Workflow CellHeap: Portabilidade e Reprodutibilidade

Gabriel Plaza, Maria Clicia Castro

[1]Departamento de Informática e Ciência da Computação
Instituto de Matemática e Estatística
Universidade do Estado do Rio de Janeiro (UERJ)
São Francisco Xavier Street, 524, Maracanã
Rio de Janeiro, Brasil

gabrielplaza@hotmail.com.br, clicia@ime.uerj.br

1. Introdução

O sequenciamento de RNA de célula única (scRNA-seq) se tornou o estado da arte na pesquisa sobre a heterogeneidade e a complexidade dos diferentes tipos de células dentro do organismo humano. Para gerar conhecimento biológico correto e confiável, a análise de dados de scRNA-seq exige software otimizados e uso de recursos computacionais de forma eficiente, seja em supercomputadores, servidores ou em nuvem.

O *workflow* CellHeap foi desenvolvido para analisar grandes conjuntos de dados de scRNA-seq com controle de qualidade, fornecendo confiabilidade, portabilidade, reprodutibilidade e extensibilidade [Silva et al. 2021]. Este artigo descreve a migração da plataforma CellHeap, originalmente desenvolvida num supercomputador, para novos ambientes de execução em servidores com recursos computacionais mais restritos (tanto em capacidade de processamento quanto em memória).

A plataforma CellHeap é baseada num *workflow* composto por cinco fases, que podem conter diferentes ferramentas que são escolhidas de acordo com o conteúdo biológico sendo investigado. As modificações realizadas buscam melhorar o desempenho, a flexibilidade e a compatibilidade com versões mais recentes das ferramentas e suas dependências. Nossos resultados mostram que a migração e execução do *workflow* CellHeap são possíveis e fáceis de serem realizadas.

2. Descrição da Migração da Plataforma CellHeap

A Figura 1 mostra as quatro primeiras fases da plataforma CellHeap com as principais ferramentas, bibliotecas e saídas produzidas. As ações realizadas em cada uma destas fases são: (i) curadoria de amostras, download e validação; (ii) geração da matriz de contagem de genes; (iii) controle de qualidade e (iv) redução de dimensionalidade e análise de *clustering*.

Na migração usamos o Miniconda, o gerenciador de pacotes Conda e a biblioteca `Anaconda Project`. Os projetos contêm todos os componentes necessários para executar um aplicativo e são definidos em um arquivo de configuração. Nenhuma configuração especial é necessária quando se troca de ambiente.

O primeiro obstáculo à reprodutibilidade foi na atualização do pacote `Cellranger`, que possui uma licença proprietária e não está disponível para instalação através do gerenciador de pacotes `conda` de forma oficial. Para contornar este obstáculo configuramos as variáveis de ambiente no `Anaconda Project`.

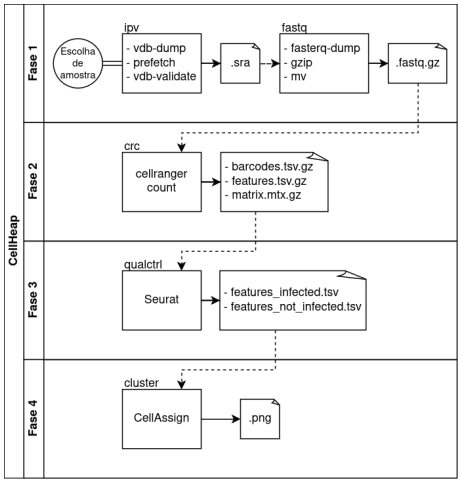

Figura 1. Fases do CellHeap

A dependência mais importante para realização do controle de qualidade é o pacote `Seurat` [Hao et al. 2024]. Atualizamos para a versão 5, que exigiu ajustes no código devido à mudanças incompatíveis na classe `SeuratObject`.

A atualização do pacote `Cellassign` [Zhang et al. 2019] gerou erros e a biblioteca não está recebendo manutenção ativa. A solução foi uma modificação simples, de apenas uma linha de código, solucionada com um *fork* no código do `Cellassign` e corrigido o erro. Disponibilizamos o pacote sem prejuízo à reprodutibilidade do *workflow*. Como ele possui aceleração através de GPU, incluímos as dependências do pacote fornecidos pela plataforma TensorFlow.

3. Ambientes de Execução e Análise dos Resultados

Para demonstrar a portabilidade e reprodutibilidade usamos dois servidores: **Minerva** para a migração e testes; e **Pixel** parar validação da migração e reprodução. As Tabelas 1 e 2 mostram as informações detalhadas sobre as CPUs e as GPUs usadas, respectivamente.

Máquina	CPU	Número de *cores*	Clock Básico	Cache
Minerva	AMD Ryzen™ 7 5700G	8	3.8 GHz	4 MB L2 e 16 MB L3
Pixel	Intel® Xeon® E-2388G	8	3.2 GHz	16 MB L3

Tabela 1. CPUs usadas nos nossos experimentos

Máquina	GPU NVIDIA	Arquitetura	CUDA Cores	Clock Básico	Memória Global
Minerva	GeForce RTX® 3060	Ampere	3584	1.32 GHz	8 GB
Pixel	GeForce RTX® 4080	Ada Lovelace	9728	2.21 GHz	16 GB

Tabela 2. GPUs usadas nos nossos experimentos

Consideramos 2 amostras de dados brutos como entrada, SRR11537954 de 2.2GB e SRR11537952 de 2.3GB, com dados broncoalveolares GSE145926 do NIH GEO.

A Tabela 3 mostra o tempo de execução do ambiente **Minerva**, em segundos, de cada fase do *workflow* Cellheap, para as duas amostras. As colunas 5 e 6 mostram a Fase 4 e o Tempo Total de execução, respectivamente, separando os tempos sem e com GPU.

Amostra	Fase 1	Fase 2	Fase 3	Fase 4		Tempo Total	
				S/ GPU	C/ GPU	S/ GPU	C/ GPU
SRR11537954	719	1040	11	3167	692	4937	2462
SRR11537952	684	1005	18			4874	2399

Tabela 3. Tempo de execução (s) por fase no servidor Minerva.

A utilização da GPU na Fase 4 reduz o tempo total de execução em praticamente 50%. Analisando individualmente as fases do *workflow*, observe que a Fase 3 é a que menos consome tempo de execução.

Para demonstrar a reprodutibilidade utilizamos o ambiente **Pixel**. A reprodução foi feita através de um arquivo do projeto, (cópia do projeto da máquina Minerva). Com a plataforma CellHeap completamente instalada na máquina **Pixel**, o *workflow* foi executado para obtenção dos tempos de execução (Tabela 4).

Amostra	Fase 1	Fase 2	Fase 3	Fase 4		Tempo Total	
				S/ GPU	C/ GPU	S/ GPU	C/ GPU
SRR11537954	895	878	8	1420	394	3201	2175
SRR11537952	829	859	14			3122	2096

Tabela 4. Tempo de execução (s) por fase no servidor Pixel.

O tempo total de execução do *workflow* CellHeap no ambiente **Pixel** é menor do que do ambiente Minerva, mas as fases têm comportamentos similares. O ambiente Pixel possui maior capacidade de memória RAM e uma GPU com mais CUDA *cores*.

4. Conclusões

Este artigo descreveu a migração e reprodução da plataforma CellHeap usando o gerenciador de pacotes Conda e a biblioteca `Anaconda Project`. Durante o processo de migração, algumas modificações foram necessárias, se compararmos com a versão original, para obter o melhor desempenho possível de acordo com o ambiente computacional alvo, que são dois servidores que possuem GPU como aceleradores.

Consideramos adequadas a portabilidade e reprodutibilidade com os procedimentos realizados. No futuro podemos estender facilmente o *workflow* CellHeap incorporando novos pacotes, se necessário.

Referências

Hao, Y. et al. (2024). Dictionary learning for integrative, multimodal and scalable single-cell analysis. *Nature Biotechnology*, 42(2):293–304. Publisher: Nature Publishing Group.

Silva, V. S. et al. (2021). CellHeap: A Workflow for Optimizing COVID-19 Single-Cell RNA-Seq Data Processing in the Santos Dumont Supercomputer. In *Anais do Simpósio Brasileiro de Bioinformática (BSB)*, pages 41–52. SBC. ISSN: 2316-1248.

Zhang, A. W. et al. (2019). Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling. *Nature Methods*, 16(10):1007–1015.