



## ESTUDO COMPARATIVO DO PROCESSAMENTO DE SINAIS BIOMÉDICO ATRAVÉS DE DUAS PLATAFORMAS

João Ludovico Maximiano Barbosa\*<sup>1</sup>, Kaliny Alice Carvalho de Oliveira Magalhães<sup>1</sup>, João Batista Destro Filho<sup>1</sup>

<sup>1</sup>PPGEB – Universidade Federal de Uberlândia

**Resumo** - Atualmente existe uma variedade de plataformas capazes de fazer processamento de sinais biomédicos, utilizando diversas técnicas e linguagens de programação diferentes. Neste trabalho, dois tipos de processamentos de dados – utilizando ora Matlab e ora Python - foram realizados com o objetivo de verificar as diferenças entre os mesmos. Ao analisar o tempo de processamento, Python mostrou-se mais rápido em diversas etapas. Já o Matlab foi mais performático nas etapas designadas “Separação de Épocas”, “pré-processamento e processamento”. Na fase de pré-processamento e processamento o Python se apresentou quase 60 vezes mais lento que o Matlab, concluindo-se que o Matlab evidenciou um comportamento mais veloz. Do ponto de vista de precisão e coleta de dados finais, as duas abordagens são bastante semelhantes.

**Palavras-Chave** - EEG, Matlab, Processamento, Python.

### COMPARATIVE STUDY OF PROCESSING OF BIOMEDICAL SIGNS THROUGH TWO PLATFORMS

**Abstract** - There are currently a variety of platforms capable of processing biomedical signals, using different techniques and different programming languages. In this work, two types of data processing, using Matlab and Python, were carried out in order to verify the differences between them. When analyzing processing time, Python was faster in several steps. Matlab, on the other hand, was more performative in the Epochs Separation, pre-processing and processing stages. In the pre-processing and processing phase, Python was almost 60 times slower than Matlab, concluding that Matlab presented a faster behavior. From a precision point of view, the two approaches are quite similar.

**Keywords** - EEG, Matlab, Processing, Python.

\*joaolmbarbosa@gmail.com

## I. INTRODUÇÃO

O eletroencefalograma (EEG) é um exame que capta os sinais elétricos do cérebro. Essa medição pode ser feita por intermédio de eletrodos invasivos ou externos. A utilização de eletrodos externos mostra-se mais frequente, visto que é uma forma menos agressiva e de baixo custo, se comparada com o método invasivo [1].

As variações dos sinais elétricos medidos no couro cabeludo, são divididos de acordo com suas faixas de frequência, sendo consideradas baixas e altas, e amplitudes. Essa divisão é feita entre cinco ritmos cerebrais e normalmente são relacionadas a estados fisiológicos, como: Ritmo Delta (< 4 Hz) está interligado a estado de sono profundo com grande amplitude. Teta (4 - 7 Hz) é encontrado durante estados de sonolência. Alfa (8 - 13 Hz) é visto durante a vigília. As ondas altas ou rápidas são nomeadas Beta (14 - 30 Hz) e Gama (> 30 Hz) são relacionados a processos cognitivos, como leitura e memória [2].

Atualmente existe uma variedade de técnicas que são capazes de fazer a leitura e processamento desses sinais [3]. Através da análise matemática e estatística, dispositivos computacionais executam desde a redução de ruídos, á gráficos detalhados [4]. Cada software utiliza abordagens e linguagens diferentes de pré-processamento e processamento dos sinais biológicos, o que limita uma padronização dos dados obtidos em diferentes medições. [5]

O Python é uma linguagem de programação que está sob intenso desenvolvimento, [4] sendo considerado hoje em dia um dos principais recursos para análises e processamentos de dados, sendo usado em diferentes tipos de softwares.

Este trabalho tem como objetivo comparar o desempenho das duas plataformas – Python (gratuito) e Matlab (licença, pago) – no desenvolvimento de rotinas dedicadas ao processamento de EEG, orientando eleições mais adequadas às demandas e finalidades de futuros estudos.

## II. MATERIAIS E MÉTODOS

### A. Softwares de Processamento

Através de uma busca na literatura [6] foram encontrados alguns softwares relevantes responsáveis por processar dados de EEG, encontrados na Tabela I a saber: EEGLAB, ELAN,

FIELDTRIP, MNE-Python, PyEEG. Mesmo tendo os mesmos objetivos de processamento, cada qual utiliza abordagens e metodologias diferentes, como também plataformas e linguagens variadas. Essas ferramentas são capazes de fazer filtragem de dados, conversão de referência de média, reamostragem de dados, remoção de artefatos entre outros tipos de análises.

### B. Comparação dos Valores Gerados

Consideramos inicialmente um software para processamento de sinais de EEG, o qual foi redigido em linguagem Matlab [6].

MATLAB é uma interface de programação de alto nível, altamente utilizado em pesquisas acadêmicas [7] devido ao seu caráter estatístico e a disponibilidade de toolbox que facilitam o processamento de dados.

Este software de processamento [6] é dividido em algumas etapas, que são mostradas abaixo:

1. Leitura Arquivo .PLG (Registro do exame)
2. Separação das Épocas
3. Pré-processamento e processamento
4. Cálculo da Porcentagem de Contribuição de Potência - PCP para cada ritmo/banda de frequência
5. Cálculo da Frequência mediana - FM para cada ritmo/banda de frequência
6. Cálculo da coerência dos eletrodos do hemisfério esquerdo com os do hemisfério direito

Esta plataforma foi transcrita da linguagem Matlab para a linguagem Python, pelo primeiro autor, a fim de identificar qual linguagem teria melhor poder de processamento sem perder a confiabilidade nos valores calculados. Além do mais, aproveitou-se a oportunidade para reestruturar os dados gerados no processamento para que o entendimento dos mesmos ficasse mais simples e clara, para isso aplicou-se alguns conceitos de orientação a objeto para tal propósito.

Para comparar o resultado gerado por ambos os softwares Matlab e Python, criou-se um código em Matlab que realiza o cálculo do erro relativo dado pela seguinte fórmula:

$$Err = \left| \frac{V_{calc} - V_{esper}}{V_{esper}} \right| \quad (1)$$

Onde:

- Err - Erro relativo.
- V<sub>calc</sub> - Valor calculado
- V<sub>esper</sub> - Valor esperado

As variáveis geradas pelos softwares que foram comparadas foram:

- **PCP** (vetor de célula) = de tamanho nFrequences [nFrequences comumente tem valor 7 (Delta, Teta, Alfa, Beta, Gama, Super Gama e Ruído)]. Cada elemento desse vetor é uma matriz de escalares de tamanho L x nEpochs que corresponde a valores da contribuição de potência de cada ritmo cerebral, para aquela Época. Onde:
  - L = quantidade de canais EEG do registro [comumente L=20];
  - nEpochs = número de épocas do exame [comumente nEpochs=10];
- **FM** (vetor de célula): de tamanho nFrequences [nFrequences comumente tem valor 7 (Delta, Teta, Alfa,

Beta, Gama, Super Gama e Ruído)]. Cada elemento desse vetor é uma matriz de escalares de tamanho L x nEpochs que corresponde a valores da Frequência Mediana de cada ritmo cerebral. Onde:

- L = quantidade de canais EEG do registro [comumente L=20];
- nEpochs = número de épocas do exame [comumente nEpochs=10];
- **COR\_PairsOfElectrodes** (vetor de cell) = de tamanho nPairsOfElectrodes [comumente nPairsOfElectrodes=8 devido ao cálculo da coerência]. Cada elemento do vetor é uma matriz de escalares de tamanho nEpochs x Y. Onde:
  - nEpochs = quantidade de épocas selecionadas no exame [comumente nEpochs=10];
  - Y = quantidade de frequências gerada pela função do Matlab 'mscohere'

Essas matrizes contêm os valores de coerência calculados, sendo que cada matriz é referente a um par de eletrodo. Ou seja, 1=FP1-FP2, 2=F7-F8, 3=F3-F4, 4=T3-T4, 5=C3-C4, 6=T5-T6, 7=P3-P4 e 8=O1-O2.

Para a comparação entre os softwares escolheu-se 10 exames coletados seguindo o padrão 10-20, todos processados pelos dois softwares, no qual 5 exames são referentes a pacientes considerados normais (ausência de patologias) e os outros 5 são referentes a pacientes que estavam em estado de coma. Adotou-se como valor esperado o valor computado em Matlab, e o valor calculado como sendo o valor computado em Python. Como os dados foram gerados em plataformas distintas, utilizou-se a biblioteca "scipy.io.savemat" que permite o python gerar um arquivo .mat, extensão que é reconhecida no matlab, uma vez que os dados são importados no matlab executa-se o programa para cálculo do erro relativo.

### III. RESULTADOS E DISCUSSÃO

Das ferramentas encontradas na Tabela I, 40% foram implementadas no software MATLAB, 40% Python e apenas 20% dos programas foram feitos com a linguagem C. Das toolbox analisadas 60% das mesmas são declaradas de código aberto. Somente 40% delas possui alguma interface gráfica, as demais necessitam de interação via linha de comandos. 80% dos softwares analisados possuem a opção de importação de arquivos, um grande recurso que facilita o processamento dos dados uma vez que não se torna necessário criar um algoritmo para importar os dados do sinal na ferramenta.

Na Tabela II é apresentado o resultado do cálculo do erro relativo para os 10 exames/registros de EEG, note que as variáveis que foram comparadas são variáveis multidimensionais com 3 dimensões, e o resultado na tabela mostra um único valor de erro para cada exame/registro, este valor de erro foi encontrado realizando a média de todos os erros calculados na matriz multidimensional, do mesmo modo também foi calculado o desvio padrão para dos mesmos.

Para efeito de comparação de tempo de execução de cada etapa mencionada anteriormente utilizou-se um notebook Itau-Tec infoWay modelo w7545 com sistema operacional Windows10 equipado com processador Intel Core i3-2350M 2.30 GHz com 8Gb de memória RAM.

Tabela I: Softwares de Processamento de Sinais

Software	Linguagem	Interface Gráfica	Licença de Uso	Tamanho para download *	Versionamento	Dados Versionamento	Características	Vantagens/Virtudes	Deficiências
EEGLAB	Matlab	Sim	GNU General Public License	30.2 MB	GIT	Commits: 12214 Releases: 0 Contribuidores:14 Último commit: 06/12/2018	Funções: 400 Linhas de código: 50000 Plataforma: Multi	Permite adição de novos plugins que podem ser compartilhados com todo o mundo.	
ELAN	C	Sim		5.5 MB			Plataforma: Linux Memória RAM necessária: > 128MB Memória RAM recomendada: 4 GB	Análise virtual de qualquer tipo de sinal contínuo eletrofisiológico (EEG, MEG, iEEG, ECoG, LFP, etc). Alta velocidade de processamento através de algoritmos otimizados e código C compilado n Analyzer4.	
FIELDTRIP	Matlab	Não	GNU General Public License	76.1 MB	GIT	Commits: 15417 Releases: 1 Contribuidores:75 Último commit: 06/12/2018	Plataforma: Multi Funções alto nível: 108 Funções baixo nível: 858 Linhas de código: 103227	Possibilidade de aplicação de análise em tempo real, devido sua velocidade de processamento	
MNE-Python	Python	Não	BSD licenced	58.5 MB	GIT	Commits: 13824 Releases: 64 Contribuidores: 150 Último commit: 07/12/2018	Linhas de Código em Python: 44000 Linhas de comentário: 22000 Documentação: 35% Está entre os mais bem documentado projetos Python	Suíte de teste com cobertura de 86% das linhas de Código. Leitura de dados em disco sobre demanda, poupando gastos de processamento.	
PyEEG	Python	Não	Open Source	22.8 KB	GIT	Commits: 58 Releases: 0 Contribuidores: 5 Último commit: 13/09/2016		Capaz de exportar recursos para o formato <i>svmlight</i> para chamar ferramentas de aprendizado de máquina. Criado inicialmente para reconhecimento de EEG epilético contra o EEG normal	Não possui um importador de arquivos de EEG, sendo necessário utilizar ferramentas de terceiros (ex: ,EDFBrowser, EEGLAB, etc)

Tabela II: Erro Relativo

Nome PLG	PCP Media ERRO	PCP Desvio Padrão	FM Media ERRO	FM Desvio Padrão	COERENCIA Media ERRO	COERENCIA Desvio padrão
COMA 1	1,24E-15	2,22E-15	2,36E-16	4,75E-16	167,8058	1,51E+04
COMA 2	2,72E-15	5,69E-15	3,16E-16	5,87E-16	13	721
COMA 3	1,68E-15	3,53E-15	2,71E-16	5,68E-16	174,1348	1,63E+04
COMA 4	4,84E-15	6,42E-15	5,18E-16	5,87E-12	Dif. Tamanho	Dif. Tamanho
COMA 5	3,02E-15	4,05E-15	4,61E-16	5,47E-16	Dif. Tamanho	Dif. Tamanho
NORMAL1	2,83E-14	5,69E-14	1,12E-15	2,31E-15	83,0848	7,08E+03
NORMAL2	3,75E-14	8,54E-14	1,31E-15	2,48E-15	193,514	1,88E+04
NORMAL3	2,07E-14	3,54E-14	7,89E-16	1,03E-15	10,8543	122,7668
NORMAL4	2,47E-14	4,87E-14	9,01E-16	1,53E-15	11,9667	196,962
NORMAL5	2,91E-14	5,73E-14	1,27E-15	2,06E-15	9,7615	178,4611

Na Figura 1, pode-se visualizar a comparação do tempo de execução para cada etapa de processamento do sinal citado anteriormente. A etapa de pré-processamento e processamento foi colocada na Figura 2 separadamente para melhorar a visualização da comparação das demais etapas, devido a uma alta discrepância do valor encontrado no tempo de execução em Python.

Figura 1: Tempo de processamento em todas etapas analisadas.

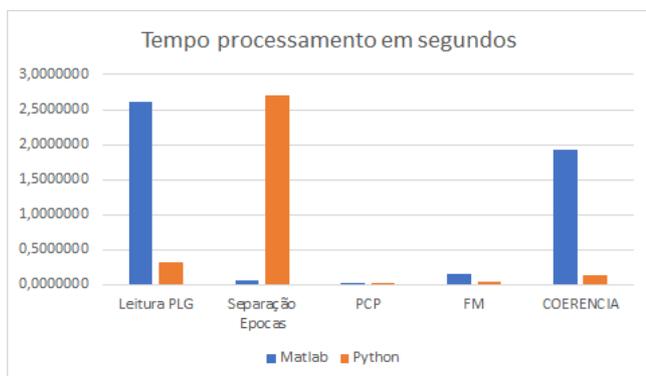
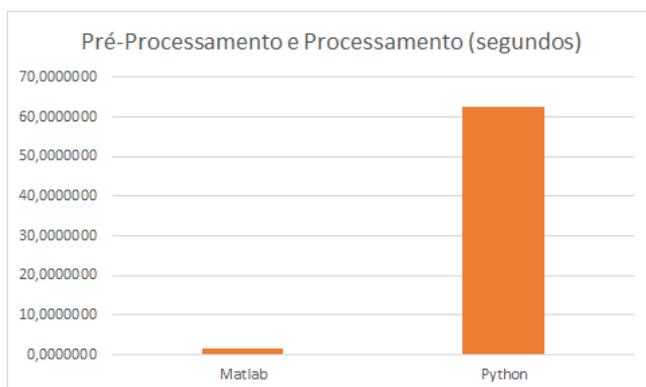


Figura 2: Etapa de Pré-Processamento e Processamento dos dados.



O software Matlab ao terminar o processamento dos exames/registros salva os resultados encontrados em um arquivo com a extensão .MAT e no Python salva-se os resultados em uma arquivo com a extensão .pkl, e na Tabela III pode-se visualizar uma comparação do tamanho dos arquivos gerados por ambos.

Tabela III: Comparação entre o tamanho dos arquivos gerados pelos processamentos em Matlab e Python

Nome PLG	Matlab (kBytes)	Python (kBytes)
COMA_1	4.767	22.233
COMA_2	4.644	23.421
COMA_3	3.885	18.920
COMA_4	4.115	20.519
COMA_5	4.589	20.519
NORMAL1	10.398	52.185
NORMAL2	9.330	49.088
NORMAL3	9.886	50.672
NORMAL4	8.958	44.660
NORMAL5	9.945	46.676

#### IV. CONCLUSÕES

Pode-se observar que o Python foi mais rápido nas etapas de: Leitura PLG, PCP, FM e Coerência. O Matlab foi mais performático nas etapas Separação de Épocas, pré-processamento e processamento. Nota-se que na fase de Leitura PLG e Separação de Épocas as linguagens se compensaram uma com relação a outra.

No entanto na fase pré-processamento e processamento o Python se apresentou quase 60 vezes mais lento que o Matlab, talvez isso possa ter ocorrido devido a forma de criação dos objetos seguindo o padrão orientação a objetos, vale a pena investigar mais profundamente o que levou uma discrepância tão grande. Já com relação aos arquivos gerados como resultado do processamento dos registros/exames, observa-se que os arquivos gerados pelo Python apresentaram tamanho quase 5 vezes maior do que o gerado em Matlab.

Diante do exposto percebe-se que o Matlab se mostrou mais robusto e veloz, no entanto, vale ressaltar que a transcrição para o Python foi realizada de forma mais simples, para trabalhos futuros pretende-se fazer um aprofundamento na linguagem e recursos encontrados no Python, para avaliar se o mesmo possa ter um desempenho semelhante ao encontrado ao Matlab.

### REFERÊNCIAS

- [1] W. Freeman, R. Q. Quiroga, *Imaging Brain Function With EEG*, 1st ed., no. c. New York: Springer-Verlag, 2013.
- [2] L. Sörnmo, P. Laguna. “Bioelectrical Signal Processing in Cardiac and Neurological Applications”, *Elsevier*. Burlington, MA 01803, USA: Academia Press; 1 edition, 2005.
- [3] A. Delorme, et al. “EEGLAB, SIFT, NFT, BCILAB, and ERICA: New tools for advanced EEG processing”. *Computational Intelligence and Neuroscience*, vol. 2011, pp 12. Feb, 2011.
- [4] A. Gramfort, et al. “MNE software for processing MEG and EEG data”. *NeuroImage*, vol. 86, pp. 446–460, Oct, 2013.
- [5] T. Francois et al. “Brainstorm: A user-friendly application for MEG/EEG analysis”. *Computational Intelligence and Neuroscience*, vol. 2011, pp 13, Jan, 2011.
- [6] J. L. M. Barbosa. “Cadastro, processamento e visualização de eletroencefalograma”. *Dissertação (Mestrado em Engenharia Biomédica) - Universidade Federal de Uberlândia*, 145 f., Uberlândia-MG, 2019;
- [7] P. Berens. “CircStat: A MATLAB Toolbox for Circular Statistics”. *Journal Of Statistical Software*, vol 31, Sep, 2009.