Bioinformatics, 2024, 24(14),btae123
doi: 10.1093/bioinformatics/123
Advance Access Publication Date: 30 maio 2024
Original Paper

Análise de sequência

Trabalho final de conclusão da disciplina *Python para Biocientistas*

Ludmilla Veado[1,*], Larissa[1], Jesus[2], Fernando[2], Anderson[2], Felipe[2], Juliana[2], Israel[1] and Célio Dias Junior[2]

[1]Departamento de Hidrobiologia, Av. Washington Luis., [2]Departamento de Genética, Av. Washington Luis.

*Autor correspondente. Email: ludmillaveado@estudante.ufscar.br

Editor Associado: Célio Dias Junior

Recebido em 30 de maio; revisado em 30 de junho; aceito em 30 de julho

Resumo

Motivation: A linguagem de Python representa uma ferramenta que permite vasta aplicação e fácil entendimento na área da pesquisa científica. Permite análises complexas, porém rápidas de parâmetros ambientais apontando um avanço para a geração de conhecimento no campo da biologia computacional.

Results: As análises de índices como Curva do Coletor e Diversidade de Shannon realizadas com a linguagem Python mostraram resultados coerentes com aqueles observados quando são realizadas em um sistema de banco de dados mais convencional, como o Excel.

Availability: A nossa implementação é pública e está disponível em https://github.com/Lyuda0405/lyudabioinformaticaoceanographer/tree/main/trabalho%20final

Contact: ludmillaveado@estudante.ufscar.br

Informação Suplementar: Dados suplementares estão disponíveis em https://github.com/Lyuda0405/lyudabioinformaticaoceanographer/blob/main/material%20suplementar

1 Introdução

A linguagem de programação*Python* é uma das mais utilizadas nas áreas de desenvolvimento *web*, análise de dados, *design* e, atualmente, no campo científico.Sua crescente utilização culminou num maior desenvolvimento da linguagem em si com a criação de Bibliotecas, ferramentas, *frameworks* que possibilitam rapidez e robustez no desenvolvimento de códigos (LOPES, 2022). Com linguagem simples e intuitiva, programadores e usuários mantêm em desenvolvimento, *softwares*em código aberto, sem necessidade de licenças pagas e ainda, os dados podem ser referenciados por qualquer variável ao escrever o código, aumentando a sua legibilidade (YURKOVICH *et al*., GOODMAN *et al*., apud LOPES, 2022). Nesse sentido, ao utilizar a linguagem do *Python*, os pesquisadores têm em mãos uma ferramenta capaz de fornecer dados de experimentos, por exemplo, manipular e processar tais dados, de visualizar os dados para se entender o processo e a partir dos resultados, gerar tabelas, gráficos, mapas para relatórios técnicos, *papers*, teses, dissertações e apresentações (https://estruturas.ufpr.br/disciplinas/pos-graduacao/introducao-a-computacao-cientifica-com-python/).

O Programa de Pós Graduação em Ecologia e Recursos Naturais (PPGERN) da Universidade Federal de São Carlos (UFSCAR) ofereceu uma disciplina que integrou o conhecimento da linguagem de *Python* à pesquisa científica. A disciplina *Python para Biocientistas* foi ministrada por aluno de pós-doc e, mesmo sendo optativa, teve praticamente todas as vagas ofertadas, preenchidas por discentes de mestrado e doutorado, vinculados ao PPGERN. O conteúdo foi estruturadode modo a oferecer aos estudantes noções de algoritmo, pensamento lógico, estruturação dos dados e palavras chave com ordenação de operações, partindo de noções de *looping*, desenvolvimento de funções e automação. Ainda o conhecimento dos pacotes com módulos e funções e a visualização dos dados por meio de geração de tabelas e diversos tipos de gráficos e suas aplicações para as distintas análises realizadas por pesquisadores científicos.O presente *paper*que é um produto da disciplina mostra o desenvolvimento do conteúdo programático dado em aula, envolvendo parte teórica e prática aplicadas em trabalho final para conclusão da disciplina *Python para Biocentistas*.

2 Metodologia

A aplicação do conteúdo teórico e prático ministrado em sala de aula se deu de forma didática por meio de realização de trabalho final para a conclusão da disciplina *Python para Biocentistas*em formato de *paper*, como este, seguindo o *template* da presente revista.

2.1 Geração Tabela de OTUs aleatória

A metodologia aplicada foi estruturada conforme as Instruções para a elaboração do trabalho final, iniciando com a geração de Tabela de OTUs aleatória, de sequenciamento de DNA de amostras ambientais. Os dados da Tabela OTU foram distribuídos em 26 colunas e 100 linhas, tendo o máximo de *reads* pré-definido nas Instruções em até 100.00 *reads*/coluna. Entre 40% e 75% dos dados de cada amostra, foram gerados zeros aleatórios, procurando manter ainda o máximo de 100.000 *reads*/coluna.

Para a geração desta tabela (Tabela_Original), a instalação de pacotes foi necessária no sistema *Pythone* já realizada durante as aulas, com importação de módulos e criação de códigos para suas respectivas funções. Nesta etapa foram utilizados os pacotes *Numpy* e *Pandas*.

2.2 Análise dos Dados

A análise dos dados foi realizada por análises estatísticas de Rarefação, que fornecem uma estimativa esperada da riqueza de espécies (número de espécies) dentro de um determinado número de indivíduos registrados em cada amostra. Com o pacote *Numpy*já instalado, seus módulos foram importados e assim, foi criada uma função para a rarefação da Tabela_Original, considerando o número mínimo de contagens (*reads*) por amostra desta tabela.

A nova tabela, a Tabela_Rarefação foi gerada pela análise e foi salva como um arquivo tabulado ou seja, onde os dados são separados por tabulação (“\t”), utilizando o pacote *Pandas*.

Na sequência, a análise estatística realizada foi a de Normalização. Tal análise considerou que cada OTU dentro da amostra fosse convertida na sua proporção dentro da amostra e então, multiplicada por um número grande.E ainda, seu resultado foi convertido a um número íntegro.

Para tanto,foi utilizado o pacote *Pandas*, com respectivos módulos importados para a criação da função de normalização da Tabela_Original, e assim, criada a Tabela_Normalização.

Para cada uma das três tabelas foi calculada a média e desvio padrão dos dados. Importando os módulos do pacote *Pandas*, códigos foram criados para as funções Média e Desvio Padrão dos dados originais, dados rarefeitos e dados normalizados.

A média e o desvio padrão foram representados por gráfico BoxPlot e que foi plotado com o uso do pacote*Matplotlib.pyplot*, considerando então os dados das médias dos tratamentos rarefeito e normalizado.

Para a criação da Curva do Coletor, a função foi gerada dentro dos módulos do pacote *Numpy* e a criação dos gráficos que representem a curva do coletor foi gerada usando o pacote *Matplotlib.pyplot*que plotou uma curva para cada tratamento (Original, Rarefação e Normalização).

O Índicede Shannon também foi calculado. Dentro do pacote *Numpy*, foi criada uma função para tal índice.

2.3 GitHub

O GitHub pode ser considerado uma ferramenta essencial para criadores de *softwares*, representando uma plataforma de hospedagem de códigos, como o da linguagem *Python*. Esta plataforma tem serviço ‘de nuvem’, portanto os programadores e demais usuários podem contribuir em projetos de códigos privadose/ou de códigos públicos de qualquer lugar do mundo (www.github.com).

Com isso, os dados gerados por meio dos respectivos pacotes, módulos e funções para cada análise, bem como seus *scripts* e *main.py* estão disponíveis nesta plataforma e podem ser acessados no repositório criado para a disciplina *Python para Biocentistas*, por meio do *link* https://github.com/Lyuda0405/lyudabioinformaticaoceanographer/tree/main/trabalho%20final. A linguagem *Python* seguiu o padrão PEP 8 (https://peps.python.org/pep-0008/).

3 Resultados

Os resultados gerados para cada uma das análises mostraram a gama de tratamentos que um conjunto de dados pode ser analisado. E ao utilizar a linguagem *Python*, tais análises puderam ser realizadas de modo simples, rápido, sendo acessível, tanto para usuários avançados como iniciantes na área da computação científica.

3.1 Dados Tabela_Original

Considerando os dados originais, a Tabela_Original (Tabela 1), encontra-se no Material Suplementar em https://github.com/Lyuda0405/lyudabioinformaticaoceanographer/blob/main/material%20suplementar .

3.2 Dados Tabela_Rarefação

Para a análise de Rarefação, a Tabela_Rarefação mostrou os dados rarefeitos (Tabela 2). No presente contexto esse método buscou responder, considerando o número mínimo de *reads* para o sequenciamento de DNA de cada amostra, quantos genes seriam registrados em cada amostra, considerando o mesmo número de *reads*, adaptado de Roswell, Dushoff, Winfree (2021).

Tabela2. Tabela_Rarefação com os dados rarefeitos por amostra e por espécie.

	A	B	C
"OTU_{1}"	3	4	1
"OTU_{2}"	5	1	2
"OTU_{3}"	2	4	2

3.3 Dados Tabela_Normalização

Considerando a análise de Normalização, onde ocorre a escalada de valores, para que eles estejam dentro de uma faixa específica, no caso, o máximo de 100.000 *reads* por amostra.

Tabela3. Tabela_Normalização com os dados normalizados por amostra e por espécie.

	A	B	C
"OTU_{1}"	0.25	0.25	0.25

"OTU_{2}"	0.50	0.50	0.50
"OTU_{3}"	0.75	0.75	0.75

3.4 Média e Desvio Padrão

As funções para a média e desvio padrão foram criadas e representadas em uma tabela específica (Tabela 4) e segue abaixo:

Tabela4. Tabela de Média e Desvio Padrão das *reads* por amostra de cada tratamento.

Tratamento	Média	DesvioPadrão
Normalizado	30.0	14.142
Rarefeito	35.0	14.142
Original	40.0	14.142

A média de *reads* por amostra entre os tratamentos foi maior com os dados originais, enquanto que o desvio padrão de cada média não variou (Figura 1. Boxplot).

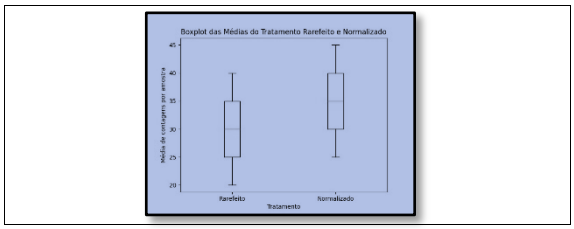

Figura 1. Gráfico boxplot com as médias de *reads* por amostra e seus respectivos desvio padrão para os tratamentos Normalizado e Rarefeito.

3.5 Curva do Coletor

A curva do coletor no presente contexto indica se o número de *reads* por amostra foi o suficiente para caracterizar a diversidade genética de cada amostra dentro da comunidade em estudo. Tal análise é representada pelo gráfrico da Curva do Coletor e segue na Figura 2.

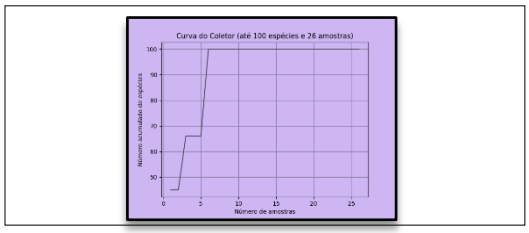

Figura 2. Curva do coletor com o número de *reads* por amostra coletada no estudo.

3.6 Índice de Shannon

O Índice de Shannon considera a proporção de indivíduos de determinada espécie dentro da comunidade registrada por amostra, por exemplo. E ele pode ser calculado pela seguinte fórmula:

$$H' = -\sum p_i * Ln\ p_i \quad (1)$$

No presente trabalho, o seu valor foi de 1,487 para os dados das 100 espécies distribuídas nas 26 amostras coletadas.

Agradecimentos

Os autores agradecem aos revisores pelas contribuições ao manuscrito. Ainda agradecem a equipe que foi a campo para coleta de dados e ao laboratório que efetuou o sequenciamento e a leitura do DNA de todas as amostras da pesquisa.

Financiamento

Este trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Conflito de Interesse: nada a declarar

Referências

Github. Disponível em: https://github.com/. Acesso em [03 maio 2024].

Lopes, E. R. **Utilização de linguagem Python para modelagem e simulação do Modelo de Silva e Cerqueira no tratamento de efluentes por microalgas.** 2022. 87 f. Dissertação (Mestrado em Recursos Hídricos e Saneamento) – Universidade Federal de Alagoas, Centro de Tecnologia, Maceió, 2022.

Matplotlib. Disponível em: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.html. Acesso em: [30 maio 2024].

Numpy The fundamental package for scientific computing with Python. Disponível em:https://numpy.org/. Acesso em: [30 maio 2024].

Roswell, M., J. Dushoff, and R. Winfree. 2021. "A conceptual guide to measuring species diversity." *Oikos* 130 (3): 321–38. https://doi.org/10.1111/oik.07202.

Pandas. Disponível em: https://pandas.pydata.org/Acesso em: [30 maio 2024].

Python Software Foundation. PEP 8 -- Style Guide for Python Code. Disponível em: https://peps.python.org/pep-0008/. Acesso em: [16 maio 2024].

Universidade Federal do Paraná. Disciplina de Pós-graduação: Introdução à Computação Científica com Python. Disponível em: https://estruturas.ufpr.br/disciplinas/pos-graduacao/introducao-a-computacao-cientifica-com-python/. Acesso em: [29 maio 2024].