

No Stress Elisa: Um resumo

João R. B. K. da Cruz, Lucas Candinho, Matheus P. V. da Silveira

Illum, Escola de Ciência, Centro Nacional de Pesquisa em Energia e Materiais (CNPEM), Brasil.

RESUMO

Este trabalho consiste em um analisador de dados provenientes de um leitor de placas Elisa de 96 poços, que serve para medir dados de absorbância ou fluorescência com alta precisão de múltiplas amostras, suas aplicações sendo vistas tanto na caracterização de nanopartículas, quantificação de proteínas, entre outros. Os dados originados por esse aparato devem ser processados, normalmente utilizando de processamento computadorizado, visto a quantidade de dados recebidos. O objetivo desse projeto, então é facilitar o processamento, provendo para o pesquisador os gráficos necessários para a análise, que deve ser feita caso a caso por um humano. Foi utilizado, então, diversas bibliotecas python para receber, processar e exibir os dados relevantes sem a necessidade do usuário manipular o código diretamente, mas somente provendo os dados necessários.

INTRODUÇÃO

ELISA (Enzyme-Linked Immunosorbent Assay) foi uma técnica criada em 1971 por Peter Perlmann e Eva Engvall que tinha como objetivo realizar testes imunoenzimáticos (ENGVAL; PERLMANN, 1971). Em si, a ideia principal era poder realizar diversos ensaios, com diferentes compostos para identificar os tipos de antígenos presentes em uma amostra (MATTHEWS; FREEDLAND; MIESFELD, 2017).

Essa técnica também está presente na área da química. Em um espectrofotômetro de absorção, é possível obter o espectro de absorção de uma determinada solução (HARRIS; LUCY, 2020). Por outro lado, se o objetivo é estudar e comparar a absorção em diferentes amostras, obter os espectros individuais e compará-los é desnecessário. Nesses casos, pode-se utilizar um leitor de placas (COOPER, 2001). Esse equipamento realiza medidas de absorção em um comprimento de onda específico, mas faz isso para uma placa inteira (96 poços, nesse caso).

A visualização dos dados obtidos diretamente do leitor é um desafio, pois esses são apenas números em uma tabela genérica (A-H – 1-12). Além disso, como as soluções estão em um meio, geralmente água, é necessário retirar a absorbância do solvente, dito "branco" (HARRIS; LUCY, 2020). Assim, criou-se um programa que recebe os dados obtidos do equipamento, realiza o tratamento de dados e entrega ao usuário o gráfico referente ao ensaio.

METODOLOGIA

O projeto foi desenvolvido utilizando da linguagem Python 3.12.7 usando as bibliotecas (1) Pandas; (2) Numpy; (3) Matplotlib; (4) Seaborn; (5) Streamlit (o funcionamento de cada uma será detalhado posteriormente).

O funcionamento geral pode ser resumido em quatro passos principais: primeiro, o programa recebe os dados do usuário, depois trata os dados para que possam ser manipulados pelo código, então, os gráficos são plotados e exibidos para o usuário.

Percebe-se, então, que a execução do programa é em série, cada passo dependendo de seu antecessor, logicamente seguiria-se que o código seria construído em série, porém, por eficiência, todas as partes foram desenvolvidas simultaneamente, em paralelo. Para isso ser possível as estruturas de dados que cada parte utilizaria foram definidas no início do projeto, para que exemplos testes pudessem ser gerado, conforme o padrão, para que cada parte pudesse ser arquitetada sem necessidade da anterior ter sido finalizada.

Assim, cada módulo do projeto foi produzido e testado individualmente com casos de teste-exemplo, construídos artificialmente para simular a saída de outra parte do código. Posterior a isso, o projeto foi testado com dados reais, de autoria própria, para que se assegurasse o funcionamento contínuo e conjunto dos módulos em harmonia.

Por fim, cada módulo recebeu melhorias estilísticas e cada função foi documentada, seguindo o padrão (1) utilidade; (2) argumento(s); (3) retorno(s).

RESULTADOS E DISCUSSÃO

Utilizar-se-á nesta parte o fluxo de execução padrão de um usuário para mostrar o funcionamento do código.

Interface

A interface do sistema foi desenvolvida utilizando a biblioteca Streamlit, proporcionando uma experiência simples e intuitiva ao usuário, seguindo a diretiva **Plug and Play** (do inglês, ligar e usar). Todo o controle da aplicação está concentrado no arquivo `main.py`, que também integra os módulos responsáveis pelo processamento e visualização dos dados.

Na tela inicial, o usuário tem acesso a uma introdução à técnica ELISA, suas aplicações e aos membros da equipe desenvolvedora. Em seguida, na aba *Gráfico*, o usuário pode:

- Fazer o upload da planilha gerada pelo equipamento de leitura ELISA (.xlsx).
- Inserir o layout experimental da placa de 96 poços, que associa cada posição (como A1, B3, etc.) a uma condição experimental. Isso pode ser feito manualmente, via editor interativo de matriz, ou por meio de upload de outro arquivo Excel.
- Selecionar o tipo de gráfico desejado (barras ou linhas).
- Visualizar a matriz montada, editar valores e exportar o layout como .csv.
- Gerar e visualizar os gráficos com as médias e incertezas calculadas automaticamente.

Todas as operações são realizadas localmente, sem necessidade de configuração adicional por parte do usuário. Além disso, os gráficos gerados podem ser baixados como imagens .png, facilitando sua utilização em relatórios e apresentações.

Processamento

Após os dados serem enviados pela interface, o módulo de processamento inicia seu funcionamento. Originário do arquivo `processa_dados.py`, ele é encarregado de receber os dados e realizar os seguintes processos:

1. **Recebimento e separação dos dados:** dividem-se os metadados do equipamento e os dados amostrais de uma planilha.
2. **Limpeza dos dados:** células vazias e colunas não nomeadas são removidas para facilitar os tratamentos posteriores.
3. **Estruturação do layout experimental:** uma lista contendo as condições experimentais e os respectivos poços é transformada em um `DataFrame` organizado. Isso envolve a normalização do número de elementos por linha e a transposição da matriz para tornar as colunas representações diretas das condições testadas.
4. **Mapeamento dos dados experimentais:** os identificadores de poços (como “A1”, “B2”, etc.) são substituídos pelos valores de absorbância medidos nos dados amostrais. Essa substituição é feita poço a poço, e é específica para um comprimento de onda por vez.
5. **Correção do branco e propagação de incertezas:** os poços preenchidos apenas com água são usados para determinar o valor de absorbância de fundo (branco). Esse valor médio é subtraído de todas as medições, com a incerteza sendo propagada de acordo com a fórmula:

$$\sigma_{\text{final}} = \sqrt{\sigma_{\text{branco}}^2 + \sigma_{\text{equipamento}}^2}$$

Isso assegura que o erro final inclua tanto a variabilidade da leitura do branco quanto o erro instrumental.

6. **Cálculo das estatísticas finais:** para cada condição experimental, calcula-se a média das absorbâncias corrigidas e a incerteza associada, somando-se quadraticamente a incerteza da média das medidas com a incerteza da correção do branco.

O resultado final do módulo de processamento é um `DataFrame` contendo, para cada comprimento de onda e condição experimental, a média das absorbâncias corrigidas e suas respectivas incertezas. Esses dados são, então, enviados ao módulo de visualização.

Para as manipulações realizadas aqui, utiliza-se das bibliotecas **Pandas** para geração e realização de operações com `Dataframes`, **Numpy** para normalização e transposição de `Dataframes` e **Math** para operações matemáticas.

Plotagem

O último módulo, `gera_graficos.py`, é responsável por transformar os dados processados em representações visuais compreensíveis. Ele recebe como entrada o `DataFrame` proveniente do processamento.

A biblioteca utilizada para geração dos gráficos é a **Seaborn**, particularmente seu submódulo orientado a objetos `seaborn.objects`, o **Pandas** também é utilizado para manipulações.

Por fim usa-se `io` para manipulação de fluxos de bytes e `PIL.Image` para o retorno da imagem gerada.

Esse módulo é encarregado de pegar os dados tratados e gerar um de dois formatos de gráfico distintos:

- **Barra:** utiliza o método `add(so.Bar(), so.Dodge())`, gerando barras agrupadas por amostra e diferenciadas por cor para cada comprimento de onda;
- **Linha:** utiliza o método `add(so.Line(marker="o"))`, conectando os valores médios com marcadores para representar cada ponto experimental.

Com isso, o módulo de visualização cumpre seu papel final no pipeline, possibilitando que os dados obtidos a partir da análise de placas ELISA sejam representados visualmente de forma clara e personalizada pelo usuário.

CONCLUSÕES

O desenvolvimento do projeto com sucesso seu objetivo de proporcionar uma ferramenta intuitiva para o processamento e visualização de dados provenientes de leituras de placas ELISA. Através da integração de bibliotecas da linguagem Python, foi possível construir um pipeline completo que abrange desde a entrada dos dados brutos até a geração de gráficos.

O uso de uma interface gráfica e a automatização do tratamento dos dados reduz o tempo necessário para a análise.

Por fim, o projeto se mostra promissor como base para futuras expansões, como a incorporação de novos tipos de análise estatística e suporte a outros formatos de entrada.

REFERÊNCIAS

COOPER, M. A. Optical biosensors in drug discovery. *Nature Reviews Drug Discovery*, v. 1, n. 7, p. 515–528, 2001.

ENGVAL, E.; PERLMANN, P. Enzyme-linked immunosorbent assay (ELISA): Quantitative assay of immunoglobulin g. *Immunochemistry*, v. 8, n. 9, p. 871–874, 1971.

HARRIS, D. C.; LUCY, C. A. *Quantitative Chemical Analysis*. 10th. ed. [S.l.]: W. H. Freeman, 2020. 634-638 p. ISBN 9781319164301.

MATTHEWS, H. R.; FREEDLAND, R. A.; MIESFELD, R. L. *Biochemistry: A Short Course*. 4th. ed. [S.l.]: John Wiley & Sons, 2017. 347-350 p. ISBN 9781119347242.