

Descrição do Projeto

Projeto realizado pelos alunos

Daniel Thielmann

Gustavo Martins

Hiero Costa

Professor e Orientador

Gabriel Souza

Disciplina

DCC103 - Neurociência Computacional

Resumo

Este projeto implementa o modelo de Hodgkin-Huxley (HH) para descrever a propagação de potenciais de ação em um neurônio modelado como um cabo unidimensional (1D) com bainha de mielina. O modelo é baseado na equação do cabo e nos princípios biofísicos do modelo HH para correntes iônicas. A simulação calcula a evolução do potencial de membrana ao longo de uma fibra axonal em função do tempo e do espaço.

O código é projetado para:

1. Simular a dinâmica do potencial de membrana em uma fibra com bainha de mielina.
2. Produzir um GIF animado mostrando a evolução da voltagem ao longo do tempo para o caso onde se tem uma região com e sem bainha de mielina.

Requisitos de Execução

Dependências

As bibliotecas necessárias são:

- **numpy**: Para operações matemáticas e manipulação de arrays.
- **matplotlib**: Para geração de gráficos e visualização.
- **imageio**: Para criação de GIFs animados.

Instale essas dependências com:

```
pip install numpy matplotlib imageio
```

Execução

A simulação é executada via terminal com a seguinte estrutura de comando:

```
$ python main.py config.txt
```

- **main.py**: Nome do script Python.
 - **config.txt**: Arquivo contendo informações do projeto.
-

Entradas

O código aceita os seguintes parâmetros, que podem ser configurados no arquivo **parametros.json** que pode ser definido no **config.txt** para a execução do código. Foram criados 2 tipos de exemplos-base do arquivo **parametros.json**, onde se tem a diferença apenas nos parâmetros referentes a colocação da mielina e maneira em que será aplicada a corrente no axônio:

Parâmetros Biofísicos

- **C_m**: Capacitância da membrana ($\mu\text{F}/\text{cm}^2$) – padrão: **1.0**.
- **g_{Na}**: Condutância máxima dos canais de sódio (mS/cm^2) – padrão: **120.0**.
- **g_K**: Condutância máxima dos canais de potássio (mS/cm^2) – padrão: **36.0**.
- **g_L**: Condutância de vazamento (mS/cm^2) – padrão: **0.3**.
- **E_{Na}**: Potencial de reversão dos canais de sódio (mV) – padrão: **50.0**.
- **E_K**: Potencial de reversão dos canais de potássio (mV) – padrão: **-77.0**.
- **E_L**: Potencial de reversão da corrente de vazamento (mV) – padrão: **-54.387**.

Parâmetros Geométricos

- **a**: Raio do axônio (μm) – padrão: 15.
- **R**: Resistência longitudinal do axônio ($\text{ohm}\cdot\text{cm}$) – padrão: 5000.

Parâmetros de Simulação

- **dx**: Passo espacial (cm) – padrão: 0.01.
- **dt**: Passo temporal (ms) – padrão: 0.01.
- **L**: Comprimento total do axônio (cm) – padrão: 3.0.
- **T**: Tempo total de simulação (ms) – padrão: 50.0.

Corrente Externa

- **J**: Array com dados onde está sendo aplicada uma corrente numa região da fibra do axônio por um determinado tempo.
 - Por ter um formato [tempo, tamanho da fibra] colocamos como base de entrada uma corrente constante no ponto inicial do axônio, ou seja, no ponto [tempo, 1] temos a corrente valendo $20 \mu\text{A}/\text{cm}^2$.
 - Para o segundo caso de parâmetros temos que se coloca a onde se inicia a mielina, onde se finaliza e qual o valor em $\mu\text{A}/\text{cm}^2$ da corrente aplicada dentro dum formato de dicionário como demonstrado no arquivo [parametros2.json](#).

Mielina

- **Mie**: Array com dados onde está sendo colocada a mielina no corpo do axônio.
 - Por ter um formato [tamanho da fibra] colocamos como base de entrada valores 0 onde não tem mielina e 1 onde tem mielina.
 - Para o segundo caso de parâmetros temos que se coloca a onde se inicia a mielina e onde se finaliza ela dentro dum formato de dicionário como demonstrado no arquivo [parametros2.json](#).

Saídas

1. Tabela de Dados

O código gera uma matriz contendo os valores do potencial de membrana para cada ponto espacial e cada instante de tempo.

2. GIF Animado

Um GIF chamado [propagacao_potencial.gif](#) é criado, mostrando a evolução do potencial ao longo do tempo e do espaço. Já para o caso com mielina um GIF nomeado

`propagacao_potencial_mielina.gif` é gerado, mostrando também a evolução do potencial ao longo do tempo e do espaço, agora contabilizando a mielina marcada em vermelho.

3. CSV dos dados

Um CSV para cada um dos casos salvo como nome de `output.csv` e `outputMie.csv` que armazenam os dados da simulação realizada.

Descrição das Funções

Funções Auxiliares

Estas funções calculam as taxas de transição para os canais iônicos:

- `alpha_n(V)`, `beta_n(V)`: Parâmetros do canal de potássio.
- `alpha_m(V)`, `beta_m(V)`: Parâmetros do canal de sódio (ativação).
- `alpha_h(V)`, `beta_h(V)`: Parâmetros do canal de sódio (inativação).

Função Principal

`hodgkin_huxley_1D()`

- Simula a propagação do potencial de ação ao longo do axônio.
- Utiliza métodos numéricos para resolver a equação diferencial parcial da equação do cabo.
- Retorna uma matriz 2D (`V_time`) com os valores do potencial de membrana.

Função de Visualização

`create_hodgkin_huxley_gif(V_time, Mie, output_gif_path, dx, dt, y_amplitude, frame_skip)`

- Cria um GIF animado mostrando a propagação do potencial ao longo do tempo.
- Configura a escala dos gráficos para manter a consistência visual.

Fluxo de Execução

1. Inicialização:

- Abre o arquivo config.txt com informações do projeto.
- Define os parâmetros biofísicos, geométricos e de simulação.
- Configura as condições iniciais para os gates e o potencial de membrana.

2. Simulação:

- Itera ao longo do tempo, calculando os valores do potencial de membrana e atualizando os estados dos canais iônicos.

3. Armazenamento dos Resultados:

- Armazena os valores calculados em uma matriz 2D para visualização posterior.

4. Visualização:

- Gera gráficos para momentos específicos da simulação.
- Cria um GIF para representar a evolução do potencial.

Exemplo de Execução

Parâmetros Utilizados

$C_m = 1.0$

$g_{Na} = 120.0$

$g_K = 36.0$

$g_L = 0.3$

$E_{Na} = 50.0$

$E_K = -77.0$

$E_L = -54.387$

$a = 15$

$R = 5000$

$dx = 0.02$ ou 0.05 , ou para o caso sem mielina 0.01

$dt = 0.0001$ ou 0.001 , ou para o caso sem mielina 0.01

$L = 3.0$

$T = 50.0$

`J = np.array(T/dt, L/dx) onde J[: , 1] = 20`

`Mie= np.array(L/dx) onde Mie[1/dx : 2/dx] = 1`

OBS: os valores de dx e dy foram separados em conjuntos para indicar as combinações q podem ser utilizadas. Os 2 primeiros valores foram calculados a partir da fórmula de restrição " $k \cdot dt/dx^2 \leq 1/2$ " para garantir a convergência do método já que se trata de uma Equação Diferencial Elíptica.

Resultado Esperado

- Um GIF mostrando a propagação da voltagem ao longo do axônio para cada caso. Um com e outro sem mielina.
-

Conclusão

Este projeto oferece uma implementação robusta e personalizável para simular e visualizar a propagação do potencial de ação em um neurônio modelado como um cabo 1D. Ele é uma ferramenta poderosa para entender a dinâmica biofísica dos neurônios, permitindo análise visual e ajustabilidade.