indicate progress.

5. **Sort and Convert to DataFrame:** Once the set contains 10,000 unique barcodes, it is converted back to a list and sorted alphabetically. Finally, the sorted list is converted into a Pandas DataFrame with a single column named "Barcode" for further analysis.

This process ensures the creation of 10,000 unique DNA barcodes with minimal similarities between them, making them valuable for various biological applications, such as identifying individual samples or tracking specific genetic sequences.

In [ ]:

In [ ]:

In [ ]:

In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
%matplotlib inline
%config InlineBackend.figure_format='svg'
```

```
import os
import random
from scipy.stats import linregress
from scipy.optimize import curve_fit
import seaborn as sns
from matplotlib.patches import Rectangle
from palettable.scientific.sequential import *
import sys
if not sys.warnoptions:
    import warnings
    warnings.simplefilter("ignore")
import re
from typing import List, Optional, Union

from sigfig import round
from sklearn.metrics import r2_score

import hillfit
```

In [ ]:

In [8]:

```
# Generate a list of all possible 10-base DNA sequences
bases = ["A", "C", "G", "T"]
all_sequences = [a+b+c+d+e+f+g+h+i+j for a in bases for b in bases for c in bas

# Shuffle the list to get a random order
random.shuffle(all_sequences)

# Initialize an empty set to store the barcodes
barcodes = set()

# Generate barcodes until we have 10,000 unique barcodes
while len(barcodes) < 10000:
    # Pick a random sequence
    sequence = all_sequences.pop()

    # Generate 3 random indices to mutate
    indices = random.sample(range(10), 3)

    # Mutate the sequence at the selected indices
    barcode = list(sequence)
    for i in indices:
        bases_except_current = [b for b in bases if b != barcode[i]]
        barcode[i] = random.choice(bases_except_current)
    barcode = "".join(barcode)

    # Add the barcode to the set
    barcodes.add(barcode)

    # Print a comment after half of the barcodes have been generated
    if len(barcodes) == 5000:
        print("Halfway done!")

# Convert the set to a list and sort it alphabetically
barcodes = sorted(list(barcodes))

# Convert the list to a Pandas DataFrame
df_barcodes = pd.DataFrame(barcodes, columns=["Barcode"])
```

```
Halfway done!
```

In [ ]:

```

```

In [9]:

```
df_barcodes.head()
```

Out[9]:

	Barcode
0	AAAAAAACTG
1	AAAAAACGGC
2	AAAAAATCCC
3	AAAAACATAT
4	AAAAACTAGT

In [ ]:

```

```

In [10]:

```
len(df_barcodes)
```

Out[10]:

```
10000
```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```