Inferência de Redes de Regulação Gênica Usando Programação Genética Cartesiana Paralela

Luciana N. S. Prachedes[1], **José Eduardo Henriques da Silva**[1], **Heder Soares Bernardino**[1], **Itamar Leite de Oliveira**[1]

[1]Universidade Federal de Juiz de Fora (UFJF)
Juiz de Fora – MG – Brasil

{luciananascimento,jehenriques,heder,itamar.leite}@ice.ufjf.br

Abstract. *The inference of gene regulatory networks (GRNs) is important in Systems Biology as it allows for the understanding of patterns of interactions between genes. These findings are useful in providing knowledge regarding diseases and helping to develop drugs. Evolutionary computation techniques, such as Cartesian Genetic Programming (CGP), have been used to infer GRNs with promising results. However, CGP has scalability issues. Here, GRNs are infered efficiently using high-performance computing approaches. Computational experiments show that the method developed in this scientific initiation program can infer GRNs faster than other ones from the literature with symbolic solutions. The gain in processing time of the presented parallel technique in relation to the sequential one is up to 104%.*

Resumo. *A inferência de Redes de Regulação Gênica (GRNs) é importante em Biologia Sistêmica, pois permite o entendimento de padrões de interações entre genes. Essas descobertas são úteis para fornecer compreensão sobre doenças e ajudar no desenvolvimento de fármacos. Técnicas de computação evolutiva, como a Programação Genética Cartesiana (CGP), têm sido utilizadas para inferir GRNs com resultados promissores. Entretanto, a CGP tem problemas de escalabilidade. Aqui, GRNs são inferidas de forma eficiente usando abordagens de computação de alto desempenho. Experimentos computacionais mostram que o método desenvolvido nesta iniciação científica é capaz de inferir GRNs mais rapidamente do que outros da literatura com soluções simbólicas. O ganho em tempo de processamento da técnica paralela apresentada em relação ao formato sequencial é de até 104%.*

1. Introdução

1.1. Caracterização do problema

Redes usualmente são estruturas representadas por nós e arestas. Nas redes de regulação gênica (GRNs, do inglês Gene Regulatory Networks), os nós são os genes e as arestas são as relações regulatórias existentes entre eles –como ativação e inibição [McCall 2013]. A Figura 1 exemplifica uma rede com 4 genes, com suas ativações (setas azuis) e inibições (vermelhas).

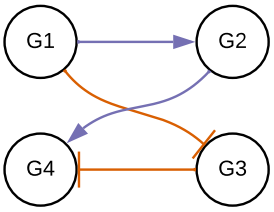

Figura 1. GRN com 4 genes (G1, G2, G3 e G4).

A computação evolucionista é amplamente usada para tratar diversos problemas, como por exemplo inferir GRNs [Noman and Iba 2007, Streichert et al. 2004]. Uma característica comum dessas metaheurísticas é que requerem muitos cálculos da função objetivo e, assim, acabam sendo estratégias de busca computacionalmente caras. Dessa forma, faz-se necessário o desenvolvimento de métodos mais eficientes e o uso de computação de alto desempenho torna-se oportuno.

1.2. Motivação

O entendimento das relações regulatórias entre genes é útil o processo de solução de problemas biomédicos e relacionados à saúde [Emmert-Streib et al. 2014b]. Isso justifica a necessidade de métodos que forneçam soluções interpretáveis para os especialistas de domínio. Particularmente, GRNs são úteis no projeto eficiente de fármacos e nos cuidados personalizados com a saúde [Medeiros et al. 2019, Delahaye-Duriez et al. 2016].

Diferentes pacientes podem responder de forma variada a um mesmo medicamento devido a –entre outras razões– divergências nas interações gênicas causadas pelos seus históricos genéticos [Van Der Wijst et al. 2018]. Essas interações gênicas podem ser compreendidas com a construção de GRNs. Isso possibilita a localização dos principais genes envolvidos em doenças específicas e a indicação de tratamentos personalizados, evitando que pacientes sejam expostos a fármacos ineficazes e efeitos colaterais.

A identificação de biomarcadores para propósitos de diagnóstico, predição e prognóstico de doenças (como câncer) podem ser feitos via modelos de GRNs, como em [Emmert-Streib et al. 2014a]. Isso é possível pois as características do câncer são representadas por vias biológicas em vez de genes individuais e uma particularidade das vias biológicas é que seus genes constituintes interagem ativamente uns com os outros.

1.3. Trabalhos relacionados

A Computação Evolucionista é aplicada em diversas áreas, tais como otimização e reconhecimento de padrões [Dumitrescu et al. 2000]. Além disso, também é aplicada à inferência de GRNs através de Otimização por Enxame de Partículas [Palafox et al. 2013] e Programação Genética [Qian et al. 2008, Ma et al. 2019], por exemplo. Em [da Silva et al. 2020], GRNs booleanas são inferidas usando Programação Genética Cartesiana (CGP, do inglês Cartesian Genetic Programming), um método evolutivo proposto por Julian Miller em 1999 [Miller et al. 1999] que usa a representação de um grafo para codificar programas. Como técnicas evolutivas são computacionalmente caras, uma CGP paralelizada em Unidades de Processamento Gráfico (GPUs, do inglês Graphic Processing Units) para evolução de redes neurais artificiais foi desenvolvida em [Silva et al. 2021], onde observou-se uma redução de 97,92% do tempo total de execução

em relação à versão sequencial. Portanto, um procedimento de inferência de GRN que utilize CGP com paralelismo tende a trazer ganhos em termos de custo computacional.

1.4. Objetivos

O objetivo principal dessa iniciação científica tem sido inferir GRNs via uma CGP paralelizada em GPU. Como objetivos secundários, pretende-se (i) utilizar computação de alto desempenho na criação de modelos booleanos via CGP, e (ii) inferir modelos simbólicos de GRNs visando facilitar a extração de conhecimento desse fenômeno.

1.5. Contribuições

Dentre as principais contribuições observadas durante o desenvolvimento desta iniciação científica pode-se destacar o projeto e implementação de uma CGP paralela via GPU para a inferência de modelos booleanos de GRNs. Experimentos computacionais foram executados com dados de expressão gênica que contemplam as características da tecnologia Single-cell RNA sequencing (scRNA-seq). Observou-se ganhos nos tempos de processamento da técnica paralela em relação à sequencial de até 104% sem haver perda na qualidade das soluções.

2. Métodos

A CGP possui esse nome pela representação dos indivíduos no formato de uma matriz bidimensional de nós. O indivíduo é um grafo direcionado acíclico. Uma das vantagens dessa representação é que a CGP pode lidar com uma ampla gama de estruturas computacionais. Além disso, esta é uma representação mais compacta quando comparada às árvores da Programação Genética tradicional [Miller 2011]. A estratégia evolutiva comumente utilizada pela CGP é apresentada no Algoritmo 1.

Algorithm 1 Estratégia Evolutiva comumente utilizada pela CGP $(1 + \lambda)$ [Miller 2011].

1: População inicial gerada aleatoriamente
2: Seleciona o indivíduo mais apto como progenitor
3: **while** critério de parada não é atingido **do**
4: Muta o progenitor para gerar λ novos indivíduos
5: Avalia os λ novos indivíduos
6: Seleciona o melhor dos $(1 + \lambda)$ indivíduos para ser progenitor
7: **end while**

A CGP pode ser usada em diversos problemas e um deles é a evolução de Redes Neurais Artificiais (RNAs), como em [Khan et al. 2010]. Dado o alto custo computacional associado à CGP, uma versão paralela da CGP foi proposta em [Silva et al. 2021] para gerar RNAs para tratar um problema de reconhecimento de atividade humana.

Algoritmos evolutivos são naturalmente paralelizáveis, uma vez que a população de soluções candidatas de cada iteração/geração pode ser avaliada simultaneamente, já que não existe dependência entre as avaliações dos indivíduos. A avaliação engloba a maior parte do esforço computacional desse tipo de método e existem duas abordagens para realizá-la em paralelo [Koza 1992]: (i) paralelizando a avaliação completa do indivíduo, para que essa aconteça simultaneamente para vários indivíduos, e (ii) ao nível

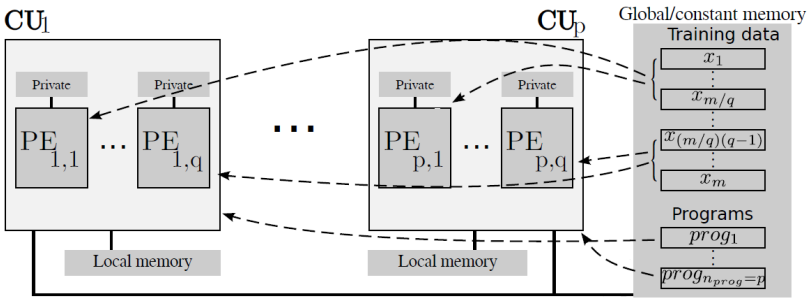

Figura 2. Esquema de paralelismo usado na P-CGPANN [Silva et al. 2021].

das instâncias, onde cada indivíduo é executado sequencialmente, mas em paralelo sobre o conjunto de dados.

A estratégia adotada em [Silva et al. 2021] e ilustrada na Figura 2 envolve o paralelismo populacional por unidade computacional e a avaliação de cada indivíduo explora o paralelismo de dados nos elementos de processamento. É importante destacar que essa estrutura de paralelismo é adequada para GPUs por conta de sua arquitetura.

Em [da Silva et al. 2020], o processo de inferência de redes de regulação gênica ocorre da seguinte forma: (i) dados de expressão gênica são binarizados, (ii) um modelo booleano é criado usando uma CGP, (iii) o modelo booleano obtido é convertido em um sistema de equações diferenciais ordinárias, e (iv) uma estratégia evolutiva define os parâmetros do modelo contínuo. Apesar de produzir boas soluções, o custo computacional é um limitador em [da Silva et al. 2020], dado que a complexidade computacional pode ser exponencial em relação a quantidade de genes envolvidos na rede.

Buscando amenizar o custo computacional do processo de geração de modelos booleanos de GRNs desenvolvido em [da Silva et al. 2020], o presente trabalho de iniciação científica discorre sobre o desenvolvimento de uma abordagem paralela em GPU da CGP. O paralelismo adotado aqui segue o desenvolvido em [Silva et al. 2021]. Neste trabalho, a representação dos indivíduos foi alterada para a geração de modelos booleanos de GRNs. Portanto, a estratégia apresentada é capaz de inferir modelos booleanos de GRNS como em [da Silva et al. 2020], mas com tempo de processamento reduzido.

3. Experimentos Computacionais

Os experimentos foram executados com o *benchmark* disponibilizado em [Pratapa et al. 2020], o qual apresenta um conjunto de problemas usando dados scRNA-Seq. Como não há noção física de tempo, este é estimado através da inferência de progressão temporal do desenvolvimento das células, denominado *pseudotime*.

Os dados utilizados aqui são oriundos do problema *Gonodal Sex Determination* (GSD). Esse problema envolve 19 genes com 2 *pseudotimes*, que são uma medida de quão longe uma célula se moveu através do progresso biológico. Além disso, 10 diferentes configurações do problema com 2.000 células foram considerados, conforme apresentado em [Pratapa et al. 2020]. Os experimentos foram executadas em uma máquina com processador Intel Core i7-4790k 4,4GHz e GPU NVIDIA GTX760Ti, sistema operacional Windows 10 64-bit, compilador g++ MinGW 8.1 sem parâmetros de otimização e biblioteca OpenCL 1.2 da NVIDIA.

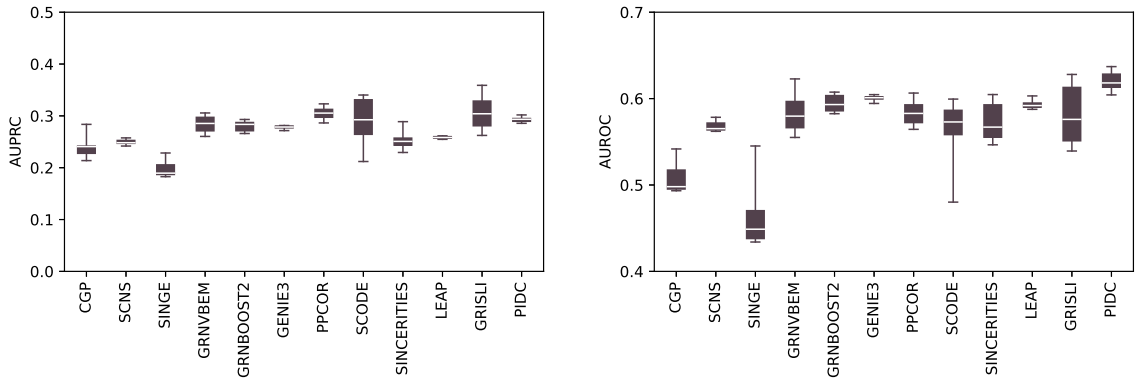

Figura 3. Boxplots de AUPRC e AUROC dos métodos do BEELINE e da CGP.

Tabela 1. Tempo (em segundos) para a execução dos problemas sequencialmente (Seq.) e em paralelo (Par.). Os melhores valores estão em negrito.

	GSD	2000-1	2000-2	2000-3	2000-4	2000-5	2000-6	2000-7	2000-8	2000-9	2000-10
Seq.	Evol.	1182.04	1145.87	1160.91	1235.97	1206.41	1227.43	1136.15	1221.60	1149.54	1180.49
	Prog.	1189.47	1198.18	1200.01	1211.81	1181.14	1188.11	1199.04	1225.58	1159.89	1230.24
Par.	Evol.	**631.04**	**614.36**	**641.38**	**605.94**	**597.35**	**603.66**	**619.12**	**594.37**	**587.78**	**619.62**
	Prog.	**633.08**	**619.98**	**647.25**	**611.36**	**602.78**	**609.34**	**609.33**	**599.82**	**593.11**	**624.94**

3.1. Resultados Obtidos

Os resultados sobre a qualidade da solução são obtidos usando a avaliação do *framework* BEELINE [Pratapa et al. 2020], que considera a área sob a curva de *precision-recall* (AUPRC) e a área sob os valores da curva característica de operação do receptor (AUROC) como métricas. Os melhores resultados são aqueles próximos de 1. A Figura 3 mostra *boxplots* dos resultados de AUPRC e AUROC obtidos pela CGP e métodos no *benchmark*.

As médias dos tempos de execução do método em sua forma sequencial e paralela são apresentados na Tabela 1. Estão disponíveis os tempos de execução dos programas como um todo (Prog.) e da evolução da CGP (Evol.). O ganho em tempo de processamento da técnica paralela apresentada em relação ao formato sequencial é de até 104%. É importante destacar que os métodos sequencial e paralelo de CGP alcançam os mesmos valores, uma vez que o paralelismo adotado aqui não altera o processo evolutivo.

4. Conclusões e Trabalhos Futuros

Como demonstrado pelos resultados, houveram ganhos significativos quanto ao tempo de execução da CGP e as GRNs foram inferidas com sucesso para os dados utilizados. Do ponto de vista prático, isso permite a inferência de GRNs maiores e mais complexas. Como trabalhos futuros, planeja-se melhorar o desempenho da técnica paralela apresentada, como as estruturas de dados envolvidas, e aplicá-la em dados relacionados a outros problemas, especialmente na saúde.

Referências

da Silva, J. E. H. et al. (2020). Inferring gene regulatory network models from time-series data using metaheuristics. In *2020 IEEE (CEC)*, pages 1–8. IEEE.

Delahaye-Duriez, A. et al. (2016). Rare and common epilepsies converge on a shared gene regulatory network providing opportunities for novel antiepileptic drug discovery. *Genome biology*, 17(1):1–18.

Dumitrescu, D., Lazzerini, B., Jain, L. C., and Dumitrescu, A. (2000). *Evolutionary computation*. CRC press.

Emmert-Streib, F. et al. (2014a). The gene regulatory network for breast cancer: integrated regulatory landscape of cancer hallmarks. *Frontiers in genetics*, 5:15.

Emmert-Streib, F. et al. (2014b). Grns and their applications: understanding biological and medical problems in terms of networks. *Front. in cell and devel. biology*, 2:38.

Khan, M. M., Khan, G. M., and Miller, J. F. (2010). Evolution of neural networks using cartesian genetic programming. In *Cong. on Evol. Comput. (CEC)*, pages 1–8. IEEE.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection*. Complex adaptive systems. MIT Press.

Ma, B. et al. (2019). Identification of gene regulatory networks by integrating genetic programming with particle filtering. *IEEE Access*, 7:113760–113770.

McCall, M. N. (2013). Estimation of gene regulatory networks. *Postdoc journal: a journal of postdoctoral research and postdoctoral affairs*, 1(1):60.

Medeiros, F. et al. (2019). Gene regulatory network inference and analysis of multidrug-resistant pseudomonas aeruginosa. *Memórias do Instituto Oswaldo Cruz*, 114.

Miller, J. F. (2011). Cartesian genetic programming. In *Cartesian Genetic Programming*, pages 17–34. Springer.

Miller, J. F. et al. (1999). An empirical study of the efficiency of learning boolean functions using a cartesian genetic programming approach. In *Proceedings of the genetic and evolutionary computation conference*, volume 2, pages 1135–1142.

Noman, N. and Iba, H. (2007). Inferring grns using differential evolution with local search heuristics. *IEEE/ACM Trans. on comp. biology and bioinfo.*, 4(4):634–647.

Palafox, L., Noman, N., and Iba, H. (2013). Reverse engineering of grns using dissipative particle swarm optimization. *IEEE Trans. on Evolutionary Comp.*, 17(4):577–587.

Pratapa, A. et al. (2020). Benchmarking algorithms for gene regulatory network inference from single-cell transcriptomic data. *Nature methods*, 17(2):147–154.

Qian, L., Wang, H., and Dougherty, E. R. (2008). Inference of noisy nonlinear differential equation models for gene regulatory networks using genetic programming and kalman filtering. *IEEE Transactions on Signal Processing*, 56(7):3327–3339.

Silva, B. M., Bernardino, H. S., and Barbosa, H. J. (2021). Human activity recognition using parallel cartesian genetic programming. In *CEC*, pages 474–481. IEEE.

Streichert, F., Planatscher, H., Spieth, C., Ulmer, H., and Zell, A. (2004). Comparing genetic programming and evolution strategies on inferring gene regulatory networks. In *Genetic and Evolutionary Computation Conference*, pages 471–480. Springer.

Van Der Wijst, M. G., de Vries, D. H., Brugge, H., Westra, H.-J., and Franke, L. (2018). An integrative approach for building personalized gene regulatory networks for precision medicine. *Genome medicine*, 10(1):1–15.