Avaliação do paralelismo em classificadores taxonômicos de sequências de rRNA usando Qiime2

Pedro Gaspary, Caetano Müller, Dalvan Griebler, Eduardo Eizirik

[1] Escola Politécnica, Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)
Porto Alegre – RS – Brasil

{gaspary2310,caemuller1}@gmail.com,

{dalvan.griebler,eduardo.eizirik}@pucrs.br

Resumo. *Classificação de sequências de rRNA é de suma importância para análise de microbiomas. Portanto, este trabalho avaliou o desempenho e eficiência do paralelismo de três algoritmos de classificação taxonômica do Qiime2. Entre eles, o VSearch apresentou a melhor eficiência na paralelização, mas também os maiores tempos de execução. Os outros dois, Naive-Bayes e Hybrid, apresentaram desempenho similar entre si, este sendo mais rápido até o quinto grau de paralelismo, e consumindo pouco menos memória que aquele.*

1. Introdução

A classificação taxonômica é de extrema importância para que possamos melhor entender e analisar microbiomas e sua relação com sistemas maiores. Como indicado por [Lu and Salzberg 2020], sequências de RNA ribossômico 16s são essenciais para analisar e identificar comunidades de bactérias e sua interação com outros organismos. Neste artigo, o software usado para tal classificação foi o *Qiime2 (Quantitative Insights Into Microbial Ecology)*[Bolyen et al. 2019], uma ferramenta de código aberto de uso frequente na literatura com recursos do estado da arte para análise de DNA microbial. O foco do estudo é na aplicação da paralelização e análise de desempenho dos algoritmos do plugin *feature-classifier* do *Qiime2* para demosntrar a diferença de performance entre eles.

Esse trabalho está estruturado de maneira a relatar trabalhos relacionados da área na Seção 2, apresentar o software *Qiime2* e seus classificadores com mais detalhes na Seção 3 e trazer os resultado encontrados e analisá-los na Seção 4. Por fim, traremos conclusões e discutiremos possíveis trabalhos futuros na Seção 5.

2. Trabalhos Relacionados

Esse artigo é uma continuação do trabalho apresentado em [Müller et al. 2022], buscando estudar a comparação de desempenho dos algoritmos já estudados naquele trabalho com a versão *Hybrid* que utiliza diferentes facetas dos outros dois. Além disso, é trazida uma análise do consumo de memória dos três algoritmos. Vale evidenciar que o algoritmo *BLAST* do *Qiime2* foi considerado como outro candidato para esse estudo, porém ele não possui versão paralela, portanto não foi testado.

O trabalho [Bolyen et al. 2019] apresenta o software *Qiime2* e [Bokulich et al. 2018] descreve inúmeros testes com seus classificadores taxonômicos, apontando o *Naive-Bayes* como o algoritmo com a maior precisão. Já [Lu and Salzberg 2020] investiga as ferramentas *Kraken2* e *Bracken*, as quais, de maneira conjunta, exercem a mesma função que

o *Qiime2*, comparando-as. Esse trabalho se restringe a avaliar o desempenho dos algoritmos de classificação do *Qiime2*.

3. Classificadores Taxonômicos de Sequências de DNA

A ferramenta *Qiime2* [Bolyen et al. 2019] tem diversas aplicações e muitas funcionalidades, contudo o foco desse trabalho está nos algoritmos de classificação taxonômica do plugin *feature-classifier*. O algoritmo *VSearch* utiliza um método de comparação dividido em duas fases. A primeira fase consiste em filtrar as sequências-alvo, com as sequências-referência do banco de dados. Isso é feito com base em códons (sequências de três nucleotídeos consecutivos), identificando as sequências do banco de dados que têm mais similaridade com a sequência a ser classificada. Na segunda fase, a busca é realizada, alinhando a sequência-alvo com diversas sequências do banco, começando pela que possui mais códons em comum. Nos experimentos desse artigo, a busca terminava assim que a primeira sequência era aceita pelos critérios do algoritmo, porém existe a opção de receber mais de uma sequência como resultado.

Já o *Naive-Bayes*, algoritmo padrão da biblioteca *scikit-learn*, utiliza um classificador taxonômico pré-treinado com aprendizado de máquina. O classificador analisa cada sequência e prevê a linhagem taxonômica mais provável a partir do treinamento com uma base de dados. Finalmente, o *Hybrid* utiliza uma conjunção dos dois algoritmos descritos previamente. O algoritmo realiza uma chamada ao VSearch que por sua vez identifica apenas correspondências exatas entre a sequência-alvo e o banco de dados. Isso é feito por meio de uma tabela *Hash* que é preenchida com as sequências do banco até que um par perfeito seja encontrado. Em seguida, as demais sequências são classificadas pelo classificador *sklearn* pré-treinado.

4. Resultados

Os experimentos foram executados em uma máquina Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz, contando com 20 núcleos e 40 *threads*. Cada core com *hyperthread* tem 640KB privada L1, 20MB privada L2 e 27.5MB de L3 compartilhada. A máquina possui 141GB de *RAM*. O kernel era o Linux 5.4.0-135-generic e usava OS Ubuntu 20.04.5 LTS. O ambiente *Qiime2*, na versão 2022.11, foi criado com *miniconda3* versão 23.1.0.

Vale ressaltar que as amostras de rRNA foram todas coletadas em cisternas de brómelias no Centro de Pesquisas e Conservação da Natureza PRÓ-MATA. Dessas amostras, foram analisados componentes do ribossomo procariótico, do tipo 16S, que devido a sua baixa taxa de mutação são comumente usados em análises taxonômicas. A preparação dos dados, e limpeza das sequências foram realizados com uso da ferramenta *DADA2*, que remove ruídos e erros de sequenciamento das amostras. Foram testados o consumo de memória e média de tempo para execução dos três algoritmos de classificação descritos na seção 3. Como referência para as classificações nos experimentos foi usada a base de dados SILVA [Quast et al. 2012]. Visando a manter os critérios usados em [Müller et al. 2022], os experimentos foram realizados alcançando um grau máximo de paralelismo de 10 *threads*, apesar dos 20 núcleos físicos.

Os experimentos relativos ao tempo foram executados 5 vezes para cada algoritmo; e, ao fim, foram calculados a média e desvio-padrão dos valores medidos. Seria

ideal executar os experimentos mais vezes, porém isso não foi possível para esse trabalho por questões de tempo. Uma comparação entre os resultados para cada algoritmo estão dispostos no gráfico da Figura 1. Como já evidenciado por [Bokulich et al. 2018] e [Müller et al. 2022], o *VSearch* possui tempo de execução bastante elevado em relação aos demais. Isso se dá por não ter um classificador, portanto precisando fazer de pesquisas de alto custo computacional para classificar sequências que não encontra na base de referência.

A Tabela 2 apresenta resultados mais detalhados para os tempos de execução. O *VSearch* apresentou os melhores *speed-ups*, tanto no maior grau de paralelismo, quanto em média de eficiência por *thread* usada, provavelmente por ter uma carga de trabalho maior. O *Hybrid*, por usar uma conjunção dos outros algoritmos, apresentou um tempo melhor até o quinto grau de paralelismo, mas não escalou tão bem quanto o *Naive-Bayes*. Isso se dá, pelo fato de que o primeiro já é bastante otimizado na versão sequencial, e tem uma carga de trabalho menor. Uma carga pequena, ao ser distríbuida, aumenta *overhead* de comunicação, diminuindo a eficiência da paralelização. Enquanto o *Naive-Bayes* tem um carga maior, que pode ser melhor distribuída em mais *threads*, logo a escalabilidade medida foi maior.

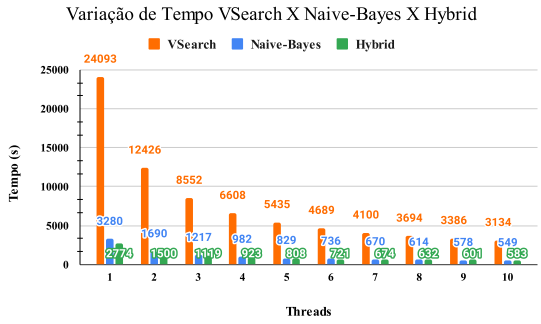

Figura 1. Tempos médios de execução.

	VSearch	*Naive-Bayes*	*Hybrid*
***Speed-up* no maior grau de paralelismo (10 *threads*)**	7.69	5.97	4.76
Eficiência média para os diferentes graus de paralelismo	86.41%	75.92%	66.17%

Figura 2. Resultados experimentais do paralelismo

Os experimentos de consumo de *RAM* foram executados uma vez para cada algoritmo para diferentes graus de paralelismo. Nos gráficos da Figura 3, estão as comparações dos consumos de memória para cada algoritmo com 1, 5 e 10 *threads*. Como o objetivo é analisar o consumo de memória, o eixo do tempo está em escala logarítmica, para que as retas dos algoritmos *Naive-Bayes* e *Hybrid* fiquem mais visíveis em comparação ao *VSearch*, que apresentou um tempo de execução bastante elevado.

O *VSearch* teve um pico próximo de 20.7GB em todos os casos, o que indica que esse alto gasto de memória ocorre após o processamento paralelo, possivelmente representando o gasto para escrita dos resultados da pesquisa. Já o *Naive-Bayes* e o *Hybrid* tiveram picos próximos de 18GB para 1 *thread*, 36GB para 5 *threads* e 60GB para 10 *threads*, mas vale ressaltar que em todos os casos, o consumo do *Naive-Bayes* foi sutilmente superior.

5. Conclusões

Esse trabalho trouxe uma análise do desempenho de três algoritmos de classificação taxonômica do *Qiime2* e como eles se comparam. Os resultados encontrados para o tempo

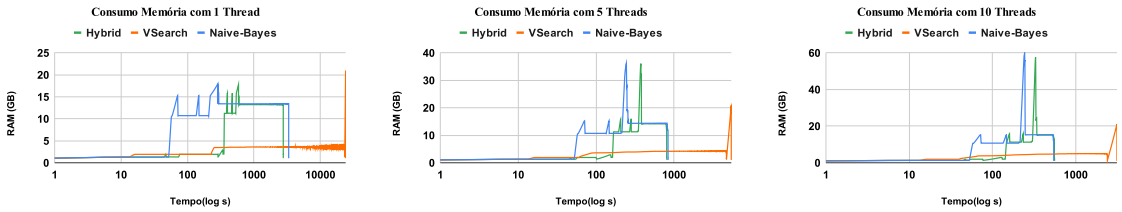

Figura 3. Comparação do consumo de memória entre os algoritmos.

de execução, tanto para o *VSearch* quanto para o *Naive-Bayes*, foram semelhantes aos expostos em [Müller et al. 2022]. Ademais, ficou evidente nos resultados apresentados que os algoritmos *Naive-Bayes* e *Hybrid* tem desempenhos parecidos e escalam de maneira parecida com o paralelismo, este tendo desempenho ligeiramente melhor em memória àquele, e até o quinto grau de paralelismo, em tempo de execução. Já o *VSearch* apresentou uma melhor eficiência na aplicação do paralelismo, não escalando em memória e apresentando os maiores *speed-ups*. Apesar disso, o *VSearch* é em media 6.51 vezes mais lento que o *Naive Bayes* e 6.79 vezes mais lento que o *Hybrid* e, como analisado em [Bokulich et al. 2018], menos preciso que o *Naive-Bayes*.

O baixo tempo de execução é importante para que mais pesquisas possam ser conduzidas sobre os ecossistemas sendo estudados. O alto consumo de memória dos algoritmos mais velozes, no entanto dificulta a produção de mais trabalhos pela necessidade de maior poder computacional. Sobre trabalhos futuros, a otimização desses algoritmos seria relevante para a comunidade, como mencionado em [Müller et al. 2022]. Além disso, a avaliação de outras ferramentas pode ser interessante, como o *Kraken2* e *Bracken*, que [Lu and Salzberg 2020] apresentou e concluiu terem desempenho superior quando comparados ao *Qiime2* em relação ao tempo de execução e ao consumo de memória.

Referências

Bokulich, N., Kaehler, B., Rideout, J. R., Dillon, M., Bolyen, E., Knight, R., Huttley, G., and Caporaso, J. (2018). Optimizing taxonomic classification of marker-gene amplicon sequences with qiime 2's q2-feature-classifier plugin. *Microbiome*, 6.

Bolyen, E., Rideout, J. R., Dillon, M., Bokulich, N., Abnet, C., Al-Ghalith, G., Alexander, H., Alm, E., Arumugam, M., Asnicar, F., Bai, Y., Bisanz, J., Bittinger, K., Brejnrod, A., Brislawn, C., Brown, C. T., Callahan, B., Caraballo Rodríguez, A., Chase, J., and Caporaso, J. (2019). Reproducible, interactive, scalable and extensible microbiome data science using qiime 2. *Nature Biotechnology*, 37:1.

Lu, J. and Salzberg, S. L. (2020). Ultrafast and accurate 16s rrna microbial community analysis using kraken 2. *Microbiome*, 8.

Müller, C., Löff, J., Griebler, D., and Eizirik, E. (2022). Avaliação da aplicação de paralelismo em classificadores taxonômicos usando qiime2. In *Anais da XXII Escola Regional de Alto Desempenho da Região Sul*, pages 25–28, Porto Alegre, RS, Brasil. Sociedade Brasileira de Computação (SBC).

Quast, C., Pruesse, E., Yilmaz, P., Gerken, J., Schweer, T., Yarza, P., Peplies, J., and Glöckner, F. O. (2012). The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. *Nucleic Acids Research*, 41(D1):D590–D596.