https://eventos.utfpr.edu.br//sicite/sicite2020

Ferramenta para tomada de decisão na análise de dados de RNAs não-codificantes

Tool for decision making in data analysis of non-coding RNAs

Vitor Gregorio
vitorgregorio@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

Alexandre Rossi Paschoal
paschoal@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Cornélio Procópio, Paraná, Brasil

RESUMO

A bioinformática é uma área da ciência que busca analisar, interpretar e solucionar eventos biológicos. Complementando com a análise exploratória de dados, é possível observar os resultados encontrados em formas gráficas ou tabelares, apresentando uma compreensão melhor dos resultados obtidos tanto por parte do bioinformata quanto de um biólogo. Com isto, foi desenvolvida uma ferramenta em Python para realizar a análise exploratória de dados biológicos, gerando boxplots do comprimento, *GC Content* e *GC Ratio*, além de gráfico de barras para dinucleotídeos e trinucleotídeos. Assim, com bibliotecas como Biopython, Numpy, Tkinter e Matplotlib, foi possível a análise das sequências biológicas e criação de gráficos, através de uma interface intuitiva e usual.

PALAVRAS-CHAVE: Bioinformática. Python. Análise exploratória de dados.

ABSTRACT

Bioinformatics is an area of science that seeks to analyze, interpret and solve biological events. Complementing with the exploratory data analysis, it is possible to observe the results found in graphic or tabular forms, presenting a better understanding of the results obtained by both the bioinformation and a biologist. So, a Python tool was developed to do exploratory analysis of biological data, generating boxplots of length, GC Content and GC Ratio, in addition to bar graphs for dinucleotides and trinucleotides. Thus, with libraries such as Biopython, Numpy, Tkinter and Matplotlib, it was possible to analyze biological sequences and create graphics, through an intuitive and usual interface.

KEYWORDS: Bioinformatics. Python. Exploratory data analysis.

Recebido: 19 ago. 2020.
Aprovado: 01 out. 2020.

INTRODUÇÃO

A Análise exploratória de dados, é o processo de sintetização de dados utilizando valores numérico estatísticos, que podem ser apresentadas via gráficos e tabelas, visando verificar a validade de premissas necessárias para a inferência estática; a validação e qualidade dos dados; ou identificar as estratégias analíticas e estatísticas apropriadas(TEO, 2009).

A bioinformática é a área interdisciplinar, que numa visão da computação, busca ajudar de forma rápida e eficiente a análise de dados biológicos, em específico os estudos da biologia molecular. De acordo com Pevsner(2015), o *National Institutes of Health* define a bioinformática como a pesquisa, o desenvolvimento e a aplicação de recursos e técnicas computacionais para aumentar o uso de dados biológicos, incluindo as tarefas de aquisição, armazenamento, análise ou visualização dos dados.

No caso, RNAs não-codificantes são RNAs ou ARNs (ácido ribonucleico) que são transcritos, mas não conseguem ser traduzidos em proteínas, ainda que tenham suas funções biológicas, como alterações da cromatina, regulação pós-transcricional, organização nuclear, tradução e outros processos de desenvolvimento (CORREIA, 2007).

O objetivo desse trabalho foi o desenvolvimento de uma ferramenta de análise exploratória de dados biológicos através de uma interface gráfica, para contribuir na tomada de decisão sobre dados de RNAs não-codificantes. Nela são apresentados gráficos e relatórios com os resultados da análise, além de arquivo para o usuário exportar a saída, visando facilitar a utilização dos dados por parte do usuário.

MATERIAL E MÉTODOS

Para a realização do experimento, foram utilizadas sequências públicas de *Circular RNA* (circRNA), que estão anotadas no genoma das plantas *Arabidopsis thaliana, Glycine max, Hordeum vulgare, Oryza sativa* e *Triticum aestivum*, e estão sendo aplicados frequentemente em pesquisas científicas. Foram extraídos os arquivos no formato FASTA dos bancos de dados (BD) PlantCircNet (ZHANG, 2017) e Plant Circular RNA Database (PlantCircBase) (CHU, 2018), totalizando assim dez genomas.

A análise exploratória foi executada através de um script em Python 3.6, juntamente com as bibliotecas Tkinter, para o desenvolvimento da interface; OS para obter o caminho dos arquivos; Pandas 1.0.1 (TEAM, 2020) e Numpy 1.18.1 (OLIPHANT, 2006) para a manipulação dos dados; Biopython 1.76 (COCK, 2009) para a leitura dos arquivos FASTA; Matplotlib 3.2.2 (HUNTER, 2007) para a criação de gráficos; e Arff 0.9 (UBERSHMEKEL, 2012) para a escrita de arquivos no formato *Attribute-Relation File Format* (ARFF).

Para a realização da leitura dos arquivos FASTA, foi utilizado o pacote Biopython (COCK, 2009) que foi desenvolvido para a utilização em ferramentas de biologia na computação. Assim, com o objeto Seq foi possível manusear as informações obtidas de cada genoma, como o comprimento de cada sequência ou a sua sequência em si.

Para a análise gráfica de cada sequência, foi utilizado o pacote Matplotlib (HUNTER, 2007) que é baseado no software MatLab (THE MATHWORKS, 2017) e permite a criação de diversos tipos de gráficos em Python. Dessa forma, foi obtido os comprimentos de cada sequência através do objeto Seq, armazenado os valores em uma lista, para posteriormente ser realizada a plotagem do *boxplots* de cada genoma. Além disso, foi editado a cor do gráfico, para o usuário realizar uma interpretação melhor dos resultados. Também, foi editada a legenda de cada *boxplots* com o nome do arquivo FASTA utilizando expressão regular, e os gráficos criados foram salvos automaticamente com o nome do arquivo.

Assim, com as configurações de plotagem e do objeto Seq já estabelecidas, se tornou possível a análise de novas características, como o valor do *GC Content* e *GC Ratio* de cada sequência, e a quantidade de dinucleotídeo e trinucleotídeo de cada genoma. Portanto, foi utilizada a Eq. GC (1) que calcula o valor do *GC Content*, onde esse valor foi armazenado em vetores para em seguida realizar a plotagem do *boxplots* de cada genoma. Para a plotagem do *GC Ratio*, foi utilizada a Eq. (2), desconsiderando as sequencias sem citosina, e da mesma forma que o *GC Content*, foi armazenado os valores em listas para posteriormente realizar plotagem do *boxplots.* Para os dinucleotídeos e trinucleotídeos, foram realizadas comparações de *strings* para 16 dinucleotídeos e 24 trinucleotídeos em cada genoma, para em seguida realizar a plotagem em barras com as quantidades de cada genoma.

$$GcContent = \frac{A+T}{G+C} \quad (1)$$

$$GcRatio = \frac{G}{C} \quad (2)$$

Em seguida, foi desenvolvido o salvamento de arquivos em formato *Comma-separated values* (CSV) e ARFF, sendo utilizada a biblioteca Numpy e Pandas para realizar a manipulação dos dados em *data frame*, onde para cada genoma, seriam salvos os nomes de cada sequência, seguido do seu comprimento, *GC Content* e *GC Ratio*. A biblioteca Pandas permitiu a criação de *data frame* para em seguida converter para CSV, e junto com o pacote Arff (UBERSHMEKEL, 2012) foi possível transformar de *data frame* para ARFF, possibilitando assim o usuário utilizar o resultado no programa Weka (HALL, 2009).

Além disso, para tornar a ferramenta mais amigável, foi desenvolvida uma interface gráfica para torná-lo mais intuitiva. Foram colocados botões para adicionar ou remover arquivos, em que foi salvo o caminho de cada arquivo em uma lista. Dessa lista, foi utilizada a biblioteca OS para remover o caminho do arquivo, e junto com expressões regulares, foram armazenados apenas os nomes de cada arquivo em outra lista, para posteriormente serem utilizados nos gráficos e para salvar os arquivos. Além disso, foram criados cinco botões para a plotagem de cada característica e dois para cada forma de salvarem os resultados. Também foram colocados três configurações para o usuário editar no gráfico, sendo a primeira uma caixa de seleção para o tamanho da fonte da legenda de cada *boxplots*, o segundo um botão de checagem que o usuário pode decidir entre mostrar ou não os *outliers* dos *boxplots*, e a terceira outra caixa seleção que

permite personalizar a cor do *boxplots* para um ou dois banco de dados, onde a cor de cada *boxplots* ficará intercalada.

Por fim, de forma a tornar a interface melhor para o usuário, buscou-se executar as heurísticas de Nielsen (MACEDO, 2017). Por exemplo, na prevenção de erros o usuário recebe um aviso quando não é possível remover arquivos, ou quando não é possível realizar a plotagem do gráfico, ou também para confirmar a exclusão de arquivos. Além disso, a heurística de visibilidade do status do sistema também é aplicada, pois o usuário está sempre sendo informado das ações que está realizando, como na adição de um novo arquivo, na exclusão, na conclusão de alguma plotagem, no salvamento do arquivo e quando a interface está carregando a ação. Por fim, a heurística de consistência e padrões também foi utilizada, buscando manter sempre as mesmas fontes e cores (MACEDO,2017).

RESULTADOS E DISCUSSÃO

Na Figura 1 é possível observar a interface criada, sendo possível evidenciar os botões de adicionar arquivo, remover o último arquivo, remover todos os arquivos, os cinco botões de plotagem e os dois botões para salvar o resultado em forma de texto CSV ou ARFF, e também as três caixas de seleção para os *boxplots.*

Figura 1 – Interface da ferramenta

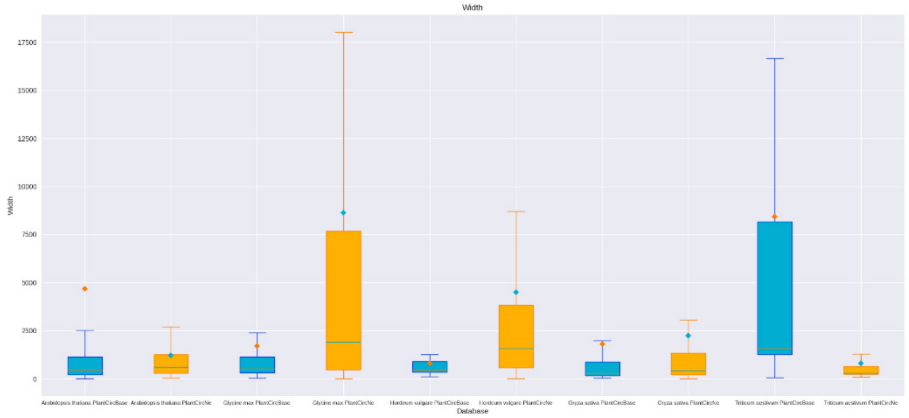
Fonte: Autoria própria (2020)

Ao clicar no botão *Open Files* é aberto uma caixa para a seleção do arquivo, e após selecionar o arquivo, é exibido o caminho do mesmo. Por fim, caso o usuário adicione dez arquivos, é apresentada uma mensagem informando que não é possível adicionar novos arquivos, em que esta decisão de limitar a quantidade de arquivos foi para tornar a exibição dos *boxplots* mais limpa.

O usuário também pode optar por remover o último arquivo adicionado ou remover todos os arquivos, clicando nos botões *Remove Last File*, e *Remove All Files*, respectivamente. Além disso, toda vez que o usuário clica em algum desses botões, aparece uma caixa de confirmação, para perguntar se deseja mesmo realizar alguma dessas ações.

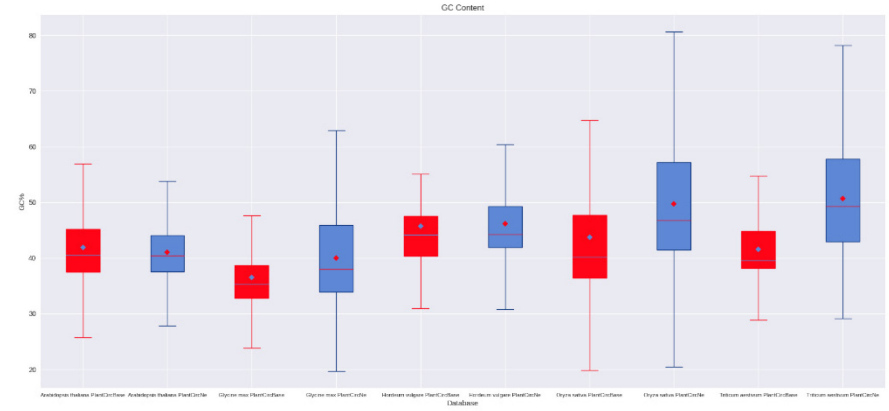
Com os dez bancos de dados adicionados, é realizada a plotagem do gráfico de comprimento pressionando o botão *Width*. Com as configurações para dois bancos de dados, sem mostrar os *outliers* e a fonte com tamanho 8, obteve-se o gráfico da Figura 2. Em seguida, foi plotado o gráfico de *GC Content* clicando no botão *GC Content*, e posteriormente o gráfico de *GC Ratio* através do botão *GC Ratio*, obtendo-se assim as Figuras 3 e 4, respectivamente. Nas três imagens, é possível observar uma linha no meio de cada *boxplot*, que representa a mediana, e também, um diamante, que representa a média de cada *boxplot*.

Figura 2 – Boxplot do comprimento



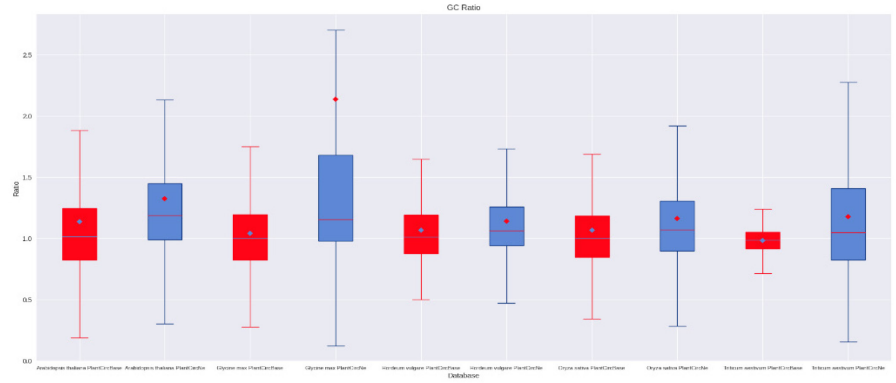
Fonte: Autoria própria (2020)

Figura 3 – Boxplot do GC Content



Fonte: Autoria própria (2020)

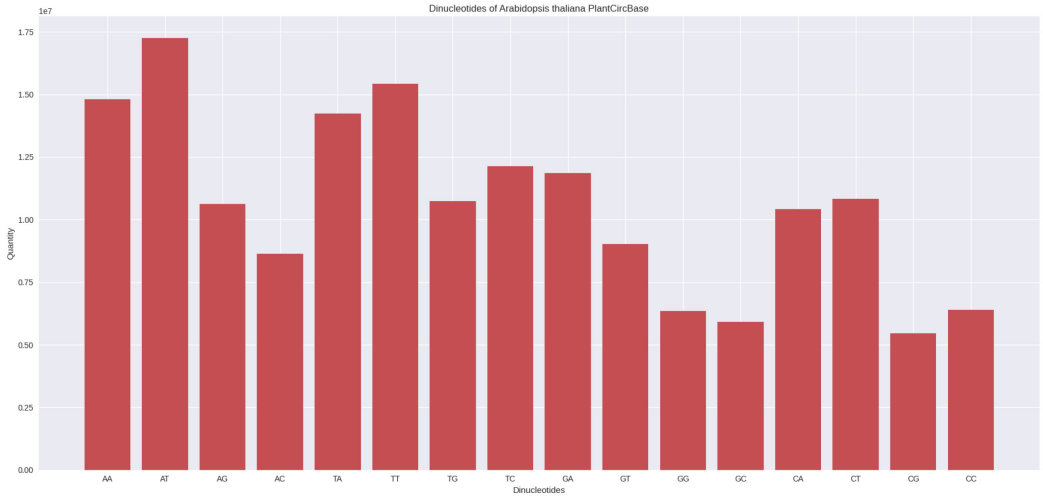
Figura 4 – Boxplot do GC Ratio



Fonte: Autoria própria (2020)

Em seguida, foi realizada a plotagem dos gráficos de dinucleotídeos de cada genoma, como por exemplo na Figura 5 pode-se observar o gráfico de barras do primeiro genoma.

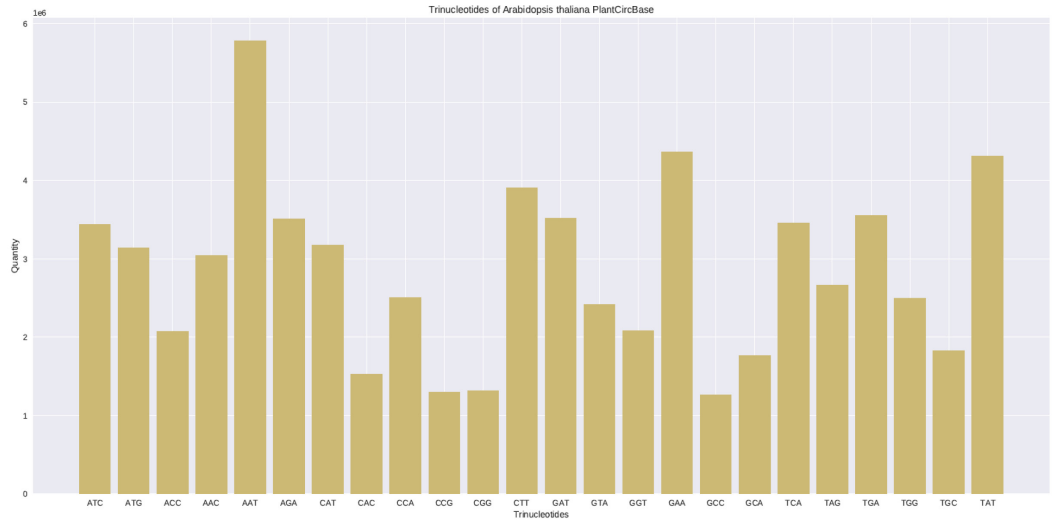
Figura 5 – Dinucleotídeo da Arabidopsis thaliana do BD PlantCircBase



Fonte: Autoria própria (2020)

Posteriormente, os gráficos de trinucleotídeos foram plotados, como por exemplo na Figura 6 pode-se observar o gráfico de barras do primeiro genoma.

Figura 6 – Trinucleotídeo da Arabidopsis thaliana do BD PlantCircBase



Fonte: Autoria própria (2020)

Por fim, os arquivos foram salvos em formato CSV e ARFF, utilizando os botões *Save in CSV* e *Save in ARFF*, respectivamente. Dessa forma, cria-se arquivos em cada formato, e os resultados obtidos são salvos neles.

CONCLUSÃO

Com o objetivo de desenvolver uma ferramenta para análise exploratória de dados, a linguagem de alto nível Python permitiu criar uma interface simples e fácil para usuário. Com gráficos coloridos e editáveis, um rápido entendimento dos resultados foi disponibilizado.

Além disso, com os resultados obtidos, o usuário pode utilizá-los de forma a serem salvos no formato CSV, ou até mesmo em ARFF, que posteriormente pode ser utilizado no software Weka para *machine learning.*

Dessa forma, a ferramenta apresenta uma ótima ergonomia, permitindo o software ter uma boa vida útil, e por ser uma ferramenta amigável, pode ser facilmente utilizada por usuários leigos na computação.

AGRADECIMENTOS

Agradeço ao professor Dr. Alexandre Rossi Paschoal pela oportunidade, confiança, paciência e aprendizado passado durante o período do projeto; à Tatianne da Costa Negri, pelo suporte; e aos meus amigos, em especial Aline Bini e Ana Clara Bergamin, pelo companheirismo e ajuda.

REFERÊNCIAS

CHU, Q. et al. Characteristics of plant circular RNAs.Briefings in Bioinformatics, v. 21,n. 1, p. 135–143, nov. 2018. ISSN 1477-4054. DOI:10.1093/bib/bby111. eprint:https://academic.oup.com/bib/article-pdf/21/1/135/32376256/bby111.pdf. Disponível em: https://doi.org/10.1093/bib/bby111 . Acesso em: 18 jul. 2020.

COCK, P. J. A. et al. Biopython: freely available Python tools for computationalmolecular biology and bioinformatics.Bioinformatics, v. 25, n. 11, p. 1422–1423, mar.2009. ISSN 1367-4803. DOI:10.1093/bioinformatics/btp163. eprint:https://academic.oup.com/bioinformatics/article-pdf/25/11/1422/944180/btp163.pdf. Disponível em: https://doi.org/10.1093/bioinformatics/btp163. Acesso em: 18 jul. 2020.

CORREIA, J. D.; CORREIA, A. D. Funcionalidades dos RNA não codificantes(ncRNA) e pequenos RNA reguladores, nos mamıferos.REDVET. Revista Electrónicade Veterinaria, Veterinaria Organización, v. 8, n. 10, p. 1–22, 2007.

HALL, M. et al. The WEKA data mining software: an update.SIGKDD Explorations,v. 11, n. 1, p. 10–18, 2009.

HUNTER, J. D. Matplotlib: A 2D graphics environment.Computing in Science &Engineering, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007. DOI:10.1109/MCSE.2007.55.

MACEDO, G. M. 10 heurísticas de Nielsen para o design de interface, 2 ago. 2017. Disponível em: https://brasil.uxdesign.cc/10-heur%C3%ADsticas-de-nielsen-para-o-design-de-interface-58d782821840 . Acesso em: 19 jul. 2020.

OLIPHANT, T. E.A guide to NumPy. [S.l.]: Trelgol Publishing USA, 2006. v. 1.

PEVSNER, J.Bioinformatics and functional genomics. [S.l.]: John Wiley & Sons, 2015.

TEAM, T. pandas development.pandas-dev/pandas: Pandas. [S.l.]: Zenodo, fev. 2020.DOI:10.5281/zenodo.3509134. Disponível em: https://doi.org/10.5281/zenodo.3509134 . Acesso em: 18 jul. 2020.

TEO, Y. Y. Exploratory data analysis in large-scale genetic studies.Biostatistics, v. 11,n. 1, p. 70–81, out. 2009. ISSN 1465-4644. Disponível em: https://doi.org/10.1093/biostatistics/kxp038 . Acesso em: 18 jul. 2020.

THE MATHWORKS, INC.MATLAB version 9.3.0.713579 (R2017b). Natick,Massachusetts, 2017.

UBERSHMEKEL.arff version 0.9. [S.l.: s.n.], 11 mai. 2012. Disponível em: https://pypi.org/project/arff/ . Acesso em: 19 jul. 2020.

ZHANG, P. et al. PlantCircNet: a database for plant circRNA–miRNA–mRNAregulatory networks.Database, v. 2017, dez. 2017. Disponível em: https://doi.org/10.1093/database/bax089 . Acesso em: 19 jul. 2020.