José Flávio de Souza Dias Júnior

Uma abordagem baseada em módulos para análise de perfis de expressão diferencial de genoma completo

Belém-PA, Brasil

Novembro, 2016

José Flávio de Souza Dias Júnior

Uma abordagem baseada em módulos para análise de perfis de expressão diferencial de genoma completo

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação. Instituto de Ciências Exatas e Naturais. Universidade Federal do Pará. Área de Concentração: Sistemas de Computação

Universidade Federal do Pará – UFPA

Instituto de Ciências Exatas e Naturais

Programa de Pós-Graduação em Ciência da Computação

Orientador: Ronnie Cley de Oliveira Alves

Belém-PA, Brasil

Novembro, 2016

José Flávio de Souza Dias Júnior
Uma abordagem baseada em módulos para análise de perfis de expressão diferencial de genoma completo/ José Flávio de Souza Dias Júnior. – Belém-PA, Brasil, Novembro, 2016-
64 p. : il. (algumas color.) ; 30 cm.

Orientador: Ronnie Cley de Oliveira Alves

Dissertação (Mestrado) – Universidade Federal do Pará – UFPA
Instituto de Ciências Exatas e Naturais
Programa de Pós-Graduação em Ciência da Computação, Novembro, 2016.

1. Transcriptograma. 2. Diferenciação. 3. Seriação. 4. Modularidade. 5. Claritate. I. Ronnie Cley de Oliveira Alves. II. Universidade Federal do Pará - UFPA. III. Instituto de Ciências Exatas e Naturais, Faculdade de Computação. IV. Mestrado.

José Flávio de Souza Dias Júnior

Uma abordagem baseada em módulos para análise de perfis de expressão diferencial de genoma completo

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação. Instituto de Ciências Exatas e Naturais. Universidade Federal do Pará. Área de Concentração: Sistemas de Computação

Belém-PA, Brasil, 23 de novembro de 2016:

Ronnie Cley de Oliveira Alves
Orientador

Claudomiro de Souza de Sales Junior
Convidado

Orlando Shigueo Ohashi Junior
Convidado

Regiane Silva Kawasaki Francês
Convidada

Rommel Thiago Juca Ramos
Convidado

Belém-PA, Brasil
Novembro, 2016

"Alia claritas solis,
alia claritas lunae et
alia claritas stellarum;
stella enim a stella differt in claritate."
(Bíblia Sagrada, 1Cor 15:41)

Agradecimentos

Agradeço ao Espírito Santo pelo conhecimento que se faz brilhar na esperança, na persistência e no compromisso com a humanidade.

À minha esposa, Amanda Dias, que reacendeu em mim o calor da juventude e expandiu o meu horizonte de possibilidades.

Aos meus pais, José Flávio e Cecília, e às minhas irmãs, Renata e Rafaela, pela energia das palavras e dos carinhos, essenciais para a manutenção dos passos desta longa caminhada.

Ao meu orientador, Prof. Ronnie, que acreditou neste trabalho e me conduziu magnificamente pelo universo da bioinformática.

Ao meu amigo Alex Vinson, pela espontânea e relevante tutoria.

Ao Instituto Evandro Chagas, que me apoiou da melhor forma possível desde o início deste desafio.

E a todos os parentes e amigos que pacientemente toleraram a minha ausência em alguns importantes instantes quando estive olhando para o “nada”.

Resumo

Transcrição é o processo de construção de RNA com base na cópia de uma sequência de gene. Esta cópia (mRNA), então, é traduzida para proteínas. Proteínas ditam o comportamento esperado no interior das células e são necessárias para a estrutura, função e regulação dos tecidos e orgãos do corpo. Juntas, transcrição e tradução são conhecidas como expressão gênica. Análise de perfis de expressão gênica, obtidos através de tecnologias tais como Microarray e RNA-Seq, tem revelado padrões sistêmicos e contribuído para a estruturação do conhecimento acerca da correlação entre genes e doenças. Muitos métodos estão sendo desenvolvidos para detectar e especificar marcadores de anormalidades biológicas. O transcriptograma é um desses métodos que, com base na seriação de rede protéica, constrói "fotografias" de transcriptomas mensurados com Microarray ou RNA-Seq. Neste trabalho, apresenta-se uma nova abordagem de análise de dados de expressão gênica, combinando métricas de diferenciação e transcriptograma para descobrir e explorar módulos de genes diferencialmente expressos entre amostras doentes e sadias. Um novo algoritmo (Claritate), sensível a redes livres de escala, foi desenvolvido para otimizar o processo de seriação de proteínas. É feito um estudo de caso com dados de pacientes com leucemia, obtendo-se, frente a abordagens tradicionais, resultados mais específicos e fortemente relacionados ao contexto biomédico. É realizada análise de sobrevivência de câncer e testes de cobertura são executados para enriquecimento de anotações funcionais.

Palavras-chave: Transcriptograma. Diferenciação. Seriação. Modularidade. Claritate.

Abstract

Transcription is the process of RNA building based on the copy of a gene sequence. This copy (mRNA) is then translated into proteins. Proteins dictate the expected behavior inside the cells and are necessary for the structure, function and regulation of tissues and organs of the body. Together, transcription and translation are known as gene expression. Analysis of gene expression profiles, obtained through technologies such as Microarray and RNA-Seq, has revealed systemic patterns and contributed to the structuring of the knowledge about the correlation between genes and diseases. Many methods are being developed to detect and specify biological abnormalities markers. The transcriptogram is one of these methods that, based on the seriation of protein network, plots "photographs" of transcriptomes measured with Microarray or RNA-Seq. In this work, a new approach is presented to gene expression data analysis, combining differentiation metrics and transcriptogram to discover and explore differentially expressed gene modules between diseased and healthy samples. A new algorithm (Claritate), sensitive to free scale networks, was developed to optimize the process of protein seriation. A case study is carried out with data from patients with leukemia, obtaining, in comparison to traditional approaches, more specific results and strongly related to the biomedical context. Cancer survival analysis is performed and coverage tests are done to enrich functional annotations.

Keywords: Transcriptogram. Differentiation. Seriation. Modularity. Claritate.

Lista de ilustrações

Lista de abreviaturas e siglas

ALL	Leucemia Linfoblástica Aguda
AML	Leucemia Mielóide Aguda
BP	Processo Biológico
CC	Componente Celular
cDNA	DNA complementar
CFM	Cost Function Method
DNA	Ácido Desoxirribonucleico
FDR	False Discovery Rate
GEO	Gene Expression Omnibus
GSEA	Gene Set Enrichment Analysis
GO	Gene Ontology Consortium
MF	Função Molecular
mRNA	RNA Mensageiro
NCBI	National Center for Biotechnology Information
RNA	Ácido Ribonucleico

Sumário

1 Introdução

Cada vez mais está claro que sistemas biológicos e redes de células são governadas por leis e princípios específicos, sobre os quais o entendimento será essencial para uma profunda compreensão da biologia (VIDAL; CUSICK; BARABÁSI, 2011).

Em organismos multicelulares, quase todas as células contém o mesmo genoma e, assim, os mesmos genes. No entanto, nem todos os genes estão transcricionalmente ativos em toda célula. Essas variações são a base da grande variedade das diferenças físicas, bioquímicas e de desenvolvimento vistas entre várias células e tecidos, e podem desempenhar um papel na distinção entre saúde e doença. Assim, ao estudar transcriptomas, pesquisadores esperam determinar quando e onde os genes estão ligados ou desligados em vários tipos de células e tecidos.

Como as interações protéicas são centrais na maioria dos processos biológicos, a identificação sistemática de todas as interações de proteínas é considerada primordial para descobrir os funcionamentos internos de uma célula (YOOK; OLTVAI; BARABÁSI, 2004).

Hernandez e Abel (2008) discutem que, classicamente, tem-se como dogma central da biologia molecular a síntese de proteínas através do processo de transcrição de DNA em RNA e, posteriormente, o processo de tradução de RNA em proteína. Renfrow et al. (2011) ressaltam ainda que o fluxo de informação do gene à proteína é um dos princípios fundamentais da biologia molecular. Esse processo de sintetização protéica, ilustrado pela Figura 1, é comumente chamado de *expressão gênica*, ou simplesmente expressão. E o conjunto completo de transcritos (RNA's) de um dado organismo, órgão, tecido ou linhagem celular é conhecido como *transcriptoma*.

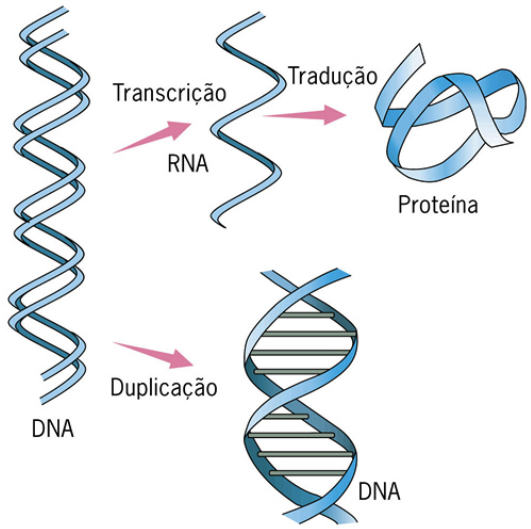

Figura 1: Ilustração do processo fundamental de expressão gênica.

Contudo, para que se compreenda esse ciclo de vida da proteína e suas funções no organismo, ou melhor, para se obter dados que dêem suporte à análise sistemática dessas redes protéicas, tem sido utilizadas várias tecnologias de extração de dados de organismos vivos, entre as quais se destacam o *microarray* e o *RNA-Seq* (HUNG; WENG, 2016).

Modelos mistos não supervisionados de algoritmos tem sido desenvolvidos para explorar perfis de expressão de genes. Contudo, eles requerem conhecimento prévio das frequências dos tipos de células dentro de um dado tecido, ou dos perfis de expressão de genes *in vitro* de cada tipo de componente celular. Na realidade, esta informação pode ser difícil de se obter e constitui uma grande desvantagem para estes tipos de abordagem. (KOUROU et al., 2015)

Rybarczyk-Filho et al. (2011) destacam ainda que dados de expressão de genomas completos consistem de níveis de expressão de milhares de genes e a análise conjunta de todos os dados representam um desafio.

Quaisquer dados não uniformes contém estruturas coesas devido à heterogeneidade dos dados. O processo de identificação dessas estruturas em termos de agrupamento de elementos é denominada *clusterização*, também chamada de *classificação de dados*. (KLEINBERG; TARDOS, 2002)

Algoritmos de *clusterização* são desenvolvidos para conjunto de dados bem grandes e complexos para uma análise manual (BERKHIN, 2006). Além disso, a *clusterização* é uma forma de *aprendizado não supervisionado*, pois não há conhecimento prévio sobre os objetos a classificar (ANDREOPOULOS et al., 2009).

Clusterização em bioinformática envolve dois grupos de usuários, ambos os quais precisam compreender as características algorítmicas que uma aplicação biológica requer. Um grupo inclui biólogos com experiência inerente ao problema biológico, que aplicam algoritmos de *clusterização* existentes para resolver o problema. O desafio é escolher um algoritmo adequado disponível em um software de análise, pois cada algoritmo produzirá resultados diferentes. O outro grupo de usuários inclui cientistas da computação que desenvolvem inovadores algoritmos de bioinformática, os quais auxiliam na detecção de padrões e no tratamento digital dos dados biológicos. (ANDREOPOULOS et al., 2009)

Este trabalho contempla os dois grupos de usuários, buscando mesclar soluções de bioinformática de tal forma a obter um novo método de análise de transcriptomas. Esta nova abordagem utiliza e potencializa conjuntamente os métodos de transcriptograma e de genes diferencialmente expressos (DEG), explorando uma análise módulo-funcional de perfis de expressão.

1.1 Contextualização

Como as interações, em que um dada proteína participa, estão provavelmente correlacionadas com as propriedades funcionais da proteína, mapas de interação protéica são frequentemente utilizados para descobrir sistematicamente o papel biológico em potencial de proteínas de classificação funcional desconhecida (YOOK; OLTVAI; BARABÁSI, 2004; PADHORNY et al., 2016).

Tentativas de integração, combinando mapas físicos de interação protéica com perfis de coexpressão, têm revelado que proteínas que se interagem são mais propensas a serem produzidas pelos genes com perfis de expressão similares do que proteínas sem interação (GE; WALHOUT; VIDAL, 2003). Além do aspecto fundamental de encontrar sobreposições significantes entre fronteiras de interação em redes protéicas e fronteiras de coexpressão em redes de perfis de transcrição, essas observações têm sido usadas para estimar a significância biológica global de conjunto de dados de interação protéica (VIDAL; CUSICK; BARABÁSI, 2011).

Essa estrutura de rede protéica pode ser especificada através de grafos e analisada com o suporte da clusterização.

Grafos são estruturas formadas por um conjunto de *vértices* (também chamados de *nós*) e um conjunto de *arestas*, que são conexões entre pares de vértices. *Clusterização de grafo* é a tarefa de agrupar os vértices do grafo em grupos, considerando a estrutura de arestas do grafo de tal forma que deveria ter muitas arestas dentro de cada grupo e relativamente poucas entre grupos. (SCHAEFFER, 2007)

Em mapas de rede de interação protéica, vértices representam proteínas e arestas representam uma interação física entre duas proteínas. As arestas não são orientadas, uma vez que não pode ser dito qual proteína se liga a outra, isto é, qual delas influencia funcionalmente a outra. (VIDAL; CUSICK; BARABÁSI, 2011)

Grafos fornecem liberdades adicionais sobre funções de distância. Em particular, é possível indicar através de um peso 0 que dois pontos não estão relacionados (LAARHOVEN; MARCHIORI, 2014). Uma rede protéica pode ser representada como uma matriz quadrada, onde um ponto diferente de zero significa interação de duas proteínas, e zero significa o contrário (ANDREOPOULOS et al., 2009).

Formalmente, dado um conjunto de dados, o objetivo da clusterização é dividir o conjunto de dados em grupos de tal modo que os elementos contidos em cada um deles sejam similares ou conectados por algum sentido predefinido. Contudo, nem todos os grafos possuem uma estrutura com grupos naturais. No entanto, um algoritmo de clusterização separa em grupos qualquer grafo de entrada. Se a estrutura do grafo é completamente uniforme, com as arestas uniformemente distribuídas sobre um conjunto de vértices, os grupos processados por qualquer algoritmo serão bastante arbitrários. (SCHAEFFER,

2007)

No caso do transcriptograma, o objetivo é clusterizar unidimensionalmente genes em interação, tal que a distância entre dois genes na lista se correlacione com a probabilidade de eles se interagirem, isto é, a probabilidade de que seus produtos protéicos estejam associados conforme bases de dados de associação protéica, como a STRING de Jensen et al. (2009). Uma vantagem é que a definição desses grupos é independente do estágio específico em que as células estão em um dado momento, ou do protocolo a que elas foram subordinadas. (RYBARCZYK-FILHO et al., 2011)

Com base nisso, perfis de expressão são normalizados sobre a lista ordenada de genes, gerando assinaturas transcricionais que permitem comparar e diferenciar experimentos distintos, possibilitando uma análise modular dos genes, isto é, uma visão sistemática de grupos de genes.

1.2 Motivação

As mudanças causais que conectam genótico a fenótipo permanecem geralmente desconhecidas, especialmente para complexos *loci* de traços e mutações associadas a doenças. Mesmo quando identificadas, muitas vezes não fica claro como uma mutação causal perturba a função do gene correspondente ou do produto do gene. Para "conectar os pontos" da revolução genômica, funções e contexto devem ser atribuídos ao grande número de mudanças genotípicas. (ROLLAND et al., 2014)

Enquanto esforços substanciais estão sendo feitos para gerar grandes conjuntos de dados *ômicos*, há uma crescente necessidade de desenvolver ferramentas para integrar esses dados e derivar modelos que descrevam interações biológicas. (SERIN et al., 2016)

Mapas de rede física de interação protéica podem evidenciar efetivamente listas de genes potencialmentes enriquecidos como novos candidatos a genes de doenças ou como genes modificadores de genes de doenças conhecidos. (VIDAL; CUSICK; BARABÁSI, 2011)

A análise por transcriptograma, que se baseia na seriação de redes protéicas, foi desenvolvida como uma solução para reduzir os ruídos nas técnicas de mensuração de transcriptoma, e tem demonstrado potencial para ser aplicado como um método para exploração de perfis de expressão gênica. (KUENTZER et al., 2014)

1.3 Justificativa

Slonim (2002) já apontava que projetos futuros podem focar na busca por conjuntos modestamente dimensionados de genes preditivos, caracterizando melhor a estrutura e o

poder de predição de dados de expressão gênica, ou combinar o conhecimento adquirido com a múltipla aplicação da clusterização.

Frequentemente, agrupam-se genes por coexpressão ou por covariação no tempo (ROMERO-CAMPERO et al., 2016; PINELLI et al., 2016). Redes especificadas por coexpressão são uma poderosa abordagem para acelerar a elucidação dos mecanismos moleculares subjacentes a importantes processos biológicos (SERIN et al., 2016).

A análise do transcriptoma sob várias condições experimentais tem provado que genes com um padrão geral de expressão semelhante têm frequentemente funções semelhantes. Consistentemente, os genes envolvidos na mesma via metabólica são encontrados em módulos coexpressos. (COMAN; RÜTIMANN; GRUISSEM, 2014)

Goh et al. (2007) afirmam que genes associados com distúrbios semelhantes mostram maior probabilidade de interações físicas entre os seus produtos e também maior similaridade de perfil de expressão entre os seus transcritos, apoiando a existência de distintos módulos funcionais para doenças específicas.

Existem muitos algoritmos que buscam grupos de vértices em redes complexas. Esses algoritmos têm sido aplicados com sucesso em redes de genes baseadas em interações protéicas. Contudo, entre eles, apenas o transcriptograma ordena genes numa lista, enquando os demais apresentam numa ordem arbitrária os genes que pertecem a um mesmo grupo. Uma lista aleatória de genes produz gráficos de níveis de expressão gênica relativa que flutuam tão grosseiramente que muito pouca, havendo alguma, informação pode ser colhida deles. (RYBARCZYK-FILHO et al., 2011)

O método de transcriptograma aumenta a relação sinal-ruído e dá uma visão geral do genoma inteiro, que pode ser especialmente útil para uma interpretação biológica de dados de expressão de genes. (SILVA et al., 2014)

Para ordenar os genes em linha, o transcriptograma utiliza o algoritmo Cost Function Method (CFM). Ele tem mostrado ser uma solução eficaz, mas sua execução exige um alto custo computacional. O trabalho de Kuentzer et al. (2014) procura medir a complexidade no tempo do CFM e melhorar o seu desempenho através de reformulação da estrutura de dados e com pequenos ajustes no ciclo de processamento. Entretanto, mesmo com as melhorias propostas, o CFM ainda continua consumindo muito tempo na sua execução. O desempenho algorítmico é importante devido ao grande e crescente volume de dados de experimentos com base em sequenciamento genético (WANG et al., 2016).

A aplicação de tecnologias de sequenciamento de nova geração tem produzido uma transformação nos estudos genômicos do câncer, gerando grandes conjuntos de dados que podem ser analisados em diferentes caminhos para responder a uma multidão de questões sobre alterações genômicas associadas a doenças. Como a nossa capacidade de produzir tais dados para múltiplos cânceres do mesmo tipo está melhorando, assim são as demandas

para analisar múltiplos genomas de tumores que crescem simultaneamente. (DING et al., 2010)

Além disso, Silva et al. (2014) observaram que o transcriptograma é aplicável a diversas técnicas de medição de expressão gênica de genoma completo, independente de plataforma ou tecnologia - ele certamente pode ser aplicado a dados de RNA-Seq (WANG; GERSTEIN; SNYDER, 2009), por exemplo. Essa flexibilidade amplia os horizontes de aplicação do transcriptograma, não o restringindo apenas ao escopo de *microarrays*. Mesmo assim, o transcriptograma não pode detectar assinaturas diferenciais de genomas completos entre classes de amostras, isto é, ele se limita à análise comparativa através de módulos de genes determinados sobre a rede protéica seriada, desconsiderando as modularizações específicas e diferencialmente expressas.

Este trabalho, portanto, busca otimizar, complementar e expandir as possibilidades analíticas do transcriptograma, apresentando um novo algoritmo de seriação (Claritate) e uma nova estratégia, baseada em módulos, de análise diferencial de perfis de expressão de genoma completo.

1.4 Contribuições

Submeteu-se um artigo com base neste trabalho, entitulado "A module-based approach for evaluating differential genome-wide expression profiles", à quinta edição da Brazilian Conference on Intelligent System (BRACIS), e obteve-se aceitação para publicação e apresentação oral. A BRACIS é um do mais importantes eventos no Brasil para pesquisadores no campo de inteligência artificial e computacional. Ela consiste da combinação do Simpósio Brasileiro de Inteligência Artificial (SBIA, 21 edições) com o Simpósio Brasileiro de Redes Neurais (SBRN, 12 edições).

Como suporte para este trabalho, foi desenvolvido um conjunto (*pipeline*) de ferramentas que podem ser executadas diretamente de um terminal de linha de comando. Ele cobre todas as etapas demonstradas na Figura 2, incluindo outras facilidades para exploração e visualização de *transcriptogramas*. As ferramentas foram implementadas nas linguagens de programação Java, R e C++, e estão disponíveis em https://github.com/joseflaviojr/transcriptograma/wiki.

Além disso, obteve-se junto ao Instituto Nacional da Propriedade Industrial (INPI) o registro de programa de computador "BR 51 2015 000155-8", correspondente à implementação de referência do algoritmo de seriação Claritate, conforme apresentado no Apêndice C.

Sendo assim, pode-se afirmar que as principais contribuições estão no escopo da bioinformática, quando da melhoria do processo de seriação de proteínas e da positiva

exploração modular de genes diferencialmente expressos.

1.5 Objetivos

1.6 Objetivo Geral

Propor uma nova estratégia computacional para análise de dados transcriptômicos usando a noção de transcriptograma e de módulos de genes diferencialmente expressos.

1.7 Objetivos Específicos

- Realizar um estudo abrangente sobre métodos computacionais para análise de dados transcriptômicos;
- Avaliar técnicas de clusterização em redes, principalmente redes biológicas;
- Definir uma nova e mais eficiente estratégia computacional (Claritate) para a seriação de proteínas;
- Executar testes comparativos entre o Claritate e o CFM, com foco na eficácia e na eficiência;
- Definir uma nova técnica de análise de genes diferencialmente expressos com base numa abordagem modular; e
- Realizar teste estatístico hipergeométrico para fins de enriquecimento funcional dos módulos de genes.

2 Referencial Teórico

2.1 Teoria dos Grafos

Um grafo consiste de um conjunto finito de vértices, um conjunto finito de arestas, e uma regra que diga quais arestas conectam pares de vértices. Normalmente, uma aresta conecta dois vértices distintos, mas excepcionalmente esses dois vértices podem coincidir; no último caso a aresta é chamada de laço. (BIGGS; LLOYD; WILSON, 1986)

Grafos são estruturas que generalizam e auxiliam a análise de caminhos e de grupos de vértices. No contexto deste trabalho, a detecção de agrupamentos (clusterização) é fundamental.

Schaeffer (2007) trabalha os detalhes qualitativos da clusterização de grafos e Vidal, Cusick e Barabási (2011) aplicam esta teoria no escopo das redes de interação protéica, especificando, inclusive, que vértices representam proteínas e arestas representam uma interação física entre duas proteínas.

2.2 Genes Diferencialmente Expressos - DEG

A análise de transcriptomas é uma importante ferramenta para caracterização e entendimento do perfil de expressão dos genes sob uma ou mais condições biológicas. (SONESON; DELORENZI, 2013)

Com o advento de tecnologias de mensuração do nível de expressão gênica, tais como *microarray* e *RNA-Seq*, tornou-se possível observar e comparar o nível de ativação de cada gene entre experimentos distintos.

Um conjunto de dados de expressão gênica contém medições de incremento ou decremento de níveis de expressão de um conjunto de genes. Um número de medições de expressão de genes são normalmente adquiridos entre pontos de tempo, amostras de tecidos ou pacientes. Sua representação é através de uma matriz de valores numéricos: gene por tempo, gene por tecido, gene por paciente. (ANDREOPOULOS et al., 2009)

De acordo com Baggerly et al. (2001), um dos principais objetivos de experimentos com *microarray* era, e continua sendo, determinar quais genes são diferencialmente expressos entre amostras, com base em técnicas estatísticas que normalizam e estipulam limiares de diferenciação, como explorado por Dudoit et al. (2002). Chen et al. (2002) utilizaram *microarray*, por exemplo, para estudar o câncer de fígado humano sob a perspectiva de distinguir padrões de expressão característicos de cada tipo tumor.

Nos últimos anos, segundo Serin et al. (2016), a *RNA-Seq* provou ser uma poderosa ferramenta para análise de perfil de transcriptoma completo, pois possui uma maior sensibilidade para a descoberta de novos transcritos. O poder destas tecnologias de sequenciamento tem permitido a análise por rede de coexpressão em espécies sem um genoma sequenciado e, como resultado, tem aberto o caminho para novas aplicações.

A análise por coexpressão gênica e por covariação no tempo é comum nas pesquisas com *RNA-Seq*, tanto que Soneson e Delorenzi (2013) comparam onze métodos para análise diferencial de dados de expressão gênica, ratificando inclusive o cuidado que se deve ter com a quantidade de amostras por classe, pois a maioria dos algoritmos perde a sensibilidade de diferenciação com pequenos conjuntos (menos de 5 amostras).

Portanto, os estudos baseados em experimentos por coexpressão e por diferenciação de expressão têm sido bastante relevantes para a compreensão dos sistemas biológicos e para o entendimento das redes e vias metabólicas.

2.3 Transcriptograma

Transcriptograma é um método para apresentar e analisar dados de transcrição numa escala de genoma completo que reduz ruído e facilita a interpretação biológica. (SILVA et al., 2014)

De acordo com Rybarczyk-Filho et al. (2011), o transcriptograma é uma ferramenta para análise de metabolismo celular, a qual é capaz de discriminar o estágio pelo qual a célula está passando em um determinado instante, como também apontar mudanças metabólicas em estados celulares alterados, em comparação a um estado de controle.

O transcriptograma, conforme retratado na Figura 2, utiliza técnica de seriação (ordenamento unidimensional) para organizar proteínas/genes ao longo de uma linha, mantendo cada proteína o mais próxima possível das proteínas com as quais ela interage mais fortemente. Esse algoritmo de seriação é denominado Cost Function Method (CFM).

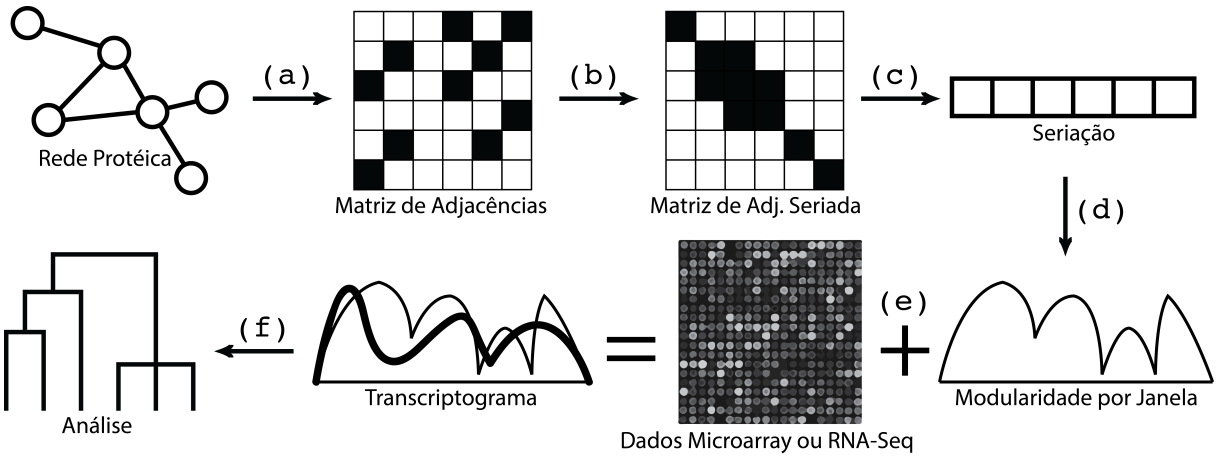

Figura 2: Processo geral da análise por transcriptograma. (a) O grafo que representa a rede protéica é transformado numa matriz de adjacências. (b) A matriz de adjacências é otimamente organizada por um algoritmo de seriação, comumente o CFM. (c) A seriação, ordem das proteínas, é extraída da matriz de adjacências. (d) A modularidade por janela é calculada com base na seriação da rede e seus módulos são evidenciados pelos picos. (e) Os dados de expressão (RNA-Seq or Microarray) são ajustados conforme a modularidade por janela, resultando no transcriptograma. (f) Análise módulo-diferencial dos perfis de expressão de genoma completo.

Dado o ordenamento, é montada uma matriz de adjacências do grafo representativo da rede protéica de tal forma que as colunas e as linhas respeitem a ordem da seriação. Ligações protéicas são representadas por valores 1 na matriz (pontos pretos). Um exemplo de matriz seriada é mostrado na Figura 3.

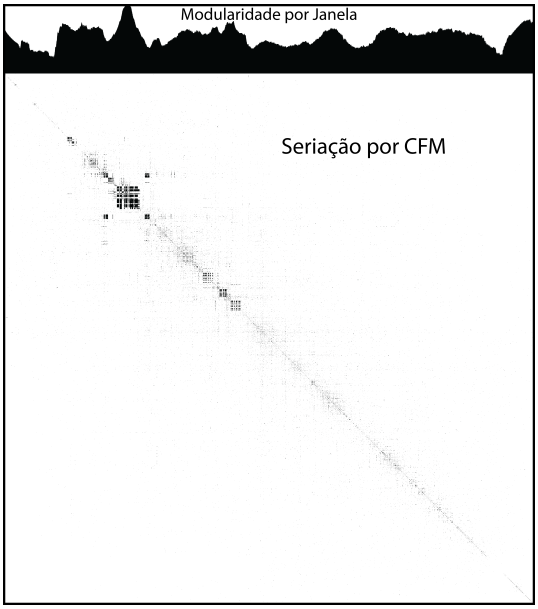

Figura 3: Matriz de adjacências (seriada com o algoritmo CFM) da rede protéica da *Saccharomyces cerevisiae*, especificada por Rybarczyk-Filho et al. (2011). No topo da figura consta a modularidade por janela, representação modular da matriz.

Observa-se na Figura 3 a formação de grupos ao longo da diagonal principal da matriz de adjacências, característica essa chamada de *diagonalização matricial* ou de diagonalização em blocos. Schaeffer (2007) promove uma detalhada discussão sobre *diagonalização matricial* no contexto da qualidade de clusterização.

O Cost Function Method (CFM) favorece a proximidade de genes em interação pela minimização de uma função custo E atribuída para cada ordenamento, dada como

$$E = \sum_{i=1}\sum_{j=1} d_{ij} \left\{ |M_{i,j} - M_{i+1,j}| + |M_{i,j} - M_{i-1,j}| + |M_{i,j} - M_{i,j+1}| + |M_{i,j} - M_{i,j-1}| \right\} \tag{2.1}$$

onde, $|..|$ mantém positivo o valor da diferença de elementos da matriz localizados na vizinhança e d_{ij} é proporcional à distância do ponto (i, j) até a diagonal, isto é, $d_{ij} = |i - j|$. Essa função custo aumenta com o número de interfaces entre elementos um e zero na matriz e aumenta mais ainda quando essas interfaces estão distantes da diagonal. (RYBARCZYK-FILHO et al., 2011)

Depois de iniciar com uma lista de genes desordenada e com sua matriz de interação correspondente, o algoritmo prossegue escolhendo aleatoriamente um par de genes e permutando suas posições no ordenamento. Uma nova matriz de interação é produzida para esse novo ordenamento e seu custo é recalculado usando a Equação 2.1. Se o custo é reduzido, a mudança é mantida. Se o custo é aumentado por ΔE, a mudança é aceita com

probabilidade $\exp[-\Delta E/T]$, onde T é uma temperatura virtual. (RYBARCZYK-FILHO et al., 2011)

Inicialmente, T é configurada como 0.01% do valor inicial da função custo E, sendo reduzida gradativamente num processo conhecido como *simulated annealing*, objetivando evitar estados metaestáveis. A simulação termina quando o número de permutações tem se estabilizado. (SILVA et al., 2014; BERTSIMAS; TSITSIKLIS, 1993)

Módulos estão presentes na forma de aglomerações de pontos pretos no entorno da diagonal da matriz de adjacências. Como essa análise visual não é boa o bastante para identificar módulos de interatividade, a identificação dos módulos é feita através de uma métrica chamada de modularidade por janela. (KUENTZER et al., 2014)

Essencialmente, módulos são grupos de genes adjacentes na seriação e, portanto, funcionalmente afins entre si. A proximidade entre módulos na seriação também denota uma afinidade funcional intermodular.

Rybarczyk-Filho et al. (2011) definem a modularidade por janela através da Equação 2.2. Para cada gene na seriação, considerar seus $w/2$ vizinhos à esquerda e seus $w/2$ vizinhos à direita, compreendendo um intervalo de $w+1$ genes. A modularidade por janela $W_w(i)$ para um gene, localizado na i-ésima posição da seriação, é definida como a razão entre o número de interações que ligam qualquer dois genes no intervalo (janela) de tamanho $w+1$, centralizado no i-ésimo gene, e o número de interações envolvendo no mínimo um gene naquela janela.

$$W_w(i) = \frac{1}{\sum_{j=1}^{N} M_{i,j}} \times \sum_{j=\ \mathrm{mod}\ (i-\frac{w}{2},N)}^{\mathrm{mod}\ (i+\frac{w}{2},N)} M_{i,j} \tag{2.2}$$

onde,

$$\mathrm{mod}\ (i+n,N) = \begin{cases} i+n & \text{if } i+n \leq N \\ i+n-N & \text{if } i+n > N \end{cases} \tag{2.3}$$

condiciona os limites regulares a lidar com genes próximos das extremidades da lista.

A escolha do tamanho da janela w depende da acurária desejada para os picos, sendo normalmente igual a 251, conforme analisado e especificado por (RYBARCZYK-FILHO et al., 2011). Assim, picos são formados e vales passam a separar módulos funcionais, como pode ser observado na Figura 4.

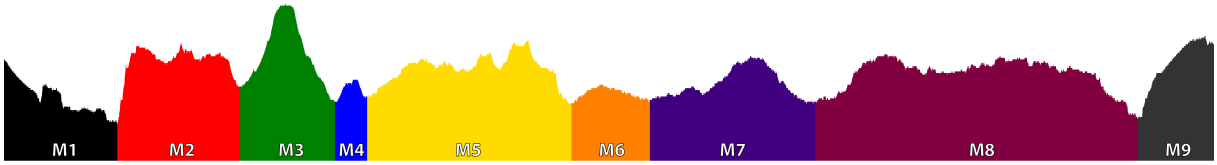

Figura 4: Modularidade por janela (tamanho $w = 251$) da rede protéica da *Saccharomyces cerevisiae*, especificada por Rybarczyk-Filho et al. (2011). Seriação com CFM. Os módulos funcionais foram numerados e separados por cores conforme a disposição dos vales e picos do gráfico. Horizontalmente, encontram-se os genes seriados e, verticalmente, o grau de interatividade do gene dentro da janela.

A modularidade por janela, construída com base na seriação de uma rede protéica de referência, serve como um guia para análise comparativa entre experimentos baseados na medição de expressão gênica. Tomando como exemplo o experimento GSE3431 (disponível no banco de dados NCBI/GEO) de Tu et al. (2005), que consiste em estudo do ciclo metabólico da *Saccharomyces cerevisiae*, podemos verificar, através da Figura 5, a aplicação do transcriptograma como ferramenta de construção de assinaturas transcrionais para fins comparativos.

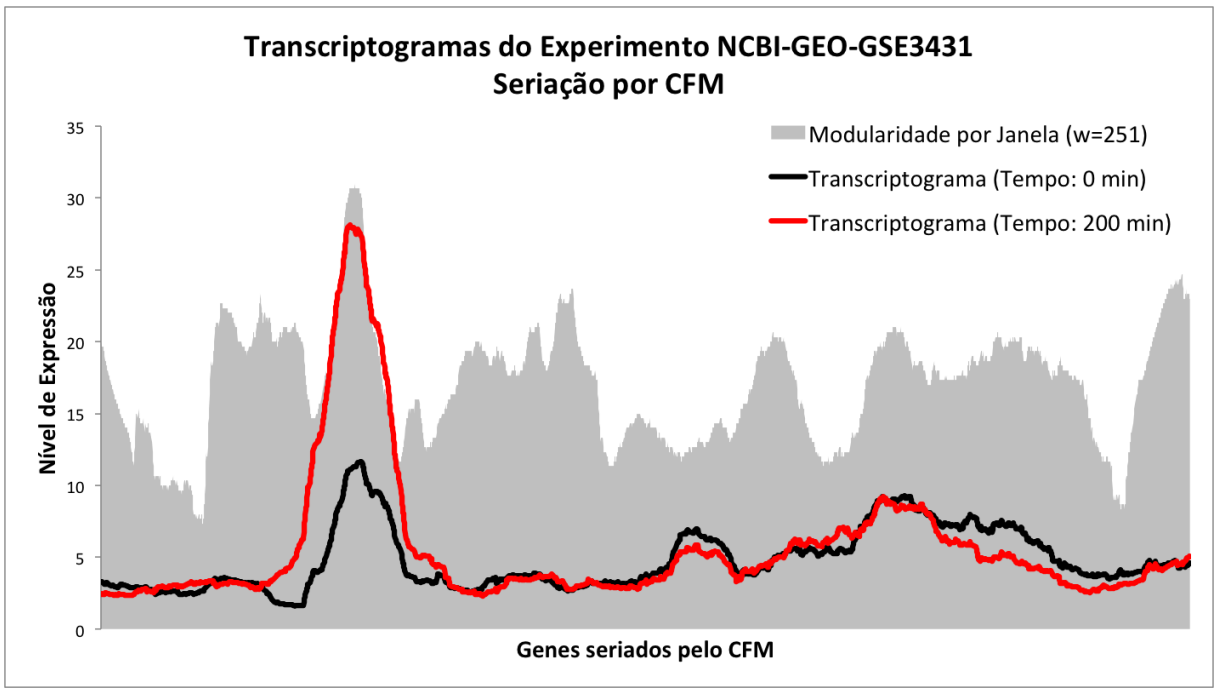

Figura 5: Transcriptogramas das fases 0 (preto) e 200 minutos (vermelho) do experimento GSE3431. No eixo horizontal estão os genes seriados e no eixo vertical está o nível de expressão medido e normalizado pelo experimento. Ao fundo, está a modularidade por janela da Figura 4.

As curvas (transcriptogramas) da Figura 5 são resultados da aplicação da Equa-

ção 2.2 sobre os dados de expressão gênica. De posse delas, é possível observar uma variação dos níveis de expressão de alguns módulos no intervalo de tempo de 200 minutos. Uma análise temporal dos módulos é apresentada na Figura 6, na qual o nível de expressão de cada módulo é definido através do nível de expressão do seu gene central, pois este valor consiste numa espécie de média da janela que envolve o módulo.

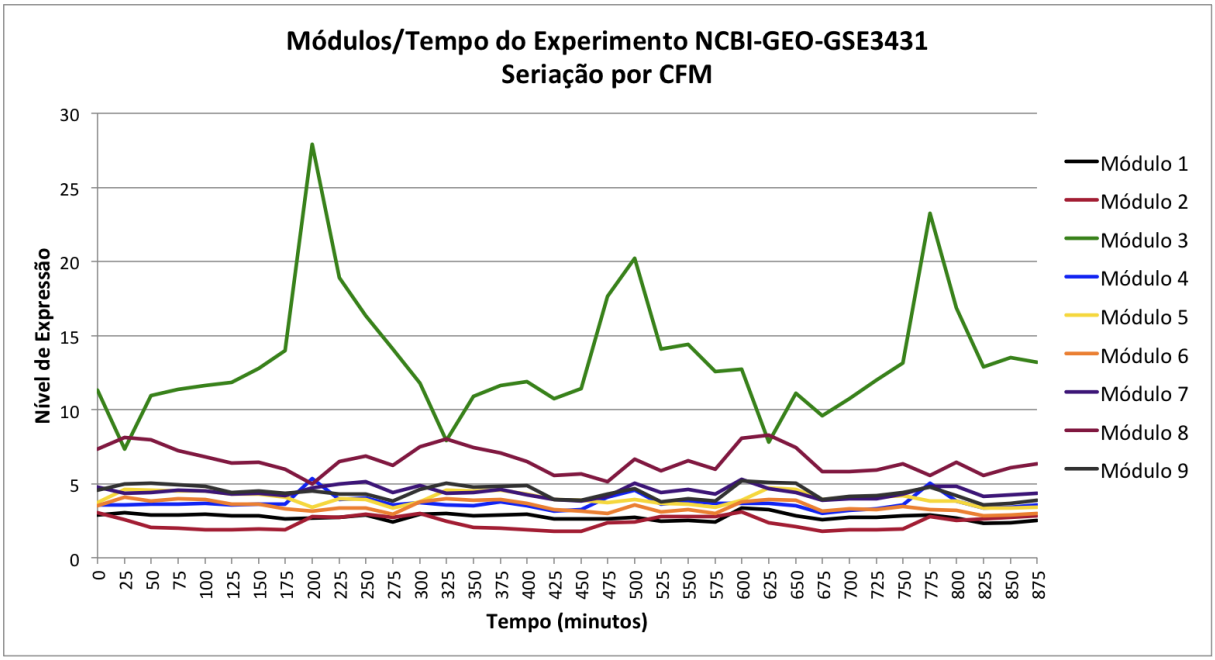

Figura 6: Nível de expressão dos módulos em função do tempo (experimento GSE3431).

2.4 Enriquecimento Funcional

O ritmo cada vez maior em que genomas estão sendo sequenciados abre uma nova área de pesquisa, a genômica funcional, que está preocupada com a atribuição de função biológica a sequências de DNA (DUDOIT et al., 2002; SERIN et al., 2016; PINELLI et al., 2016).

Para verificar a afinidade funcional entre genes, pode-se utilizar a técnica de enriquecimento funcional, disponível originalmente pela ferramenta AmiGO/Term Enrichment Service de (CARBON et al., 2009), pertencente ao projeto Gene Ontology Consortium (GO) de (CONSORTIUM, 2015). Esta ferramenta identifica as funções metabólicas mais representativas de um grupo de genes (módulo), estipulando *p-values* através de teste estatístico hipergeométrico sobre um domínio de relações funcionais ontologicamente organizadas.

Esse tipo de análise é compatível com o contexto dos transcriptogramas, pois as modularidades por janela são agrupamentos de genes definidos através de seriação de

mapas de relações funcionais laboriatorialmente constatadas ou empiricamente deduzidas.

Assim, tomando como exemplo a modularidade por janela apresentada na Figura 4 da Seção 2.3, realizou-se enriquecimento funcional conforme a técnica de Carbon et al. (2009), considerando apenas *p-values* menores que 10^{-3} e corrigindo-os através do método FDR de Benjamini e Hochberg (1995). O resultado é demonstrado no formato de *heat map* na Figura 7, associando os módulos a funções moleculares da ontologia "Biological Process" (BP).

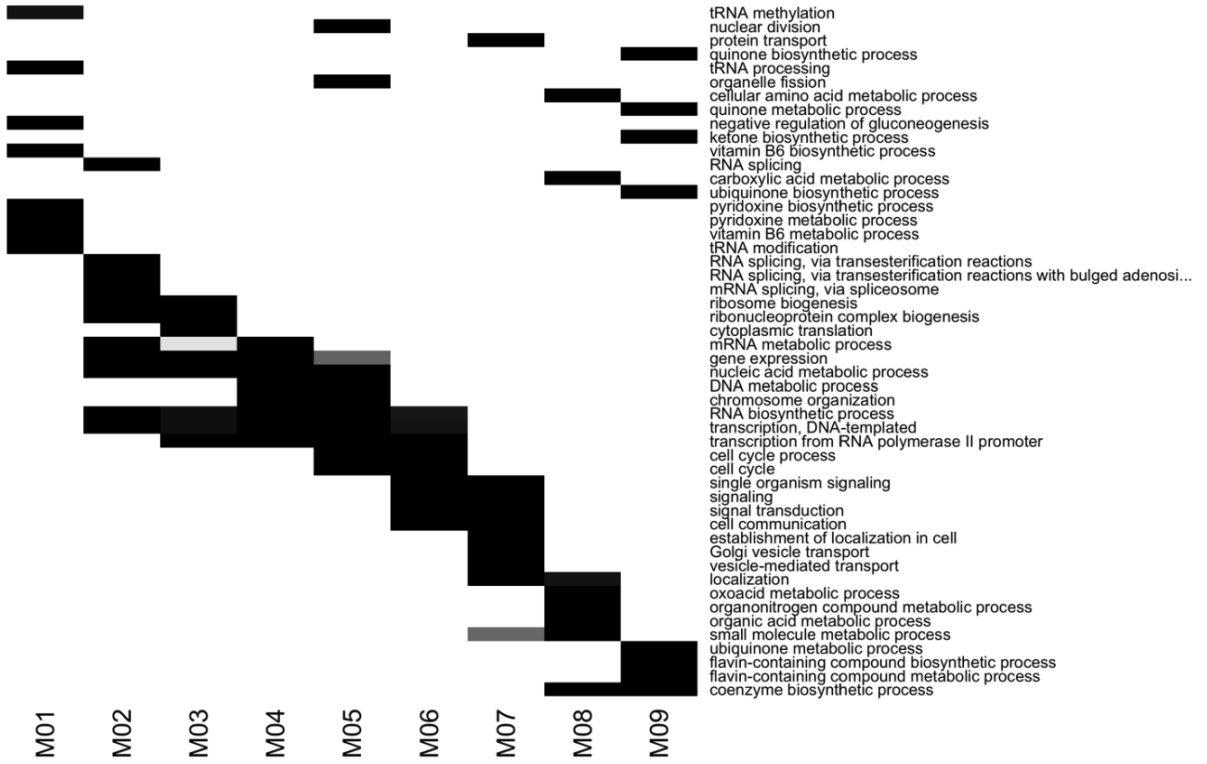

Figura 7: *Heat map* associando funções do Gene Ontology (Biological Process) com os módulos da rede de interação protéica de Rybarczyk-Filho et al. (2011), demonstrada na Figura 4. As cores do *heat map* correspondem aos *p-values* corrigidos por FDR do teste hipergeométrico, e seguem o padrão escala de cinza, no qual quanto mais escura a cor, melhor é o valor do *p-value*.

É possível perceber facilmente na Figura 7, por exemplo, que o módulo 1 ($M1$) está fortemente associado a processos biológicos relacionados à Vitamina B6/Piridoxina.

Outro método eficaz de enriquecimento funcional é a Gene Set Enrichment Analysis (GSEA) de Mootha et al. (2003), que determina se os membros de um dado conjunto de genes estão enriquecidos entre os genes mais diferencialmente expressos. De acordo com Subramanian et al. (2005), a GSEA provém o seu pontecial pelo foco em conjuntos de genes, isto é, grupos de genes que compartilham função biológica, localização cromossômica ou regulação.

A GSEA considera experimentos com perfis de expressão de genomas completos de amostras pertencentes a duas classes. Genes são classificados com base na correlação entre sua expressão e a distinção de classe, conforme alguma métrica adequada. Assim, os genes são ordenados numa lista L, de acordo com a expressão diferencial entre as classes. O desafio é extrair significado desta lista. (SUBRAMANIAN et al., 2005)

Dado um conjunto S de genes definidos a priori (por exemplo, genes que codificam produtos numa via metabólica, localizados numa mesma banda citogenética ou que partilham a mesma categoria GO), o objetivo da GSEA é determinar se os membros de S são aleatoriamente distribuídos ao longo da lista L ou se encontram principalmente no ínicio ou no fim da L.

3 Metodologia

Este trabalho consiste na detecção e na análise de módulos de genes diferencialmente expressos.

Dados de expressão gênica obtidos com RNA-Seq de três classes de pacientes (saudáveis, leucemia ALL e leucemia AML) foram submetidos à classificação diferencial (DEG) e, posteriormente, ordenados conforme a seriação de uma rede protéica que passou por uma curadoria de alta qualidade.

Por conseguinte, módulos extraídos da seriação dos níveis de diferenciação gênica foram enriquecidos funcionalmente, identificando-se as funções metabólicas potencialmente irregulares entre pacientes doentes e saudáveis.

Além disso, foi realizada a projeção de transcriptogramas dos níveis de diferenciação dos genes seriados, a fim de comparar as assinaturas diferenciais dos pacientes ALL com as dos pacientes AML.

3.1 Dados RNA-Seq

Dados reais de RNA-Seq, utilizados no estudo de Macrae et al. (2013), foram selecionados para explorar perfis de expressão gênica diferencial, os quais estão disponíveis *on-line* no banco de dados Gene Expression Omnibus (GEO), através do código de acesso GEO Series GSE48173. Os dados compreendem 72 amostras de pacientes, sequenciadas em Illumina HiSeq 2000 (*Homo sapiens*) e classificadas como a seguir: 43 Leucemia Mielóide Aguda (AML), 12 Leucemia Linfoblástica Aguda (ALL) e 17 saudáveis.

3.2 Rede Protéica de Alta Qualidade

Transcriptogramas têm sido concebidos unicamente com base em redes protéicas do banco de dados STRING de Jensen et al. (2009). Embora ele possa mapear um catálogo maior de genes, isso não implica em ter informação com curadoria.

Portanto, neste estudo em particular, utilizou-se a rede protéica humana introduzida por Rolland et al. (2014), denominada HI-II-14, a qual corresponde a um mapa sistemático de 4.303 proteínas e 13.944 interações. O mapa também revela significativa interconectividade entre conhecidos e candidatos produtos gênicos associados a câncer, fornecendo evidências sem viés para uma eficaz exploração módulo-funcional de redes protéicas.

Ressalta-se que a rede HI-II-14 segue a topologia livre de escala, padrão estrutural entre redes biológicas, resultado de estudo de Barabási e Albert (1999).

3.3 Seriação

Neste trabalho, propõe-se uma alternativa ao tradicional Cost Function Method (CFM), realizando o processo de seriação com um novo algoritmo, denominado Claritate.

O Claritate é um algoritmo metaheurístico para agrupamento unidimensional de vértices afins em grafos. Ele utiliza uma estratégia de proporção espacial da distância real entre as proteínas na lista ordenada (seriação) com a distância mínima virtual observada na estrutura de grafo representante da rede protéica, conforme ilustrada na Figura 8.

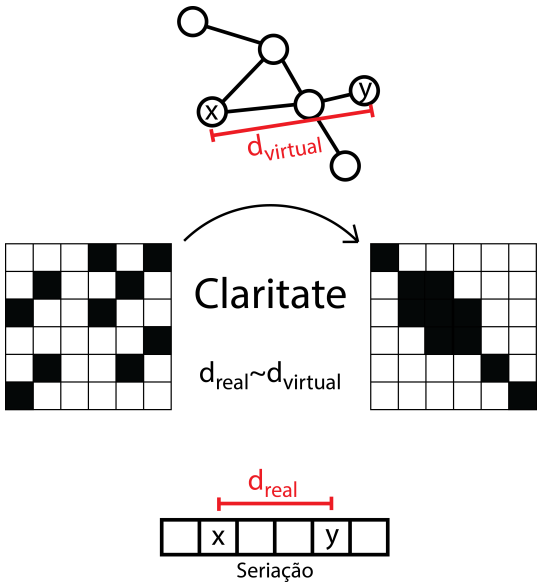

Figura 8: Esquema básico de funcionamento do algoritmo de seriação Claritate.

A sequência de etapas do Claritate está descrita a seguir, sendo que uma implementação de referência em C++ se encontra no Apêndice C.

1. Representar grafo da rede protéica na forma de matriz de adjacências (MA).
2. Gerar matriz de distâncias mínimas (MD) entre todos os vértices conforme algoritmo Floyd-Warshall (FLOYD, 1962).
3. Calcular excentricidade de cada vértice. Segundo Hage (1995), a *excentricidade* $e(v)$ de um vértice v num grafo conexo G é a máxima distância $d(v, u)$ para todo u.
4. Ordenar crescentemente os vértices pelo inverso da excentricidade, normalmente entitulado centralidade, estabelecendo assim o primeiro vetor-resultado (VR) parcial, que consiste na seriação das proteínas/genes.

5. Reposicionar cada vértice no VR de acordo com a média das posições de seus filhos[1].
6. Calcular dispersão inicial de acordo com a MA ordenada pelo VR. A dispersão é discutida adiante.
7. Sortear 3 posições sequenciais do VR: S1, S2 e S3.
8. Sortear um valor entre 0 e 100, denominado pivô.
9. Se o pivô for <= 33, reposicionar S1 no VR; se pivô <= 66, reposicionar S2; caso contrário, S3. O deslocamento é discutido adiante.
10. Calcular nova dispersão conforme novo arranjo do VR.
11. Se a nova dispersão for maior ou igual à dispersão anterior, reverter o reposicionamento realizado no VR; caso contrário, assumir a nova dispersão e o novo VR como o melhor resultado até este momento.
12. Executar a partir de 7 até que a dispersão desejada seja alcançada.

A pré-seriação do Claritate, baseada nos conceitos de excentricidade e centralidade de Hage (1995), justifica-se pela sua forte sensibilidade na detecção de *hubs* em redes livres de escala (BARABáSI; ALBERT, 1999).

Dispersão é o nome dado à métrica utilizada pelo Claritate para determinar a qualidade do agrupamento unidimensional do grafo, técnica definida em Schaeffer (2007). Seu cálculo é realizado sobre a matriz de adjacências MA ordenada conforme o vetor-resultado VR. Entende-se por ordenamento a permutação das colunas e linhas da MA de tal forma que a posição delas estejam conforme sua posição dentro do VR.

O objetivo da dispersão é orientar o Claritate no deslocamento das arestas (elementos não zeros, pontos pretos) para as proximidades da diagonal principal da MA ordenada. A Equação 3.1 especifica o cálculo de dispersão para MA de grafos orientados.

$$Dispersão = \sum_{i=1}^{n} \sum_{j=1}^{n} diagonal(m_{i,j}) \tag{3.1}$$

onde,

- n : quantidade de vértices do grafo (ordem da MA)
- $m_{i,j}$: valor do elemento (i, j) da MA ordenada, isto é, da MA com colunas e linhas permutadas conforme VR

[1] Em grafos orientados, filho será todo vértice com aresta proveniente da referência; em grafos não orientados, filho será todo vértice adjacente com grau menor ou igual ao grau da referência.

- $diagonal(...)$: função de $m_{i,j} = \begin{cases} 0 & m_{i,j} = 0 \\ |i-j| & \cdots \end{cases}$

Para grafos não orientados, o cálculo da dispersão é simplificado para apenas metade da MA ordenada, como especificado pela Equação 3.2.

$$Dispersão = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} diagonal(m_{i,j}) \tag{3.2}$$

A convergência do algoritmo para a solução se dá gradativamente através de deslocamentos de vértices dentro do VR. A escolha do vértice a ser deslocado em cada ciclo do Claritate é feita através de 4 sorteios consecutivos: os 3 primeiros determinam os vértices candidatos e o último, denominado pivô, determina qual dos 3 vértices será deslocado.

O deslocamento do vértice escolhido é realizado conforme a proporção entre as distâncias reais e as desejadas entre ele e os outros 2 vértices candidatos. Compreende-se por distância desejada o valor especificado na matriz de distâncias MD entre o escolhido e os demais candidatos; enquanto que distância real é a diferença das posições no VR dos vértices em questão. A Equação 3.3 especifica o cálculo de deslocamento.

$$\begin{aligned} w_{1\leftrightarrow 2} &= \frac{d_{1\leftrightarrow 2}}{d_{1\leftrightarrow 3}} \times r_{1\leftrightarrow 3} \\ w_{2\leftrightarrow 3} &= r_{1\leftrightarrow 3} - w_{1\leftrightarrow 2} \end{aligned} \tag{3.3}$$

onde,

- 1, 2 e 3 : vértices sorteados para o cálculo do deslocamento, os quais estão dispostos sequencialmente conforme disposição no VR
- $r_{1\leftrightarrow 3}$: distância real entre os vértices 1 e 3, correspondente à diferença de suas posições no VR
- $w_{1\leftrightarrow 2}$: nova distância real entre os vértices 1 e 2
- $w_{2\leftrightarrow 3}$: nova distância real entre os vértices 2 e 3
- $d_{1\leftrightarrow 2}$: distância desejada entre os vértices 1 e 2 conforme especificação na MD
- $d_{1\leftrightarrow 3}$: distância desejada entre os vértices 1 e 3 conforme especificação na MD

Ainda não se estabeleceu uma determinística condição de parada para o Claritate, pois as estruturas de redes protéicas são complexas e não uniformes, o que dificulta uma delimitação do estágio de completude ou de aproximação do resultado ótimo.

Além disso, de acordo com Rybarczyk-Filho et al. (2011), o ordenamento de genes em linha é um processo frustado, no sentido que conflitos aparecem sobre como ordenar genes. Pode acontecer que um gene interaja com dois diferentes grupos, o qual pode ser posicionado perto de qualquer um dos grupos ou em algum lugar entre eles.

Em algumas aplicações, pode ser que não seja viável o esforço de buscar computacionalmente a melhor solução para o problema em questão, mas uma solução aproximada será suficiente. Sempre que a computação exata é demorada, impossível, ou simplesmente não é justificada pelas necessidades da aplicação, métodos aproximados e *heurísticos* são úteis. Muitos desses métodos fornecem uma saída *não determinística*, isto é, o método pode apresentar uma solução diferente em cada execução. Contudo, pode-se precisar executar repetidamente um algoritmo heurístico e então filtrar as saídas que respeitam alguma métrica de qualidade. (SCHAEFFER, 2007)

No caso do Claritate, o ciclo de vida compreende 3 fases, sendo a última do tipo metaheurística - inicialização, dispersão e compressão. Na inicialização, realiza-se um conjunto de cálculos que determinam os seguintes valores:

- Matriz de distâncias mínimas entre todos os vértices conforme algoritmo Floyd-Warshall (FLOYD, 1962).
- Excentricidade de cada vértice, conforme especificação de Hage (1995).
- Identificação do tipo de grafo: orientado ou não orientado.
- Grau de entrada de cada vértice.

Todos esses cálculos são realizados dentro do fluxo de execução do Floyd-Warshall, que possui complexidade no tempo $O(n^3)$, onde n corresponde ao total de vértices do grafo.

Na fase de dispersão, é realizada uma clusterização preliminar de acordo com a centralidade (inverso das excentricidades) dos vértices. Para isso, executa-se primeiro o algoritmo de classificação Quicksort (HOARE, 1961) para as centralidades, que possui complexidade, no caso médio, de $O(n \log n)$. Por último, reposicionamentos de vértices são realizados, consumindo um tempo de $O(n^2)$.

Por conseguinte, o algoritmo entra na fase de compressão, a qual consiste de um processo metaheurístico de otimização da solução. São executados p passos, valor este ainda em pesquisa, pois depende uma condição de parada. Em cada passo, executa-se o cálculo de dispersão, que possui complexidade $O(n^2)$ para grafos orientados e $O(n^2/2)$ para grafos não orientados. Além disso, são executados um ou dois deslocamentos de vértices, cada um consumindo $O(n)$ no pior caso.

Portanto, a complexidade no tempo do algoritmo Claritate é $O(n^3 + n \log n + n^2 + p \times (n^2 + 2n))$, isto é, assintoticamente, $O(n^3 + p \times n^2)$.

3.4 Comparação entre CFM e Claritate

Com a finalidade de comparar a eficácia dos algoritmos CFM e Claritate, escolheu-se como ponto de referência a rede protéica da *Saccharomyces cerevisiae*, especificada por Rybarczyk-Filho et al. (2011), com base nos dados do projeto STRING, versão 8 (JENSEN et al., 2009).

Conforme Schaeffer (2007), no contexto de avaliação da qualidade da clusterização, a visualização por matriz de adjacências ajuda a revelar a presença de grupos densos. Matematicamente, isso pode ser alcançado, por exemplo, através do cálculo da distância de cada elemento que tem o valor um até a diagonal da matriz de adjacências - quanto menor o valor, melhor a clusterização. A diagonalização em blocos tem sido utilizada em relação à clusterização por Schaeffer (2006) e Carrasco et al. (2003).

Como os dois algoritmos se orientam essencialmente pela aglomeração de pontos ao longo da diagonal principal da matriz de adjacências da rede, tem sido utilizada como métrica de eficácia o conceito matemático de diagonalização em blocos de matrizes, denominada particularmente de dispersão.

Além disso, especificou-se um cenário de testes para realizar uma análise mais apurada, com base na diversidade, especificidade e aleatoriedade, procurando evidenciar a eficiência dos algoritmos.

A mensuração da eficiência tem sido feita através da análise de tempo de execução dos algoritmos de seriação em questão, considerando, como condição de parada, um tempo limite de execução suficiente para atingir estágios predeterminados de qualidade de clusterização obtidos com a métrica de eficácia, pois, de acordo com Schaeffer (2007), como o algoritmo de aproximação não objetiva encontrar a melhor solução possível, mas sim uma que não esteja longe de ser a melhor, deve-se apresentar limites demonstráveis sobre o quão longe a solução aproximada pode estar da solução ótima. Com base nisso, implementou-se os algoritmos na mesma linguagem de programação (C++), utilizando recursos computacionais otimizados e adaptados para registrar os estágios de execução para fins estatísticos e comparativos, como exposto no Apêndice C e Apêndice D.

A partir de um ponto de vista mais prático, enquanto o algoritmo está executando, deve-se registrar o melhor estado encontrado até agora e o seu custo associado (BERTSIMAS; TSITSIKLIS, 1993). Com base nisso, também tem sido realizados testes de acurácia máxima, quando os algoritmos são submetidos a longos tempos de execução e se registra o melhor nível de convergência em função do tempo, ou seja, tem sido medido o alcance

quantitativo da eficácia de cada algoritmo.

3.5 Cálculo dos Genes Diferencialmente Expressos (DEG)

Para calcular a diferenciação de expressão entre as classes de pacientes, escolheu-se o teste estatístico Welch t, apropriado para o contexto de classes com médias desiguais (DERRICK; TOHER; WHITE, 2016; AHAD; YAHAYA, 2014; JEANMOUGIN et al., 2010). Utilizou-se a implementação em linguagem R da biblioteca *GeneSelector* (BOULESTEIX; SLAWSKI, 2009), que disponibiliza vários testes estatísticos para diferenciação. Como resultado, tem-se a lista de genes enumerada conforme o nível de diferenciação detectado.

3.6 Enriquecimento Funcional

Para identificar funções biológicas associadas aos módulos de genes, utilizou-se a técnica de enriquecimento funcional disponível através da ferramenta AmiGO/Term Enrichment Service de Carbon et al. (2009), do projeto Gene Ontology Consortium (GO) de Consortium (2015), considerando apenas os resultados com *p-values* menores que 10^{-3} e ajustando-os através do método FDR de Benjamini e Hochberg (1995).

4 Resultados e Discussão

Primeiramente, a fim de comparar os algoritmos de seriação, executou-se o Claritate sobre a rede protéica da *Saccharomyces cerevisiae*, especificada e seriada com CFM por Rybarczyk-Filho et al. (2011), a qual contém 4.655 proteínas e 47.415 ligações. A Figura 9 apresenta a comparação na forma de matriz de adjacências e modularidade por janela. Nos apêndices A e B se encontram transcriptogramas de experimentos que evidenciam semelhanças entre os resultados dos algoritmos.

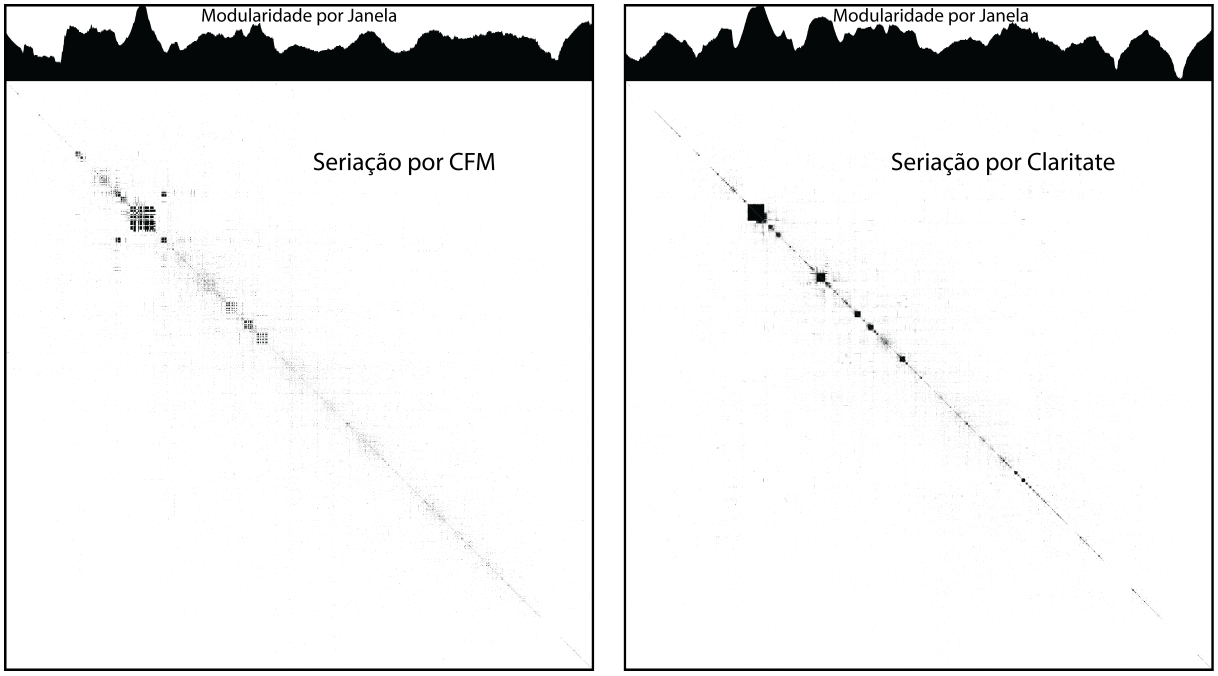

Figura 9: Matrizes de adjacências e modularidades por janela da rede protéica da *Saccharomyces cerevisiae*, especificada por Rybarczyk-Filho et al. (2011). (a) Resultado da seriação por CFM. (b) Resultado da seriação por Claritate.

A dispersão aleatória inicial da matriz de adjacências era de 73.966.533 unidades, sendo que o CFM alcançou uma dispersão final de 11.659.741 (84,23% de redução) em 3.000 ciclos de Monte Carlo, enquanto que o Claritate chegou a 12.777.894 (82,72%) em 72 horas de execução. Observa-se que o Claritate privilegiou a aglomeração local de proteínas (*hubs*) em detrimento à disposição global das arestas (pontos pretos), isso devido à pré-seriação baseada em centralidade de Hage (1995), sensível a redes livres de escala (BARABáSI; ALBERT, 1999), o que o levou a ter um resultado na acurácia levemente pior neste caso (1,51% de diferença). Ressalta-se, porém, que o Claritate alcançou 62,72% de redução em apenas 1 hora, 74,61% em 2 horas e ultrapassou os 80% nas primeiras 5 horas de execução.

Contudo, considerando o cenário de testes da Seção 3.4, especificado a seguir:

- 32 redes livres de escala (topologia biológica), geradas aleatoriamente, sendo 4 redes para cada tamanho diferente: 100, 200, 300, 400, 500, 600, 700 e 800 vértices.
- Adaptou-se os algoritmos para registrarem custo, tempo da última melhoria, percentagem de redução do custo inicial, quantidade de chamadas à função custo, quantidade de permutações, quantidade de permutações desfeitas e quantidade de melhorias detectadas.
- Cada algoritmo foi executado 2 vezes para cada uma das 32 redes geradas, totalizando 128 execuções.
- A dispersão foi escolhida como métrica de qualidade para fins de comparação, pois ela consiste exatamente na lógica descrita por Schaeffer (2007) como alternativa para diagonalização matricial, técnica de avaliação de clusterização.
- A condição de parada para cada execução foi determinada por tempo fixo, quadraticamente proporcional ao tamanho da rede: 2 minutos para 100 vértices, 8 minutos para 200 vértices, 18 min. para 300, 32 min. para 400, 50 min. para 500, 72 min. para 600, 98 min. para 700 e 128 minutos para 800 vértices.
- Sistema operacional: Linux Ubuntu Desktop 14.04.1 LTS.
- Hardware: HP Pavilion dv4 Notebook PC, Intel Core i3 M 330 2.13GHz, 2 núcleos, 64 bits, 32KB L1, 256KB L2, 3MB L3, 4GB SODIMM DDR3 1067MHz.

A Figura 10 apresenta a redução média da dispersão alcançada pelos algoritmos em cada tamanho de rede, ficando evidente que o CFM perde em eficácia à medida que se aumenta o tamanho da rede, significando que precisaria de um tempo maior para alcançar a redução desejada. O Claritate mostrou a mesma eficácia em todos os testes, independente da quantidade de vértices.

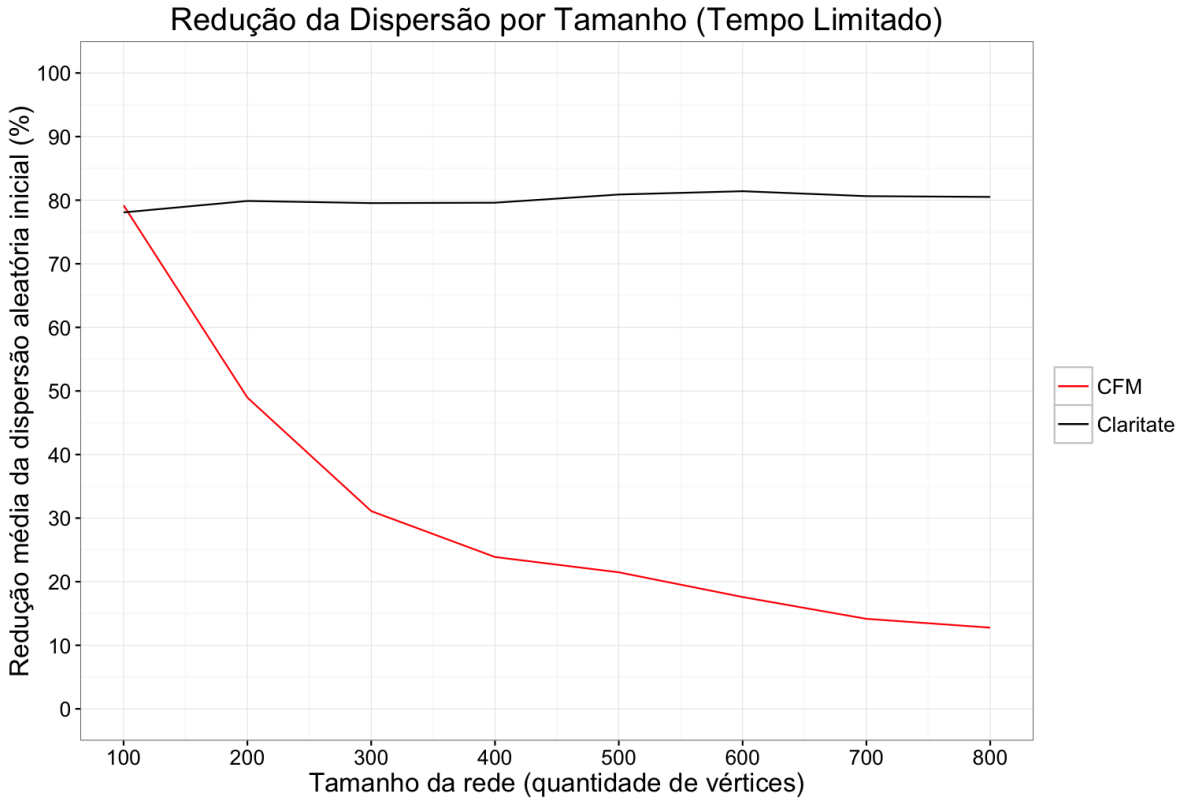

Figura 10: Redução média da dispersão aleatória inicial alcançada pelos algoritmos CFM e Claritate sobre 8 tamanhos diferentes de redes.

A Figura 11 mostra os resultados alcançados especificamente no contexto de execução do Claritate. Observa-se que este novo algoritmo possui um comportamento inicial muito acelerado e uma evolução lenta, chegando próximo do resultado final logo no início da execução, conforme o tempo disponível.

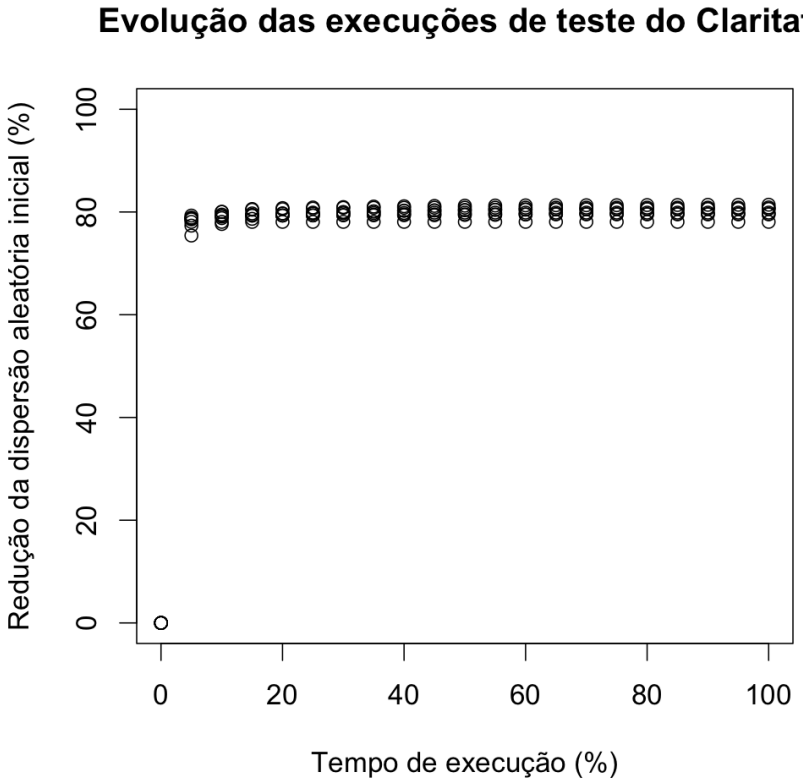

Figura 11: Redução média da dispersão aleatória inicial alcançada por cada execução do Claritate, em função do tempo decorrido.

Com base nisso, utilizou-se o Claritate para seriar a rede humana HI-II-14 de Rolland et al. (2014), com um tempo de execução de 96 horas, alcançando 71,68% de redução da dispersão inicial.

Posteriormente, calculou-se o nível de diferenciação de cada gene entre as classes de pacientes doentes e saudáveis, isto é, grupo de amostras com ALL verso saudáveis e grupo com AML verso saudáveis. A resultante enumeração de genes foi então reorganizada conforme a disposição deles na seriação da rede HI-II-14, obtendo a Figura 12.

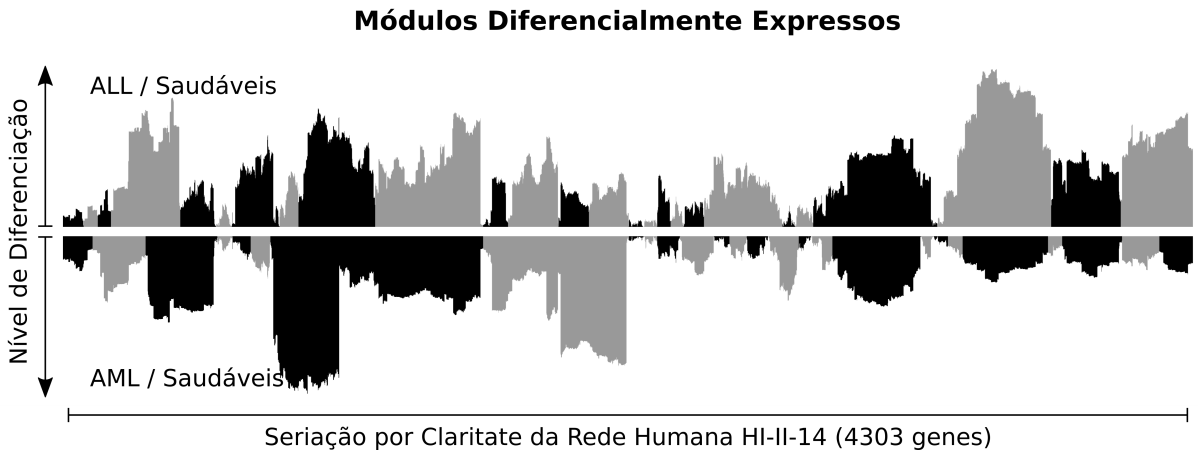

Figura 12: Perfis de expressão diferencial obtidos a partir do cruzamento da seriação por Claritate (rede protéica humana HI-II-14) com os níveis de diferenciação dos genes entre classes de pacientes doentes e saudáveis: i) ALL verso Saudáveis e ii) AML verso Saudáveis. A alternância de cores destaca os módulos detectados.

Observa-se padrões de coexpressão por adjacência, evidenciando módulos funcionais de genes que se interagem na rede protéica correspondente, o que se aproxima da noção de complexas redes regulatórias de genes. A modularização foi baseada na inspeção dos picos e vales, na qual os menores níveis de diferenciação foram usados como fronteiras modulares, pois correspondem a genes relativamente inertes ao comportamento sistêmico dos grupos vizinhos. Muitos desses genes podem estar exercendo a função de ponte entre vias metabólicas.

Sem perda da generalidade, a discussão remanescente está focada na diferenciação entre amostras com AML verso saudáveis.

Na análise de expressão diferencial clássica, o objetivo principal é a seleção de uma lista classificada dos genes que poderiam ser potenciais marcadores para diferenciar o grupo de controle do grupo alvo. Em seguida, é realizado enriquecimento funcional para encontrar solidez biológica nas assinaturas de genes descobertas. Ao invés de focar apenas nos genes com alta diferenciação, que podem ter sido mal anotados, esta nova abordagem considera todos os genes na experiência, não apenas aqueles resultantes de um limite arbitrário.

O enriquecimento funcional pode ser feito, portanto, para dois conjuntos: lista com os genes mais diferenciados de um módulo; ou lista com os genes mais diferenciados de alguns ou de todos os módulos detectados.

A disponibilidade de tantos métodos de classificação e a falta de consenso na comunidade, no que diz respeito às limitações e capacidades de todos eles, abre um espaço claro para estudos sistemáticos para avaliar melhor os métodos atuais, com critérios

relevantes e objetivos (BOULESTEIX; SLAWSKI, 2009). No entanto, a escolha de um único método não é recomendado, e então esta abordagem baseada em módulos pode ser uma alternativa em potencial para o "dilema da escolha das listas de genes".

Como um exemplo, selecionando os 100 genes mais diferencialmente expressos da lista de classificação geral dos pacientes AML verso saudáveis, obtém-se apenas 8 termos (funções biológicas) através de enriquecimento funcional, considerando as três ontologias do banco de dados Gene Ontology - Processo Biológico (BP), Componente Celular (CC) e Função Molecular (MF):

- BP GO:0045930 - negative regulation of mitotic cell cycle (p-value = 6.02×10^{-4})
- BP GO:0000075 - cell cycle checkpoint (6.02×10^{-4})
- BP GO:2000785 - regulation of autophagosome assembly (6.75×10^{-4})
- BP GO:0044088 - regulation of vacuole organization (8.75×10^{-4})
- BP GO:2000786 - positive regulation of autophagosome assembly (8.75×10^{-4})
- MF GO:0005515 - protein binding (2.44×10^{-11})
- MF GO:0005488 - binding (1.00×10^{-5})
- MF GO:0016308 - 1-phosphatidylinositol-4-phosphate 5-kinase activity (9.22×10^{-4})

Em contrapartida, aplicando a nova abordagem, selecionou-se um total de 100 genes da seriação dos DEG, considerando os 25 genes mais diferencialmente expressos de cada um dos 4 módulos com maior média de diferenciação, e obteve-se 17 termos com o enriquecimento funcional:

- BP GO:0001776 - **leukocyte homeostasis** (p-value = 7.51×10^{-4})
- BP GO:0035821 - modification of morphology or physiology of other organism (7.78×10^{-4})
- BP GO:0002513 - tolerance induction to self antigen (7.78×10^{-4})
- BP GO:0002260 - **lymphocyte homeostasis** (8.18×10^{-4})
- BP GO:0032945 - negative regulation of mononuclear cell proliferation (8.18×10^{-4})
- BP GO:0050672 - negative regulation of lymphocyte proliferation (8.18×10^{-4})
- BP GO:0001782 - **B cell homeostasis** (8.18×10^{-4})
- BP GO:0070664 - negative regulation of leukocyte proliferation (8.18×10^{-4})

- BP GO:0051817 - modification of morphology or physiology of other organism involved in symbiotic interaction (8.18×10^{-4})
- BP GO:0018107 - peptidyl-threonine phosphorylation (8.18×10^{-4})
- BP GO:0050869 - negative regulation of B cell activation (8.18×10^{-4})
- BP GO:1901841 - regulation of high voltage-gated calcium channel activity (8.18×10^{-4})
- BP GO:0010799 - regulation of peptidyl-threonine phosphorylation (8.18×10^{-4})
- BP GO:0018210 - peptidyl-threonine modification (8.18×10^{-4})
- BP GO:0007435 - salivary gland morphogenesis (8.27×10^{-4})
- MF GO:0005515 - protein binding (1.09×10^{-16})
- MF GO:0005488 - binding (8.00×10^{-7})

De fato, a estratégia proposta foi capaz de destacar termos (em negrito) fortemente relacionados com o estudo experimental, sendo mais sensível ao contexto biomédico da leucemia.

Considerando que todos os módulos identificados, mesmo com baixa diferenciação média, possuem importância analítica, pois revelam um conjunto de genes afins diferencialmente expressos, pode-se também executar o enriquecimento funcional para cada um deles, sem descartar genes membros. Dessa forma, considerando o exemplo em discussão (AML verso saudáveis), obteve-se conjuntamente 737 termos BP, 210 termos CC e 108 termos MF. Selecionando o termo BP com melhor *p-value* de cada módulo, respeitando a ordem decrescente por diferenciação média dos módulos, têm-se como os 15 primeiros:

- GO:0035556 - intracellular signal transduction
- GO:0007049 - cell cycle
- GO:0002376 - immune system process
- GO:0000209 - protein polyubiquitination
- GO:1901685 - glutathione derivative metabolic process
- GO:0016070 - RNA metabolic process
- GO:0010467 - gene expression
- GO:0015031 - protein transport

- GO:0044260 - cellular macromolecule metabolic process
- GO:0044403 - symbiosis, encompassing mutualism through parasitism
- GO:0090174 - organelle membrane fusion
- GO:0008380 - RNA splicing
- GO:0060370 - **susceptibility to T cell mediated cytotoxicity**
- GO:0051534 - negative regulation of NFAT protein import into nucleus
- GO:0030422 - production of siRNA involved in RNA interference

Nota-se que os termos são mais gerais, o que pode ser explicado pela inclusão de mais módulos diferencialmente expressos. No entanto, há um termo em negrito que está intimamente relacionado ao contexto biomédico.

Para expandir esta análise comparativa, utilizou-se também o método GSEA de Subramanian et al. (2005). Tomando ainda o estudo de caso dos pacientes com AML verso saudáveis, com foco na Gene Ontology, têm-se como resultado o enriquecimento de 225 termos, dentre os quais se destacam, no contexto da ontologia BP, os 15 termos a seguir:

- GO:0006508 - proteolysis
- GO:0007088 - regulation of mitotic nuclear division
- GO:0032940 - secretion by cell
- GO:0007610 - behavior
- GO:0044257 - cellular protein catabolic process
- GO:0006512 - obsolete ubiquitin cycle
- GO:0030163 - protein catabolic process
- GO:0051248 - negative regulation of protein metabolic process
- GO:0009628 - response to abiotic stimulus
- GO:0045045 - obsolete secretory pathway
- GO:0007626 - locomotory behavior
- GO:0043285 - biopolymer catabolic process
- GO:0045184 - establishment of protein localization

- GO:0009967 - positive regulation of signal transduction
- GO:0051641 - cellular localization

Os termos identificados são mais genéricos que os termos previamente citados, o que evidencia uma menor sensibilidade do método GSEA ao contexto biomédico, em comparação à nova estratégia modular.

A Figura 12 demonstra os módulos diferencialmente expressos identificados para fins de análise funcional sobre duas classes de problema (ALL/Saudáveis e AML/Saudáveis). Entretanto, fica difícil estabelecer uma comparação entre elas, pois os módulos divergem em quantidade e tamanho. Entretanto, pode-se realizar uma comparação de DEG seriados através do método de transcriptograma, o qual normaliza e generaliza a assinatura transcricional. Com base nisso, efetuou-se o cálculo de transcriptograma dos resultados expressos na Figura 12, obtendo-se duas curvas comparáveis através da modularidade por janela (fundo cinza), demonstradas na Figura 13. Percebe-se, assim, uma relevante alteração na expressão diferencial dos genes nos casos de AML/Saudáveis em comparação aos de ALL/Saudáveis.

Além disso, selecionou-se também um grupo de 461 genes, apresentado como módulo sob demanda na Figura 13, a fim de analisar uma região similar entre as duas assinaturas. Através de enriquecimento funcional, obteve-se 220 termos BP, 26 termos CC e 15 termos MF, sendo GO:0000209 (protein polyubiquitination, p-value=6.23×10^{-10}), GO:0005634 (nucleus, p-value=4.15×10^{-12}) e GO:0005515 (protein binding, p-value=1.59×10^{-58}) os termos com melhor *p-value*.

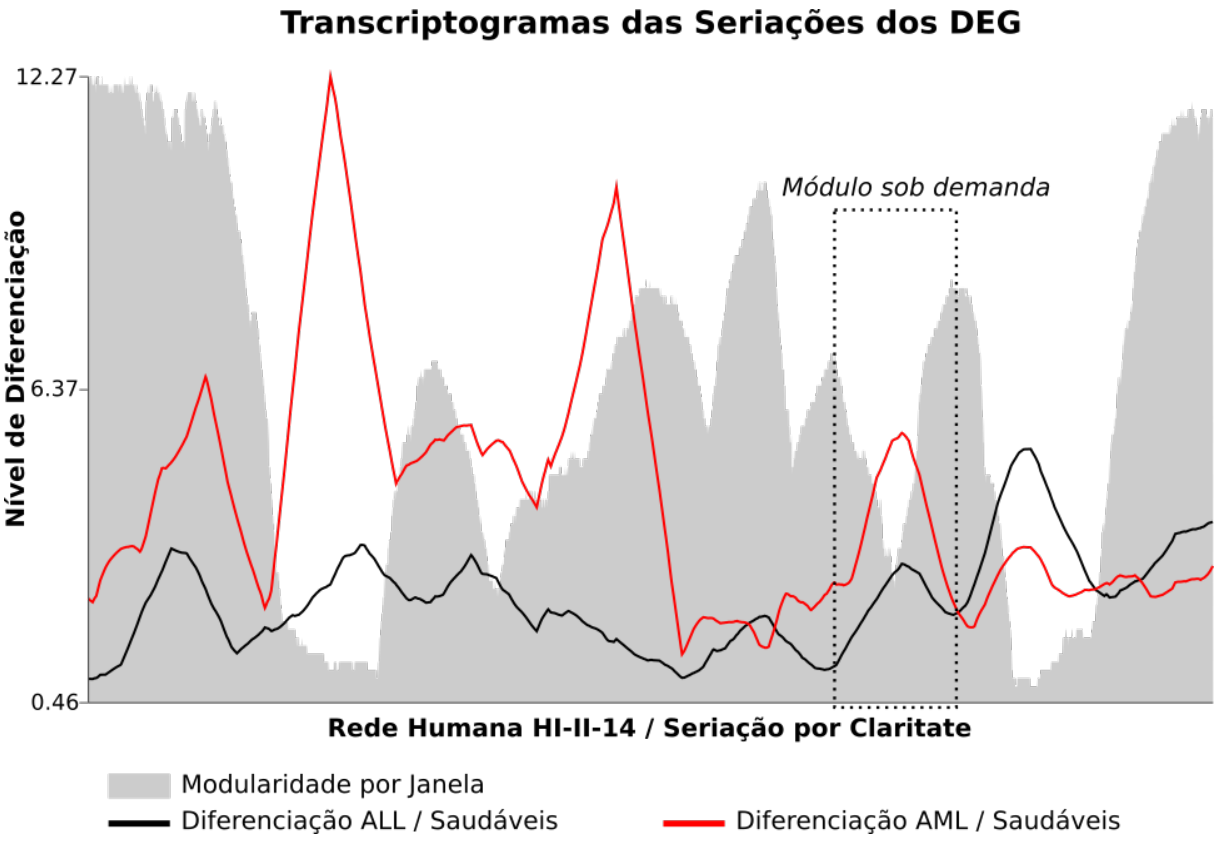

Figura 13: Transcriptogramas dos níveis de diferenciação dos genes seriados por Claritate (rede protéica humana HI-II-14), sobre dois grupos de comparação: i) pacientes ALL verso saudáveis (cor preta) e ii) pacientes AML verso saudáveis (cor vermelha). A imagem de fundo (cor cinza) corresponde à modularidade por janela da rede seriada. Há um exemplo de módulo selecionado sob demanda, destacando um grupo de genes para comparação por meta-análise.

Por fim, através da ferramenta PPISURV <http://bioprofiling.de>, realizou-se uma análise de sobrevivência de câncer sobre a lista de 100 genes composta pelos 25 genes mais diferencialmente expressos de cada um dos 4 módulos com maior média de diferenciação do experimento AML/Saudáveis. Curiosamente, 37 genes foram identificados como positivos para linfoma difuso de grandes células B, conforme o experimento GSE10846, disponível no banco de dados Gene Expression Omnibus (GEO). Um deles, o gene UBE2R2, está diretamente associado aos genes UBE2I e DTX3L, conforme exposto em <http://biograph.be/concept/graph/C1150669/C1421283>. O UBE2I é apontado por Macrae et al. (2013) como um candidato gene de controle endógeno em células hematopoiéticas normais. Além disso, alguns genes foram associados, positivo ou negativamente, a outros tipos de câncer pela PPISURV: mama (15 genes) e pulmão (21 genes).

Todos os dados e a sequência de comandos desta análise estão disponíveis em https://github.com/joseflaviojr/transcriptograma/ tree/master/UseCase-Leukemia.

5 Conclusão

Análises de transcriptomas frequentemente agrupam genes por coexpressão ou por covariação no tempo, o que implica numa forte dependência do estágio em que a célula se encontra ou do protocolo usado para extrair os dados. Transcriptogramas são independentes de protocolo ou tecnologia, isso porque a seriação identifica módulos diretamente sobre a rede protéica para melhor explorar perfis de expressão de genes.

A nova abordagem que conjuga transcriptograma com análise orientada a módulos de genes diferencialmente expressos apresenta resultados mais específicos e sensíveis ao contexto biomédico do câncer, tornando-se uma estratégia analítica promissora na área da bioinformática.

A tradicional seleção de genes diferencialmente expressos e potencialmente relacionados a um estado biológico pode ser melhorada relevantemente, conforme exposto neste trabalho, através da seleção por módulos seriados, considerando todo o espectro do perfil de expressão diferencial.

Além disso, o algoritmo de seriação de proteínas Claritate mostra-se como uma alternativa equivalente ao CFM, destacando-se principalmente pela eficiência, alcançada com a otimização para redes livres de escala.

Existem, entretanto, algumas questões e limitações a serem exploradas mais precisamente, tais como:

- Rede Protéica: topologias compatíveis, ligações ponderadas.
- Seriação: condição de parada do Claritate, dispersão máxima aceitável, otimização por paralelismo na execução.
- Modularidade: ajuste do tamanho da janela, particionamento automático, índice de qualidade dos módulos.
- Análise: enriquecimento funcional de módulos de genes associado a metadados clínicos.

Referências

AHAD, N. A.; YAHAYA, S. S. S. Sensitivity analysis of welch's t-test. *AIP Conference Proceedings*, v. 1605, n. 1, 2014. Citado na página 35.

ANDREOPOULOS, B. et al. A roadmap of clustering algorithms: finding a match for a biomedical application. *Briefings in Bioinformatics*, v. 10, n. 3, p. 297–314, 2009. Disponível em: <http://dblp.uni-trier.de/db/journals/bib/bib10.html#AndreopoulosAWS09>. Citado 3 vezes nas páginas 14, 15 e 20.

BAGGERLY, K. A. et al. Identifying Differentially Expressed Genes in cDNA Microarray Experiments. *Journal of Computational Biology*, v. 8, n. 6, p. 639–659, nov. 2001. ISSN 1066-5277. Disponível em: <http://dx.doi.org/10.1089/106652701753307539>. Citado na página 20.

BARABáSI, A.-L.; ALBERT, R. Emergence of Scaling in Random Networks. *Science*, American Association for the Advancement of Science, v. 286, n. 5439, p. 509–512, out. 1999. ISSN 1095-9203. Disponível em: <http://dx.doi.org/10.1126/science.286.5439.509>. Citado 3 vezes nas páginas 30, 31 e 36.

BARKAI, N. *Sporulation transfer to YPA1*. 2006. Disponível em: <http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE3815>. Citado 3 vezes nas páginas 9, 55 e 56.

BENJAMINI, Y.; HOCHBERG, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, Blackwell Publishing for the Royal Statistical Society, v. 57, n. 1, p. 289–300, 1995. ISSN 00359246. Disponível em: <http://dx.doi.org/10.2307/2346101>. Citado 2 vezes nas páginas 27 e 35.

BERKHIN, P. A Survey of Clustering Data Mining Techniques. In: KOGAN, J.; NICHOLAS, C.; TEBOULLE, M. (Ed.). *Grouping Multidimensional Data*. Berlin/Heidelberg: Springer Berlin Heidelberg, 2006. cap. 2, p. 25–71. ISBN 3-540-28348-X. Disponível em: <http://dx.doi.org/10.1007/3-540-28349-8_2>. Citado na página 14.

BERTSIMAS, D.; TSITSIKLIS, J. Simulated Annealing. *Statistical Science*, v. 8, n. 1, p. 10–15, 1993. Citado 2 vezes nas páginas 24 e 34.

BIGGS, N.; LLOYD, E. K.; WILSON, R. J. *Graph Theory, 1736-1936*. New York, NY, USA: Clarendon Press, 1986. ISBN 0-198-53916-9. Citado na página 20.

BOULESTEIX, A.-L.; SLAWSKI, M. Stability and aggregation of ranked gene lists. *Briefings in Bioinformatics*, Oxford University Press, v. 10, n. 5, p. 556–568, set. 2009. ISSN 1477-4054. Disponível em: <http://dx.doi.org/10.1093/bib/bbp034>. Citado 2 vezes nas páginas 35 e 41.

CARBON, S. et al. AmiGO: online access to ontology and annotation data. *Bioinformatics (Oxford, England)*, v. 25, n. 2, p. 288–289, jan. 2009. ISSN 1367-4811. Disponível em: <http://dx.doi.org/10.1093/bioinformatics/btn615>. Citado 3 vezes nas páginas 26, 27 e 35.

CARRASCO, J. J. et al. *Clustering of Bipartite Advertiser-Keyword Graph*. 2003. Citado na página 34.

CHEN, X. et al. Gene Expression Patterns in Human Liver Cancers. *Molecular Biology of the Cell*, American Society for Cell Biology, v. 13, n. 6, p. 1929–1939, jun. 2002. ISSN 1939-4586. Disponível em: <http://dx.doi.org/10.1091/mbc.02-02-0023>. Citado na página 20.

COMAN, D.; RÜTIMANN, P.; GRUISSEM, W. A flexible protocol for targeted gene co-expression network analysis. In: _____. *Plant Isoprenoids: Methods and Protocols*. New York, NY: Springer New York, 2014. p. 285–299. ISBN 978-1-4939-0606-2. Disponível em: <http://dx.doi.org/10.1007/978-1-4939-0606-2_21>. Citado na página 17.

CONSORTIUM, T. G. O. Gene ontology consortium: going forward. *Nucleic Acids Research*, v. 43, n. D1, p. D1049–D1056, 2015. Disponível em: <http://nar.oxfordjournals.org/content/43/D1/D1049.abstract>. Citado 2 vezes nas páginas 26 e 35.

DERRICK, B.; TOHER, D.; WHITE, P. Why welchs test is type i error robust. *The Quantitative Methods for Psychology*, TQMP, v. 12, n. 1, p. 30–38, 2016. Disponível em: <http://www.tqmp.org/RegularArticles/vol12-1/p030/p030.pdf>. Citado na página 35.

DING, L. et al. Analysis of next-generation genomic data in cancer: accomplishments and challenges. *Human Molecular Genetics*, v. 19, n. R2, p. R188–R196, 2010. Disponível em: <http://hmg.oxfordjournals.org/content/19/R2/R188.abstract>. Citado na página 18.

DUDOIT, R. et al. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. In: *Statistica Sinica*. [s.n.], 2002. p. 111–139. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.117.9702>. Citado 2 vezes nas páginas 20 e 26.

FLOYD, R. W. Algorithm 97: Shortest path. *Communications of the ACM*, ACM, New York, NY, USA, v. 5, n. 6, p. 345–, jun. 1962. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/367766.368168>. Citado 2 vezes nas páginas 30 e 33.

GE, H.; WALHOUT, A. J.; VIDAL, M. Integrating 'omic' information: a bridge between genomics and systems biology. *Trends in genetics : TIG*, Dana-Farber Cancer Institute and Department of Genetics, Harvard Medical School, SM858, 44 Binney Street, Boston, MA 02115, USA., v. 19, n. 10, p. 551–560, out. 2003. ISSN 0168-9525. Disponível em: <http://view.ncbi.nlm.nih.gov/pubmed/14550629>. Citado na página 15.

GOH, K.-I. et al. The human disease network. *Proceedings of the National Academy of Sciences*, v. 104, n. 21, p. 8685–8690, 2007. Disponível em: <http://www.pnas.org/content/104/21/8685.abstract>. Citado na página 17.

HAGE, P. Eccentricity and centrality in networks. *Social Networks*, v. 17, n. 1, p. 57–63, jan. 1995. ISSN 03788733. Disponível em: <http://dx.doi.org/10.1016/0378-8733(94)00248-9>. Citado 4 vezes nas páginas 30, 31, 33 e 36.

HERNANDEZ, P. J.; ABEL, T. The role of protein synthesis in memory consolidation: progress amid decades of debate. *Neurobiology of learning and memory*, v. 89, n. 3, p. 293–311, mar. 2008. ISSN 1095-9564. Disponível em: <http://dx.doi.org/10.1016/j.nlm.2007.09.010>. Citado na página 13.

HOARE, C. A. R. Algorithm 64: Quicksort. *Commun. ACM*, ACM, New York, NY, USA, v. 4, n. 7, p. 321–, jul. 1961. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/366622.366644>. Citado na página 33.

HUNG, J.-H.; WENG, Z. Analysis of microarray and rna-seq expression profiling data. *Cold Spring Harbor Protocols*, 2016. Disponível em: <http://cshprotocols.cshlp.org/content/early/2016/08/27/pdb.top093104.abstract>. Citado na página 14.

JEANMOUGIN, M. et al. Should we abandon the t-test in the analysis of gene expression microarray data: a comparison of variance modeling strategies. *PloS one*, Public Library of Science, v. 5, n. 9, p. e12336+, set. 2010. ISSN 1932-6203. Disponível em: <http://dx.doi.org/10.1371/journal.pone.0012336>. Citado na página 35.

JENSEN, L. J. et al. STRING 8–a global view on proteins and their functional interactions in 630 organisms. *Nucleic acids research*, Oxford University Press, v. 37, n. Database issue, p. D412–D416, jan. 2009. ISSN 1362-4962. Disponível em: <http://dx.doi.org/10.1093/nar/gkn760>. Citado 3 vezes nas páginas 16, 29 e 34.

KLEINBERG, J.; TARDOS, E. Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Labeling and Markov Random Fields. *J. ACM*, ACM, New York, NY, USA, v. 49, n. 5, p. 616–639, set. 2002. ISSN 0004-5411. Disponível em: <http://dx.doi.org/10.1145/585265.585268>. Citado na página 14.

KOUROU, K. et al. Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal*, v. 13, p. 8 – 17, 2015. ISSN 2001-0370. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2001037014000464>. Citado na página 14.

KUENTZER, F. et al. Optimization and analysis of seriation algorithm for ordering protein networks. In: *Bioinformatics and Bioengineering (BIBE), 2014 IEEE International Conference on*. [S.l.: s.n.], 2014. p. 231–237. Citado 3 vezes nas páginas 16, 17 e 24.

LAARHOVEN, T. V.; MARCHIORI, E. Axioms for Graph Clustering Quality Functions. *J. Mach. Learn. Res.*, JMLR.org, v. 15, n. 1, p. 193–215, jan. 2014. ISSN 1532-4435. Disponível em: <http://portal.acm.org/citation.cfm?id=2627435.2627441>. Citado na página 15.

MACRAE, T. et al. RNA-Seq Reveals Spliceosome and Proteasome Genes as Most Consistent Transcripts in Human Cancer Cells. *PloS one*, Public Library of Science, v. 8, n. 9, p. e72884+, set. 2013. ISSN 1932-6203. Disponível em: <http://dx.doi.org/10.1371/journal.pone.0072884>. Citado 2 vezes nas páginas 29 e 45.

MOOTHA, V. K. et al. PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature genetics*, Nature Publishing Group, Whitehead Institute/MIT Center for Genome Research, Cambridge, Massachusetts, USA., v. 34, n. 3, p. 267–273, jul. 2003. ISSN 1061-4036. Disponível em: <http://dx.doi.org/10.1038/ng1180>. Citado na página 27.

PADHORNY, D. et al. Protein–protein docking by fast generalized Fourier transforms on 5D rotational manifolds. *Proceedings of the National Academy of Sciences*, v. 113, n. 30, p. E4286–E4293, jul. 2016. Disponível em: <http://dx.doi.org/10.1073/pnas.1603929113>. Citado na página 15.

PINELLI, M. et al. An atlas of gene expression and gene co-regulation in the human retina. *Nucleic Acids Research*, v. 44, n. 12, p. 5773–5784, 2016. Disponível em: <http://nar.oxfordjournals.org/content/44/12/5773.abstract>. Citado 2 vezes nas páginas 17 e 26.

RENFROW, J. J. et al. Gene–protein correlation in single cells. *Neuro-Oncology*, v. 13, n. 8, p. 880–885, 2011. Disponível em: <http://neuro-oncology.oxfordjournals.org/content/13/8/880.abstract>. Citado na página 13.

ROLLAND, T. et al. A proteome-scale map of the human interactome network. *Cell*, v. 159, n. 5, p. 1212 – 1226, 2014. ISSN 0092-8674. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0092867414014226>. Citado 3 vezes nas páginas 16, 29 e 39.

ROMERO-CAMPERO, F. J. et al. Chlamynet: a chlamydomonas gene co-expression network reveals global properties of the transcriptome and the early setup of key co-expression patterns in the green lineage. *BMC Genomics*, v. 17, n. 1, p. 227, 2016. ISSN 1471-2164. Disponível em: <http://dx.doi.org/10.1186/s12864-016-2564-y>. Citado na página 17.

RYBARCZYK-FILHO, J. L. L. et al. Towards a genome-wide transcriptogram: the Saccharomyces cerevisiae case. *Nucleic acids research*, Oxford University Press, v. 39, n. 8, p. 3005–3016, abr. 2011. ISSN 1362-4962. Disponível em: <http://dx.doi.org/10.1093/nar/gkq1269>. Citado 14 vezes nas páginas 8, 9, 14, 16, 17, 21, 23, 24, 25, 27, 33, 34, 36 e 62.

SCHAEFFER, S. E. *Algorithms for Nonuniform Networks*. Espoo, Finland, 2006. xxii+183 p. Doctoral dissertation. Citado na página 34.

SCHAEFFER, S. E. Survey: Graph clustering. *Comput. Sci. Rev.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 1, n. 1, p. 27–64, ago. 2007. ISSN 1574-0137. Disponível em: <http://dx.doi.org/10.1016/j.cosrev.2007.05.001>. Citado 8 vezes nas páginas 15, 16, 20, 23, 31, 33, 34 e 37.

SERIN, E. A. R. et al. Learning from co-expression networks: Possibilities and challenges. *Frontiers in Plant Science*, v. 7, p. 444, 2016. ISSN 1664-462X. Disponível em: <http://journal.frontiersin.org/article/10.3389/fpls.2016.00444>. Citado 4 vezes nas páginas 16, 17, 21 e 26.

SILVA, S. R. da et al. Reproducibility enhancement and differential expression of non predefined functional gene sets in human genome. *BMC genomics*, BioMed Central Ltd, v. 15, n. 1, p. 1181, 2014. Citado 4 vezes nas páginas 17, 18, 21 e 24.

SLONIM, D. K. From patterns to pathways: gene expression data analysis comes of age. *Nat Genet*, Nature Publishing Group, Department of Genomics, Wyeth Research, 35 Cambridge Park Drive, Cambridge, Massachusetts 02140, USA. dslonim@wyeth.com, v. 32 Suppl, p. 502–508, dez. 2002. ISSN 1061-4036. Disponível em: <http://dx.doi.org/10.1038/ng1033>. Citado na página 16.

SONESON, C.; DELORENZI, M. A comparison of methods for differential expression analysis of RNA-seq data. *BMC bioinformatics*, BioMed Central Ltd, v. 14, n. 1, p. 91+, mar. 2013. ISSN 1471-2105. Disponível em: <http://dx.doi.org/10.1186/1471-2105-14-91>. Citado 2 vezes nas páginas 20 e 21.

SUBRAMANIAN, A. et al. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, v. 102, n. 43, p. 15545–15550, 2005. Disponível em: <http://www.pnas.org/content/102/43/15545.abstract>. Citado 3 vezes nas páginas 27, 28 e 43.

TU, B. et al. *Logic of the yeast metabolic cycle*. 2005. Disponível em: <http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE3431>. Citado 4 vezes nas páginas 9, 25, 53 e 54.

VIDAL, M.; CUSICK, M. E.; BARABÁSI, A.-L. Interactome networks and human disease. *Cell*, v. 144, n. 6, p. 986 – 998, 2011. ISSN 0092-8674. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0092867411001309>. Citado 4 vezes nas páginas 13, 15, 16 e 20.

WANG, Y.-C. et al. Computational probing protein–protein interactions targeting small molecules. *Bioinformatics*, Oxford University Press, v. 32, n. 2, p. 226–234, jan. 2016. ISSN 1460-2059. Disponível em: <http://dx.doi.org/10.1093/bioinformatics/btv528>. Citado na página 17.

WANG, Z.; GERSTEIN, M.; SNYDER, M. RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews. Genetics*, Nature Publishing Group, v. 10, n. 1, p. 57–63, jan. 2009. ISSN 1471-0064. Disponível em: <http://dx.doi.org/10.1038/nrg2484>. Citado na página 18.

YOOK, S. hyung; OLTVAI, Z. N.; BARABÁSI, A. lászló. Functional and topological characterization of protein interaction networks. *Proteomics*, v. 4, p. 928–942, 2004. Citado 2 vezes nas páginas 13 e 15.

Apêndices

APÊNDICE A – Transcriptogramas do experimento NCBI/GEO/GSE3431

Transcriptogramas do experimento GSE3431 de Tu et al. (2005), disponível no banco de dados NCBI/GEO, que consiste em análise do ciclo metabólico da *Saccharomyces cerevisiae*.

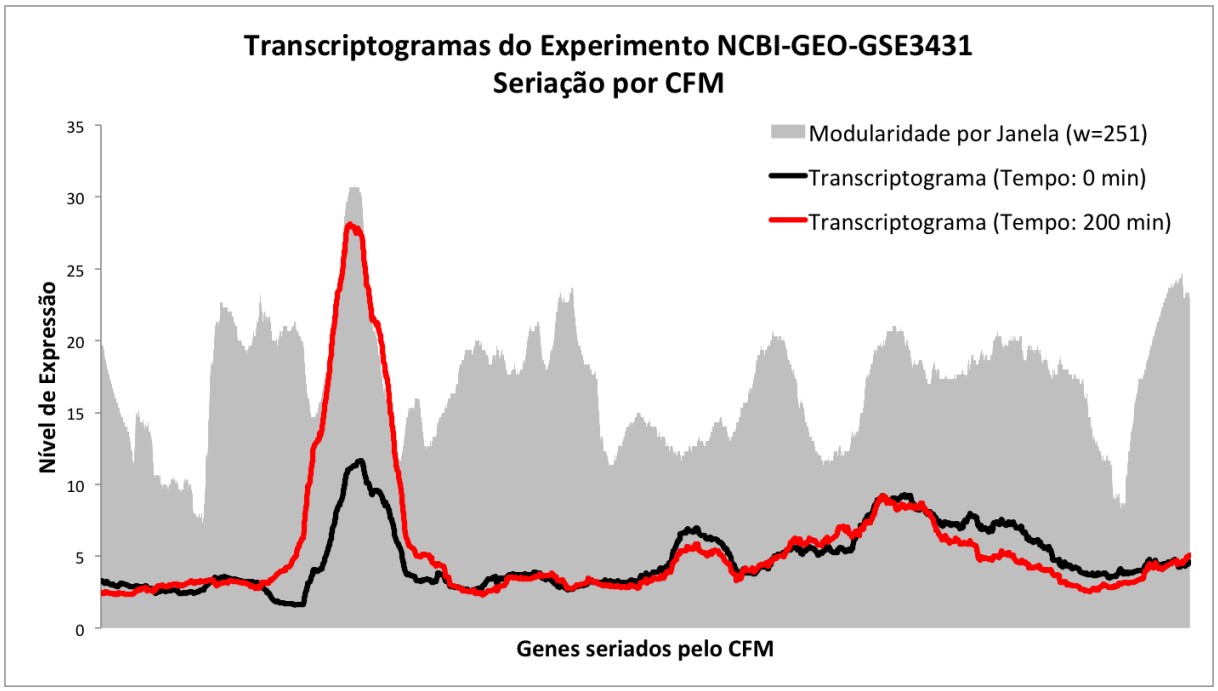

Figura 14: Transcriptogramas do experimento NCBI/GEO/GSE3431 de Tu et al. (2005). Seriação por CFM.

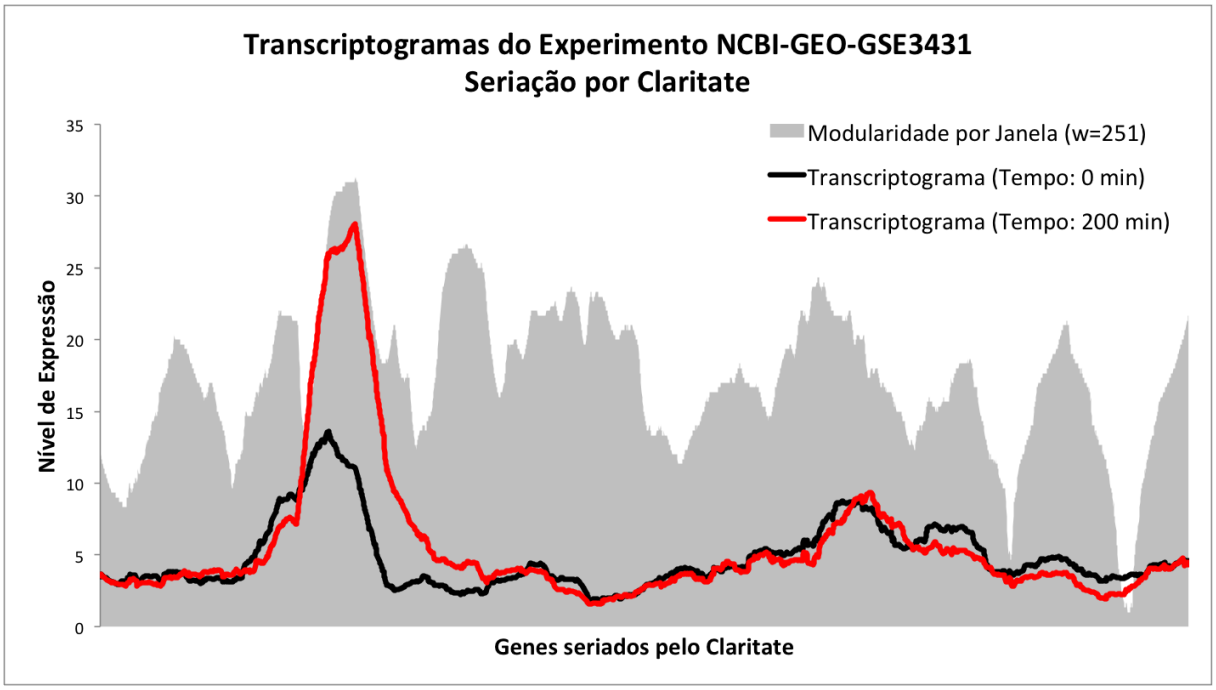

Figura 15: Transcriptogramas do experimento NCBI/GEO/GSE3431 de Tu et al. (2005). Seriação por Claritate.

APÊNDICE B – Transcriptogramas do experimento NCBI/GEO/GSE3815

Transcriptogramas do experimento GSE3815 de Barkai (2006), disponível no banco de dados NCBI/GEO, que consiste em análise da *Saccharomyces cerevisiae* em processo de esporulação.

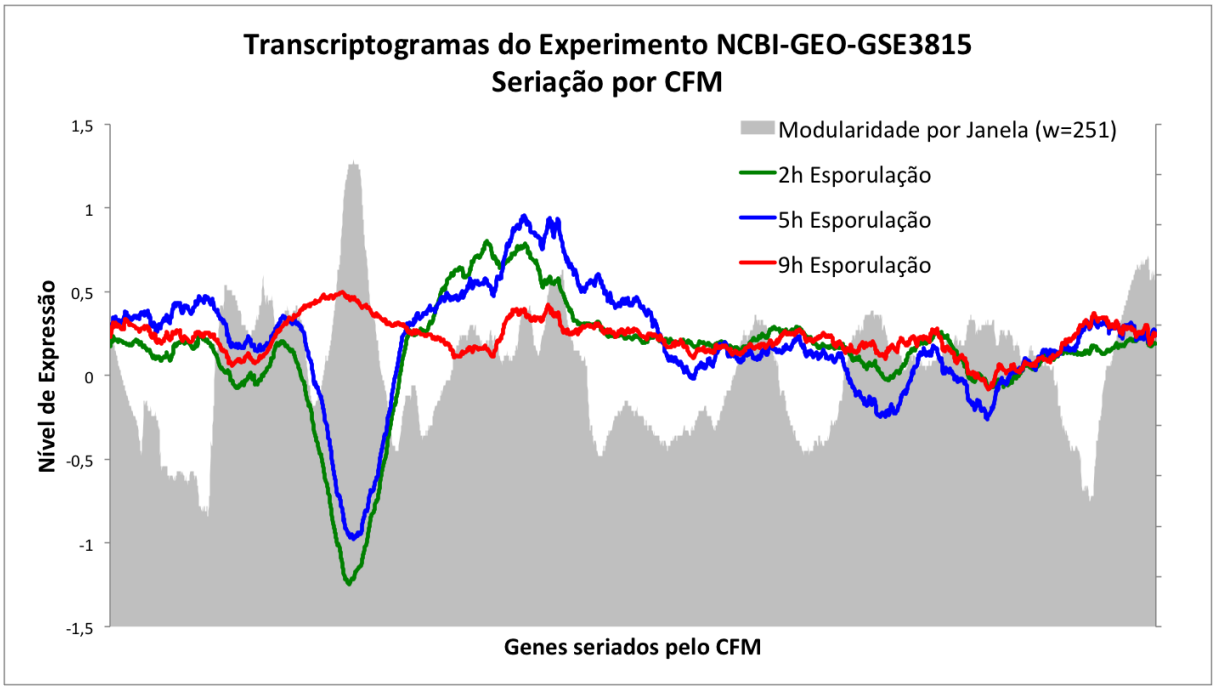

Figura 16: Transcriptogramas do experimento NCBI/GEO/GSE3815 de Barkai (2006). Seriação por CFM.

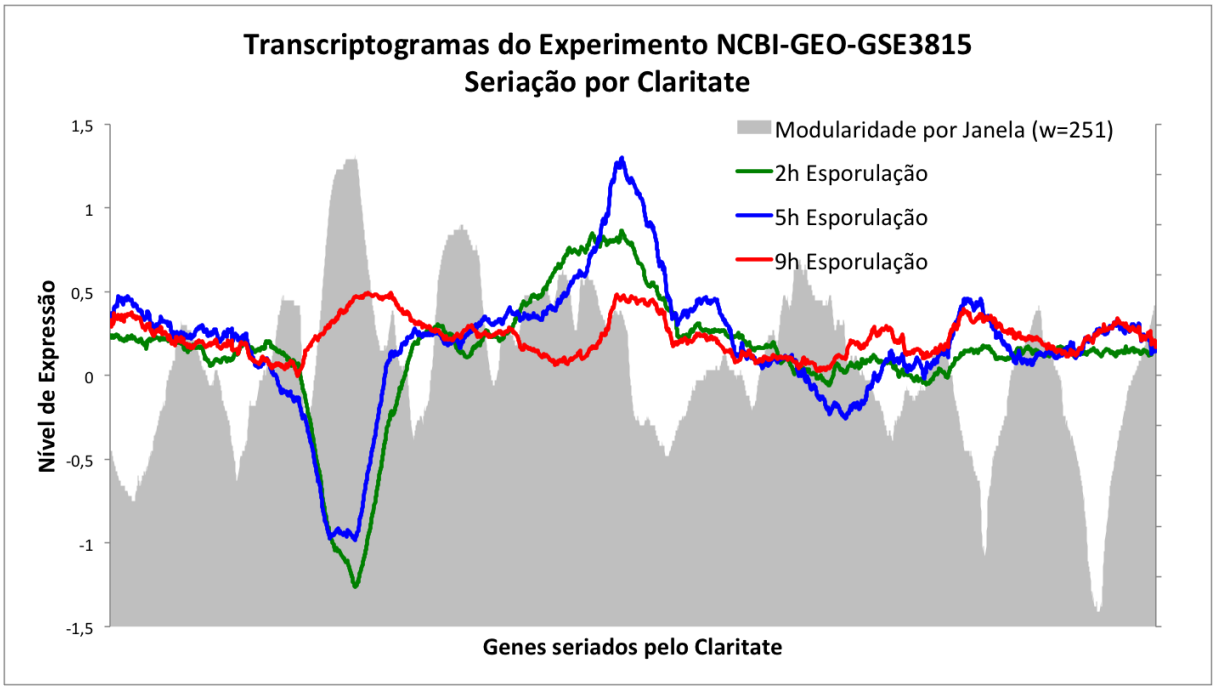

Figura 17: Transcriptogramas do experimento NCBI/GEO/GSE3815 de Barkai (2006). Seriação por Claritate.

APÊNDICE C – Código Fonte de Referência do Claritate

O código fonte em C++, a seguir, corresponde à implementação de referência do algoritmo Claritate, disponível publicamente no endereço Web <https://github.com/joseflaviojr/claritate>, sob a Licença Pública Menos Geral do GNU <http://www.gnu.org/licenses/>.

Código Fonte C.1: Código fonte de referência do Claritate, implementado na linguagem de programação C++.

```
1
2  int Claritate( int** matriz1, int** matriz2, int total, int L1, int L2, ofstream* saida ) {
3
4    // Inicializacao
5
6    int**         adjacencia   = matriz1;
7    int**         distancia    = matriz2;
8    int*          grau_entrada = new int[total];
9    Centralidade* centralidade = new Centralidade[total];
10
11   for( int i = 0; i < total; i++ ){
12     grau_entrada[i]         = 0;
13     centralidade[i].posicao = i;
14     centralidade[i].valor   = 0;
15   }
16
17   bool orientado =
18   Inicializar( adjacencia, distancia, grau_entrada, centralidade, total );
19
20   // Dispersao
21
22   qsort( centralidade, total, sizeof(Centralidade), CompararCentralidade );
23
24   int ordem[total];
25   int posicao[total], posicao_atual, posicao_nova;
26
27   for( int i = 0, pos; i < total; i++ ){
28     pos = centralidade[i].posicao;
29     ordem[i] = pos;
30     posicao[pos] = i;
31   }
32
33   for( int i = 0, j, filhos, media; i < total; i++ ){
34
35     filhos = 0;
36     media = 0;
37
38     for( j = 0; j < total; j++ ){
39       if( adjacencia[i][j] != DESCONEXO && ( orientado || grau_entrada[j] <= grau_entrada[i] ) )
```

```
    {
40      filhos++;
41      media += posicao[j];
42    }
43  }
44
45  if( filhos == 0 ) continue;
46
47  posicao_atual = posicao[i];
48  posicao_nova = ceil( (float) media / filhos );
49
50  Deslocar( posicao_atual, posicao_nova, ordem, total );
51  for( j = 0; j < total; j++ ) posicao[ordem[j]] = j;
52
53 }
54
55 // Compressao
56
57 long dispersao, dispersao_nova;
58
59 dispersao = Dispersao( adjacencia, ordem, total, orientado );
60
61 int s1, s2, s3;
62 int v1, v2, v3;
63 int dist1, dist2;
64 int real1, real2;
65 int novo1, novo2;
66 int pivo;
67
68 int melhorias = 0;
69 int ciclos = 0;
70
71 while( dispersao > 0 ){
72
73   s1 = Sortear(total-1);
74   s3 = Sortear(total-1);
75   if( abs(s3-s1) <= 1 ) continue;
76   if( s1 > s3 ){
77     s2 = s1;
78     s1 = s3;
79     s3 = s2;
80   }
81   s2 = s1 + Sortear(s3-s1-2) + 1;
82
83   v1 = ordem[s1];
84   v2 = ordem[s2];
85   v3 = ordem[s3];
86
87   dist1 = distancia[v1][v2];
88   dist2 = distancia[v2][v3];
89
90   if( dist1 == DESCONEXO || dist2 == DESCONEXO ) continue;
91
92   real1 = s2 - s1;
93   real2 = s3 - s2;
94
95   novo1 = ceil( ( ((float)dist1) / ( dist1 + dist2 ) ) * ( real1 + real2 ) );
```

```
96      novo2 = real1 + real2 - novo1;
97
98      pivo = Sortear(100);
99
100     if( pivo <= 33 ){
101
102       posicao_atual = s1;
103       posicao_nova  = s2 - novo1;
104       if( posicao_nova < 0 ) posicao_nova = 0;
105
106     }else if( pivo <= 66 ){
107
108       posicao_atual = s2;
109       posicao_nova  = s1 + novo1;
110
111     }else{
112
113       posicao_atual = s3;
114       posicao_nova  = s2 + novo2;
115       if( posicao_nova >= total ) posicao_nova = total - 1;
116
117     }
118
119     Deslocar( posicao_atual, posicao_nova, ordem, total );
120
121     dispersao_nova = Dispersao( adjacencia, ordem, total, orientado );
122
123     if( dispersao_nova < dispersao ){
124       dispersao = dispersao_nova;
125       melhorias++;
126     }else{
127       Deslocar( posicao_nova, posicao_atual, ordem, total );
128     }
129
130     ciclos++;
131
132     if( melhorias == L1 || ciclos == L2 ){
133       saida->seekp( 0, saida->beg );
134       ImprimirResultado( ordem, total, saida );
135       saida->flush();
136       melhorias = 0;
137       ciclos = 0;
138     }
139
140   }
141
142   return 0;
143
144 }
145
146 /*
147  *  Algoritmos conjugados para inicializacao do Claritate.
148  *  Floyd, R. W. (1962). Algorithm 97: Shortest path. Communications of the ACM, 5(6):345.
149  *  Hage, P. and Harary, F. (1995). Eccentricity and centrality in networks. Social Networks,
       17:5763.
150  */
151 bool Inicializar( int** adjacencia, int** distancia, int* grau_entrada, Centralidade*
```

```
      centralidade, int total ) {
152
153   bool orientado = false;
154   int i, j, k, a, b, distanciaij;
155
156   for( i = 0; i < total; i++ ){
157
158     for( j = 0; j < total; j++ ){
159
160       // Floyd-Warshall
161       distanciaij = distancia[i][j];
162       for( k = 0; k < total; k++ ){
163         a = distancia[i][k];
164         b = distancia[k][j];
165         b = a == DESCONEXO || b == DESCONEXO ? DESCONEXO : a + b;
166         if( b != DESCONEXO && ( distanciaij == DESCONEXO || b < distanciaij ) ) distanciaij = b;
167       }
168       distancia[i][j] = distanciaij;
169
170       // Centralidade por excentricidade
171       if( distanciaij > centralidade[i].valor ) centralidade[i].valor = distanciaij;
172
173       // Grafo orientado?
174       if( ! orientado && adjacencia[i][j] != adjacencia[j][i] ) orientado = true;
175
176       // Grau de entrada
177       if( adjacencia[j][i] != DESCONEXO ) grau_entrada[i]++;
178
179     }
180
181     if( centralidade[i].valor != 0 ) centralidade[i].valor = 1 / centralidade[i].valor;
182
183   }
184
185   return orientado;
186
187 }
188
189 void Deslocar( int posicao_atual, int posicao_nova, int* ordem, int total ) {
190
191   int valor = ordem[posicao_atual];
192
193   if( posicao_atual <= posicao_nova ){
194     for( int i = posicao_atual; i < posicao_nova; i++ ) ordem[i] = ordem[i+1];
195   }else{
196     for( int i = posicao_atual; i > posicao_nova; i-- ) ordem[i] = ordem[i-1];
197   }
198
199   ordem[posicao_nova] = valor;
200
201 }
202
203 long Dispersao( int** adjacencia, int* ordem, int total, bool orientado ) {
204
205   long dispersao = 0;
206   int i, j;
207
```

```
208   if( ! orientado ){
209     for( i = 0; i < (total-1); i++ ){
210       for( j = i+1; j < total; j++ ){
211         if( adjacencia[ordem[i]][ordem[j]] != DESCONEXO ){
212           dispersao += (j-i);
213         }
214       }
215     }
216   }else{
217     for( i = 0; i < total; i++ ){
218       for( j = 0; j < total; j++ ){
219         if( adjacencia[ordem[i]][ordem[j]] != DESCONEXO ){
220           dispersao += (i>=j) ? (i-j) : (j-i);
221         }
222       }
223     }
224   }
225
226   return dispersao;
227
228 }
229
230 void ImprimirResultado( int* ordem, int total, ostream* saida ) {
231   *saida << ordem[0] + 1;
232   for( int i = 1; i < total; i++ ) *saida << ',' << ordem[i] + 1;
233 }
234
235 int CompararCentralidade( const void* a, const void* b ) {
236   return ( ((Centralidade*)a)->valor - ((Centralidade*)b)->valor );
237 }
238
239 inline int Sortear( int maximo ) {
240   return rand() % ( maximo + 1 );
241 }
```

APÊNDICE D – Código Fonte do Cost Function Method (CFM)

O código fonte em C++, a seguir, corresponde à implementação do algoritmo Cost Function Method (CFM), conforme especificação de Rybarczyk-Filho et al. (2011). Ele foi adaptado para registrar dados comportamentais para fins de estatística e de comparação com o algoritmo Claritate, e também está disponível publicamente no endereço Web <https://github.com/joseflaviojr/cfm>, sob a Licença Pública Menos Geral do GNU <http://www.gnu.org/licenses/>.

Código Fonte D.1: Código fonte do Cost Function Method (CFM), implementado na linguagem de programação C++.

```
1
2  int CFM( int** matrizadj, int total, ofstream* saida ) {
3
4    // Inicializacao
5
6    int ordem[total];
7    int ordem_melhor[total];
8    for( int i = 0; i < total; i++ ) ordem[i] = ordem_melhor[i] = i;
9
10   long custo, custo_novo, custo_melhor;
11   custo = Custo( matrizadj, ordem, total );
12   custo_melhor = custo;
13
14   double temperatura = 0.0001 * custo;
15   double temperatura_reducao = 0.8;
16   int temperatura_passos = 100;
17
18   // Monte Carlo
19
20   int passo = 1, sorteio = 0;
21   int s1, s2;
22
23   while( custo_melhor > 0 ){
24
25     if( ++sorteio > total ){
26       passo++;
27       if( ( passo % temperatura_passos ) == 0 ) temperatura *= temperatura_reducao;
28       sorteio = 1;
29     }
30
31     s1 = Sortear(total);
32     s2 = Sortear(total);
33
34     Permutar( ordem, total, s1, s2 );
35
36     custo_novo = Custo( matrizadj, ordem, total );
```

```
37
38      if( custo_novo < custo || Sortear(1) <= exp((custo-custo_novo)/temperatura) ){
39
40        custo = custo_novo;
41
42        if( custo < custo_melhor ){
43
44          custo_melhor = custo;
45          memcpy( ordem_melhor, ordem, total * sizeof(int) );
46
47          saida->seekp( 0, saida->beg );
48          ImprimirResultado( ordem, total, saida );
49          saida->flush();
50
51        }
52
53      }else{
54        Permutar( ordem, total, s1, s2 );
55      }
56
57    }
58
59    return 0;
60
61  }
62
63  long Custo( int** matrizadj, int* ordem, int total ) {
64
65    long custo = 0;
66
67    const int ultimo = total - 1;
68    int i, j, valor;
69
70    for( i = 0; i < ultimo; i++ ){
71      for( j = i+1; j < total; j++ ){
72        valor = 0;
73        if( matrizadj[ordem[i]][ordem[j]] ){
74          if( i != ultimo ) valor += 1 - matrizadj[ordem[i+1]][ordem[j  ]];
75          if( j != ultimo ) valor += 1 - matrizadj[ordem[i  ]][ordem[j+1]];
76          if( i != 0 )      valor += 1 - matrizadj[ordem[i-1]][ordem[j  ]];
77          if( j != 0 )      valor += 1 - matrizadj[ordem[i  ]][ordem[j-1]];
78        }else{
79          if( i != ultimo ) valor += matrizadj[ordem[i+1]][ordem[j  ]];
80          if( j != ultimo ) valor += matrizadj[ordem[i  ]][ordem[j+1]];
81          if( i != 0 )      valor += matrizadj[ordem[i-1]][ordem[j  ]];
82          if( j != 0 )      valor += matrizadj[ordem[i  ]][ordem[j-1]];
83        }
84        custo += abs(i-j) * valor;
85      }
86    }
87
88    return custo;
89
90  }
91
92  void Permutar( int* ordem, int total, int a, int b ) {
93    int x = ordem[a];
```

```
94    ordem[a] = ordem[b];
95    ordem[b] = x;
96  }
97
98  void ImprimirResultado( int* ordem, int total, ostream* saida ) {
99    *saida << ordem[0] + 1;
100   for( int i = 1; i < total; i++ ) *saida << ',' << ordem[i] + 1;
101 }
102
103 inline double Sortear( int limiteNaoIncluso ) {
104   return ( ( rand() % 100 ) / 100.0 ) * limiteNaoIncluso;
105 }
```