Functional specification for simulate_xna_signal.py

Jade Minzlaff

Background:

In nanopore DNA sequencing, a sequence of nucleic acids is passed through a protein nanopore, and the change in current for each nucleic acid is recorded as a step graph [1]. Each nucleic acid produces a unique and predictable current value, allowing the genetic sequence to be decoded from the signal using a base-calling program, such as Remora [2].

Conventional genetic code sequencing is based on experimentally determined current values produced by each base (A, T, G, C, U), and basecallers like Remora are not designed to decode genetic sequences which include synthetic, non-standard nucleic acids, known as XNAs. This motivates the creation of tools to aid in sequencing and decoding of XNAs.

This tool will take a genetic sequence including XNAs as an input, and will output the predicted nanopore signal plot for the sequence.

This tool was created using experimentally measured nanopore sequencing data from known sequences of XNAs by the Marchand Lab at the University of Washington , published in 2023 by Hinako Kawabe et al [3].

The specific XNAs included in this study are from the Artificially Expanded Genetic Information System (AEGIS), and are abbreviated as J, K, P, S, V, X, and Z. (S here refers to the carbon based nucleotide S, sometimes abbreviated as Sc).

tldr: pip-installable python package to simulate the expected nanopore signal produced by a given genetic sequence including unnatural base pairs (XNA).

Sources:

[1] https://nanoporetech.com/applications/dna-nanopore-sequencing

[2] https://github.com/nanoporetech/remora

[3] Kawabe, H., Thomas, C.A., Hoshika, S. *et al.* Enzymatic synthesis and nanopore sequencing of 12-letter supernumerary DNA. *Nat Commun* **14**, 6820 (2023). https://doi.org/10.1038/s41467-023-42406-z

User Profile:
This tool is intended for use by lab workers who perform nanopore sequencing of segments of nucleic acids and wish to have a reference to compare to their generated signals, particularly those working with AEGIS XNA bases. This population could include lab technicians, grad students, and research scientists in the fields of synthetic biology and genomics. The ideal user is able to download packages with pip and has a basic proficiency in using Python, but advanced programming skills are not required to use this tool.

Use case 1:
User seeks to generate a simulation of the expected nanopore sequence data generated for a known sequence of bases, to compare to their experimental data. This would allow the user to check for any large deviations between the experimental and simulated sequences and to investigate potential mispairings, mutations, or experimental errors before continuing on to decode their sequence, increasing the user's confidence in their data. The user would input their known sequence of bases, and the tool would output the expected raw nanopore signal, as confident base calling software for XNAs are still being developed

Use case 2:
Each of the AEGIS XNA bases has a natural base it is most similar to, and conventional base callers are unable to differentiate between natural bases and XNA, resulting in incorrect base-calls. If a user is unsure that a base in a sequence is a natural base or an XNA, the user could use the tool to generate simulated plots of both possible sequences, giving the user a visual aid to determine which base is most likely present in the sequence.

Preliminary Project Plan

1. Generate data library.

Download .csv files of experimental data and convert to a pandas dataframe where all possible 4mers (inputs) are paired with their experimentally generated current signal (outputs).

2. Create a for-loop to generate the plot, so that for each given input sequence, a segment of the plot with the appropriate current is added to the output, using step() function in matplotlib.
3. Add realistic noise to simulated data
4. Create a user-interface to prompt the user to input a nucleotide sequence.