Aprendendo Histórias: Aprendizado Computacional para a Classificação de Hiper-histórias Metabólicas

Aluna: Larissa de Oliveira Penteado **Orientador:** Roberto Marcondes Cesar Jr
Coorientador: Ricardo Luiz de Andrade Abrantes
Departamento de Ciência da Computação, Instituto de Matemática e Estatística da Universidade de São Paulo

IME-USP

1. Introdução

Quando a levedura *S. cerevisiae* é exposta ao metal pesado cádmio (Cd^{2+}), esta sofre um estresse oxidativo, cujos mecanismos ainda não são completamente conhecidos [1]. Para modelar os dados provenientes deste experimento, utilizamos os conceitos de redes metabólicas que são o conjunto de reações que ocorrem no organismo; e de perfil metabolômico que apresenta a variação dos metabólitos entre dois estados (antes e depois do evento). A partir desta rede e perfil é possível obter hiper-histórias metabólicas que são um conjunto de reações que explicam o fluxo de metabólitos entre os estados. Quando aplicado a esta rede, em particular, o método adotado na pesquisa deste TCC devolve um número grande (aproximadamente 10^5 soluções) de hiper-histórias, o que torna difícil extrair informações manualmente destes dados. Diante disso, torna-se interessante a aplicação técnicas de aprendizagem de máquina que ajudem a agrupar as soluções para que seja possível melhor analisá-las e compreendê-las.

2. *Totoro* e dados

Totoro é um método para obtenção de hiper-histórias metabólicas que recebe como entrada uma rede metabólica e um perfil metabolômico. Neste caso, as redes são modeladas como hiper-grafos [2]. Neste trabalho, foram utilizados dois conjunto de hiper-histórias obtidas da rede da levedura a partir deste método. Um com 268 soluções e outro composto por 56 515 391.

3. Algoritmos de *clustering*

Algoritmos de *clustering* têm por objetivo o particionamento de um conjunto de dados (geralmente em k partes) de modo que cada parte possua alguma similaridade. Foram estudados e aplicados 4 métodos deste tipo neste trabalho, disponíveis na biblioteca *Scikit-Learn* [3] do Python. São eles:

K-Means: Neste algoritmo são mantidos um conjunto de k centróides que são calculados iterativamente até atingir-se a convergência. Cada ponto do conjunto de dados é atribuído ao subconjunto cujo centróide é o mais próximo dele [4].

Mini-Batch K-Means: Trata-se de uma versão modificada do *K-Means* para um conjunto grande de dados. Em que apenas um subconjunto deste de tamanho fixo é escolhido aleatoriamente a cada iteração para serem os pontos considerados para o cálculo de atualização dos centróides [5].

Affinity Propagation Neste algoritmo os pontos do conjunto de dados são tratados como nós em uma rede, entre os quais mensagens (números reais) podem ser transmitidos até que o conjunto de exemplares –que são o representantes de cada *cluster*– satisfaça um critério de similaridade pré-estabelecido [6].

Spectral Clustering Para particionar o conjunto de dados em k partes, é feita a transformação deste em um grafo G não-dirigido, com peso nas arestas, representado por uma matriz de afinidades. Os vértices deste grafo são os pontos do conjunto de dados, já as arestas são construídas seguindo alguma medida de similaridade ou distância. Para particionar o grafo, considera-se o problema do corte balanceado [7].

Para a escolha do número de partes k em que o conjunto de dados seria particionado, foi utilizada a heurística da silhueta [8]. Esta por meio de medidas de dissimilaridade média intra e entre *clusters*, indica o quão bom foi particionar o conjunto de dados em um certo número de partes. Para escolher k dentre um conjunto de valores, prefere-se o que obteve maior *score* de silhueta.

Como o conjunto de dados aqui tratado tinha vetores de características com dimensão grande, foram realizados testes com redução de dimensionalidade destes antes da aplicação do método de particionamento [9]. O algoritmo escolhido para tal tarefa foi o *Principal Component Analysis–PCA*.

4. *Pipeline* desenvolvido

O *pipeline* para tratamento das hiper-histórias, aplicação do algoritmo de *Clustering*, rearranjo dos dados e visualização, pode ser conferido na figura 1. Um arquivo que contém as soluções encontradas pelo *Totoro* –representadas pelos identificadores das reações que contêm– é a entrada do *pipeline*. Com isso, são montados os vetores de características binários para cada hiper-história. Em que cada posição do vetor indica a presença (*True*) ou não (*False*) de uma reação na hiper-história correspondente. Após este tratamento, é possível aplicar a redução de dimensionalidade por PCA. Então são executados os algoritmos de *clustering*. Assim para deixar as hiper-histórias pertencentes a um mesmo *cluster* organizadas de forma contígua na nossa matriz de dados, rearranja-se as linhas da matriz de acordo com seus rótulos, obtidos na etapa anterior. Por fim, é gerado um *heatmap* para a visualização do resultado.

4.1 Diagrama do *Pipeline*

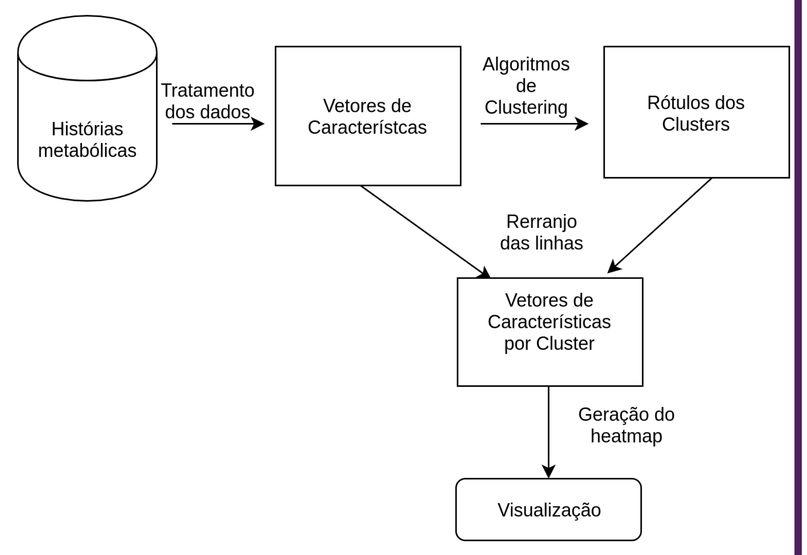
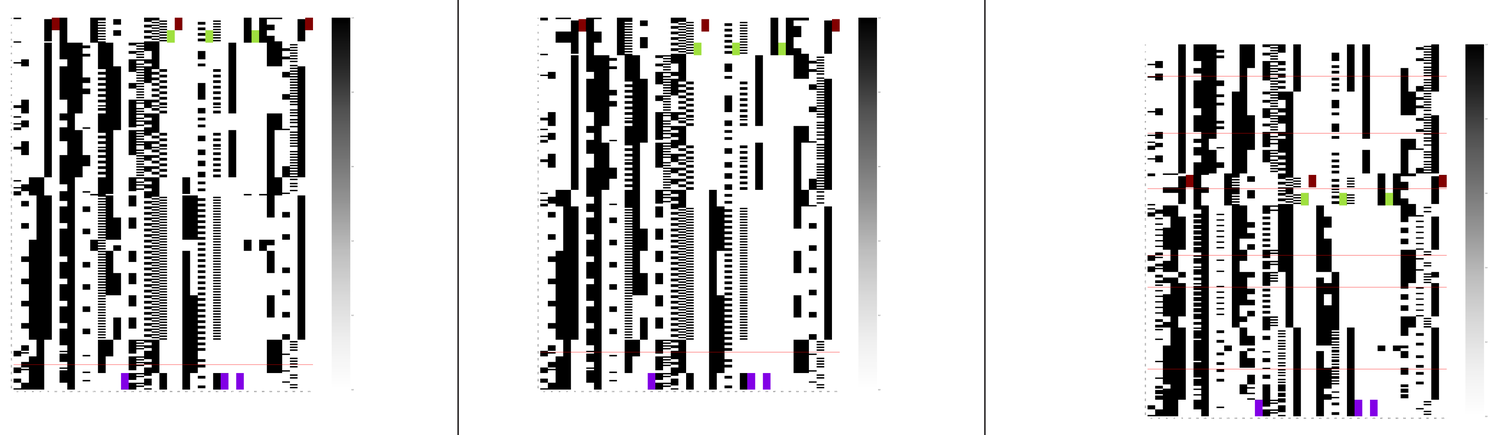


Figura 1: Diagrama do *pipeline* desenvolvido para o tratamento e particionamento das hiper-histórias

5. Resultados

Foi obtido um resultado interessante, em três algoritmos: *K-Means* sem PCA, *Mini-Batch K-Means* sem PCA e *Affinity Propagation* com PCA. Foi possível recuperar partições em que cada um dos seguintes conjuntos de reações: [102, 408, 575], [250, 459, 478], [396, 440, 503], estava contido completamente em um *cluster* da partição. Como é possível ver nas figuras da tabela 1 abaixo.



a) *K-Means* b) *Mini-Batch K-Means* c) *Affinity Propagation*

Tabela 1: Em a) temos o *heatmap K-Means* sem redução de dimensionalidade por PCA, k = 2. Em b) *heatmap Mini-Batch K-Means* sem redução de dimensionalidade por PCA, k = 2. E em c) *Heatmap Affinity Propagation* com redução de dimensionalidade por PCA (n = 5), k = 7

O primeiro subconjunto de reações está relacionado ao metabolismo do nicotinato, o segundo ao metabolismo do aminoácido purina e o terceiro ao PPP (*Pentose Phosphate Pathway*).

Para o segundo conjunto de dados, conseguimos obter um particionamento das hiper-histórias por meio do *Mini-Batch K-Means*, porém a análise do resultado obtido ainda não foi concluída, devido ao tamanho do conjunto de dados.

6. Conclusão

Sabe-se que tanto o metabolismo da purina como o do nicotinato/nicotinamida estão ligados à resposta ao estresse oxidativo. Isso mostra que o *pipeline* desenvolvido pode ajudar a compreender os resultados obtidos pelo *Totoro*, identificando grupos de hiper-histórias metabólicas ligadas à diferentes funções biológicas que podem explicar o fenômeno estudado.

7.Trabalhos Futuros

Em relação ao futuro, seria importante tentar interpretar os resultados obtidos com o conjunto maior de dados (composto por 56 515 391 hiper-histórias), e tentar aplicar técnicas mais eficientes de manipulação, *clustering* e visualização, do ponto de vista de memória e tempo, a este tipo de dado.

8. Referências

Para mais informações: https://www.linux.ime.usp.br/~larissadeop/Mac0499/

[1] Karin Vido, Daniel Spector, Gilles Lagniel, Sébastien Lopez, Michel B Toledano, and Jean Labarre. A proteome analysis of the cadmium response in saccharomyces cerevisiae. *Journal of Biological Chemistry*, 276(11):8469–8474, 2001.

[2] Alice Julien-Laferrière. *Models and algorithms applied to metabolism: From revealing the responses to perturbations towards the design of microbial consortia* . PhD thesis, Université Lyon 1 - Claude Bernard, 2016.

[3] `http://scikit-learn.org/stable/index.html`.

[4] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):881–892, 2002.

[5] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178. ACM, 2010.

[6] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[7] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, pages 849–856, 2002.

[8] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[9] B Dash, Debahuti Mishra, A Rath, and Milu Acharya. A hybridized k-means clustering approach for high dimensional dataset. *International Journal of Engineering, Science and Technology*, 2(2):59–66, 2010.

FAPESP