



Project Name: CpG Detector

Description:

The project requires you to build a neural network to count the number of CpGs (consecutive CGs) in given DNA (of N, A, C, G, T) sequences.

Example, given “NCACANNTN**CG**GAGG**CG**NA”, the corrected output should be 2.

Requirement:

1. You need to build a LSTM-based solution using PyTorch. (Hint: why we require to solve with LSTM)
2. You can test different settings (model architecture / loss functions / hyper-parameters) along with the parameters already provided in the material (jupyter notebook we provided) to optimize your solution. (we would like to see how you think through this process)
3. You can use any tools like google search / stackoverflow / even chatGPT, but you have to show us how you use them to help you finish the task.

Task:

We will provide a jupyter notebook with some code implemented (not complete and may have bugs), you need to complete the task based on the requirement, make the model running, perform a simple evaluation, and demo your results (build a simple web app).

1. You need to prepare a web app (using streamlit or flask) to allow us to test your model. Basically, you will expect we give you an input and your app (model+UI) can give a proper model output. For example, you can have an input window where we ask you to input a string like “NCACANNTN**CG**GAGG**CG**NA”, then you can have an output where you show us the outputs from the model (your model outputs could be 1.96 or 2.04 and we would like you to show us the raw outputs).
2. You need to send back the code (only the jupyter notebook) with the outputs that you think are valuable for the team.
3. Be prepared on how you approach it. We are not interested in how good the model you build but how you think of the problem and how you solve it with a proper method.

Note: Feel free to implement whatever you need and modify functions/classes/codes that are not in “# DO NOT CHANGE HERE” cell.

Jupyter notebook download link:

<https://drive.google.com/file/d/1cs3XGRggwi5CJqfrydPCl7BkpaxnK6jU/view?usp=sharing>