UNIVERSIDADE DE SÃO PAULO
NOME DA UNIDADE USP

Ian Berthold de Haan

Ontologias Gênicas Construídas por Redes de Interação

São Carlos

2021

Ian Berthold de Haan

Ontologias Gênicas Construídas por Redes de Interação

Monografia apresentada ao Curso de Ciências Físicas e Biomoleculares, da Unidade Instituto de Física de São Carlos da Universidade de São Paulo, como parte dos requisitos para conclusão do curso de Ciências Físicas e Biomoleculares.

Orientador: Prof. Dr. Luciano da Fontoura Costa

São Carlos

2021

RESUMO

HAAN, I. **Ontologias Gênicas Construídas por Redes de Interação**. 2021. 26p. Monografia (Trabalho de Conclusão de Curso) - Nome da Unidade USP, Universidade de São Paulo, São Carlos, 2021.

O presente trabalho busca fazer uma revisão da literatura no que tange a utilização de métodos para extração de ontologias no contexto da biologia a partir de redes de Dados. O trabalho inclui progresssão histórica, detalhamento dos métodos utilizados, resultado comparativo sobre os métodos, o estado da arte e promessas para o futuro.

Palavras-chave: Ontologia. GO. Gene Ontology. NeXO. CliXO.

SUMÁRIO

1 INTRODUÇÃO

1.1 O que são ontologias?

Para a filosofia, ontologias são o estudo dos tipos de coisas que existem. Não obstante, ao longo do tempo, a palavra recebeu um segundo significado, mais ou menos relacionado ao original. Ontologias, no sentido tratado nesse trabalho, derivado da área de inteligência artificial, são vocabulários destinados à representação do conhecimento disponível sobre determinado assunto (1).

Guarino (1998) tenta trazer uma definição para esse novo uso da palavra:

"[...] ontologia se refere a um artefato constituído por um vocabulário usado para descrever uma certa realidade, mais um conjunto de fatos explícitos e aceitos que dizem respeito ao sentido pretendido para as palavras do vocabulário. Este conjunto de fatos tem a forma da teoria da lógica de primeira ordem, onde as palavras do vocabulário aparecem como predicados unários ou binários."(2)

Entretanto, essa definição, mesmo trazendo diversos aspectos realmente presentes em ontologias, não é consensual. Por conseguinte, para o correto entendimento da essência do termo, faz-se necessário exemplificar e elencar características comuns de ontologias. (3)

Os componentes básicos de uma ontologia são classes (conceitos do domínio em questão), juntamente com suas relações taxonômicas e propriedades. Quando adicionadas instâncias a essas classes cria-se uma base de conhecimento. (4)

Um exemplo concreto pode ser facilmente visualizado com uma ontologia hipotética criada para representar o conhecimento disponível sobre vinhos. Classes poderiam ser branco, espumante, rosé, tinto, Cabernet Sauvignon, Chardonnay etc., as relações entre as classes são, por exemplo, Cabernet Sauvignon é um *tipo de* branco que, por sua vez é um *tipo de* da superclasse vinho.

Já propriedades são, por exemplo, *produzido com uvas verdes* e *baixa quantidade de polifenóis*, características da classe branco. Essas propriedades são utilizadas, então, para instanciar exemplares, garrafas individuais de vinho, formando uma base de conhecimento.

Através dessa exemplificação fica claro o poder de um constructo como esse, ontologias conseguem sistematizar, de maneira clara e eficiente, conhecimentos das mais diversas áreas; linguística, ciências da computação e biologia são apenas alguns exemplos dos muitos domínios que se utilizam desse conceito.

1.2 A GO

Com o advento do sequenciamento de genomas inteiros se fez necessária uma ontologia que captasse todo o conhecimento disponível sobre os genes dos organismos. Tal necessidade é não apenas organizacional, como também fruto do desafio de comparar e transferir anotações entre diferentes espécies. Para tal, em 2000 criou-se a Gene Ontology (GO).(5)

A GO é, na realidade, um conjunto de três ontologias distintas que contém termos relacionados a processo biológico, componente celular e função molecular. A primeira carrega termos como *DNA repair* (reparo de DNA) e *signal transduction* (transdução de sinal). Já a ontologia de componente celular traz classes como *ribosome* (ribossomo) e *mitochondrion* (mitocôndria). Por fim, a GO de função molecular contém termos como *adenylate cyclase activity* (atividade de adenilato ciclase) e *transporter activity* (atividade de transporte).(5) (6)

O objetivo da iniciativa é produzir um vocabulário estruturado, bem definido e comum para descrever os papéis dos genes e produtos gênicos de diversos organismos. Para esse fim, criou-se uma estrutura padrão que a GO deve respeitar. As ontologias são estruturadas como grafos acíclicos dirigidos (DAG, do inglês Directed Acyclic Graph), ou seja, um grafo dirigido no qual, caminhando no sentido das ligações, é impossível sair de um vértice e voltar para ele fechando um ciclo. No grafo os termos são respresentados por nós e as relações entre eles como arestas.(5) (6)

Os termos presentes na GO (com excessão dos termos raiz) possuem uma relação de *is a* (é um) com alguma subclasse, por exemplo, glucose transport is a monosaccharide transport (transporte de glicose é um transporte de açúcar). Porém, além dela, outras relações são frequentemente empregadas na ontologia, como *part of* (parte de), *has part* (tem parte), *regulates* (regula), *positively regulates* (regula positivamente) e *negatively regulates* (regula negativamente).(6)

Entretanto, o que começou como um projeto destinado principalmente para fins anotacionais hoje já representa muito mais, ela é essencial, seja para um biólogo realizando trabalho de base em um aspecto específico de um único organismo até para físicos e geneticistas que se propõem a compreender polimorfismos genéticos humanos, além de servir como padrão ouro para medir o sucesso de métodos de bioinformática.(7) Além disso, ela já serve de base para iniciativas muito além do que o imaginado inicialmente, como um modelo de aprendizado profundo capaz de modelar com precisão o crescimento celular. (8)

1.3 A NeXO

Mesmo com a GO crescendo vertiginosamente em importância, em meados de 2010 ela enfrentava sérios desafios relativos a anotação. Nessa época, ela crescia vertiginosamente também em tamanho e complexidade, já apresentando mais de 30000 termos e 60000 relações entre termos, com genes anotados de mais de 80 espécies.(9) (10)

Além de problemas inatos da subjetividade humana na anotação, os quais podem ser parcialmente contornados com padrões estritos nessa tarefa, as ontologias encontravam dificuldade em traduzir novas descobertas de domínio específico para a relação de termos da GO, produzindo viés em favor de termos mais bem estudados. (9) (10)

Tendo esses problemas em vista, imaginou-se que uma possível solução seria inferir ontologias a partir de dados experimentais. A ideia era inferir uma hierarquia entre clusters de redes de interação que seja análoga, em alguma escala, à GO.(10)

Em virtude de resultados iniciais que demonstravam maior densidade de interação gênica presentes dentro de um mesmo termo da GO em quatro tipos de redes de interações, sendo elas físicas proteína-proteína, gênicas (epitasia e letalidade sintética), co-expressão gênica e interação funcional (YestNet), desenvolveu-se um método para montar ontologias a partir de redes biológicas e ele foi batizado de *Network Extracted Ontology* (NeXO). (10)

O procedimento começa com o emprego de um método probabilístico de detecção de comunidades originalmente desenvolvido para a predição de relações ausentes em uma rede. Tal método, explicado com maior detalhamento posteriormente, constrói uma árvore binária que maximiza as chances de recriação de todas as redes de interação empregadas. Como elas presumidamente possuem padrões de interação que refletem as relações hierárquicas intrínsecas da células, espera-se que esse dendograma também o faça. Entretanto, uma árvore binária é apenas uma aproximação das relações hierárquicas entre genes, não possuindo a capacidade de refletir a real complexidade destas. (10) (11) (12)

O problema é que um dendograma impõe a restrição artificial de que todos os termos (com excessão da raiz e nós terminais) possuam extamente dois termos mais específicos abaixo dele e um mais geral acima, o que não reflete a realidade celular, na qual um processo pode ter mais de dois componentes envolvidos e um mesmo componente pode participar de mais de um processo, realidade bem representada pela GO. Para transformar essa primeira aproximação em algo mais realista, procuram-se por novas ligações que possam ser criadas na árvore binária para aumentar a probabilidade de reconstrução das redes.(10)

Nesse estágio, cada termo da ontologia possui apenas uma anotação numérica, a qual, sem uma análise, não significa nada. Para completar a ontologia os autores desenvolveram um algoritmo de alinhamento de ontologias baseado em métodos derivados das ciências congnitivas e da computação que faz o match entre termos baseado em semelhança entre

instâncias e posicionamento na hierarquia. Os objetivos dessa etapa são três: transferência de termos presentes na GO, identificação de termos não presentes na GO (o que viria a ser um dos mais importantes resultados conseguidos) e identificação de relações conflitantes entre termos.(10)

Utilizando essa metodologia criou-se uma NeXO de levedura (Figura 1). Alinhando ela com as ontologias da GO para a espécie, pôde-se reconstruir 60% da ontologia de componentes celulares e ao redor de 25% das outras duas, com termos que possuem bom suporte nas redes apresentando melhor alinhamento, o que sugere que, com uma melhora nos dados experimentais, é possível melhorar a ontologia gerada, o que foi testado pelos autores e confirmado. (10)

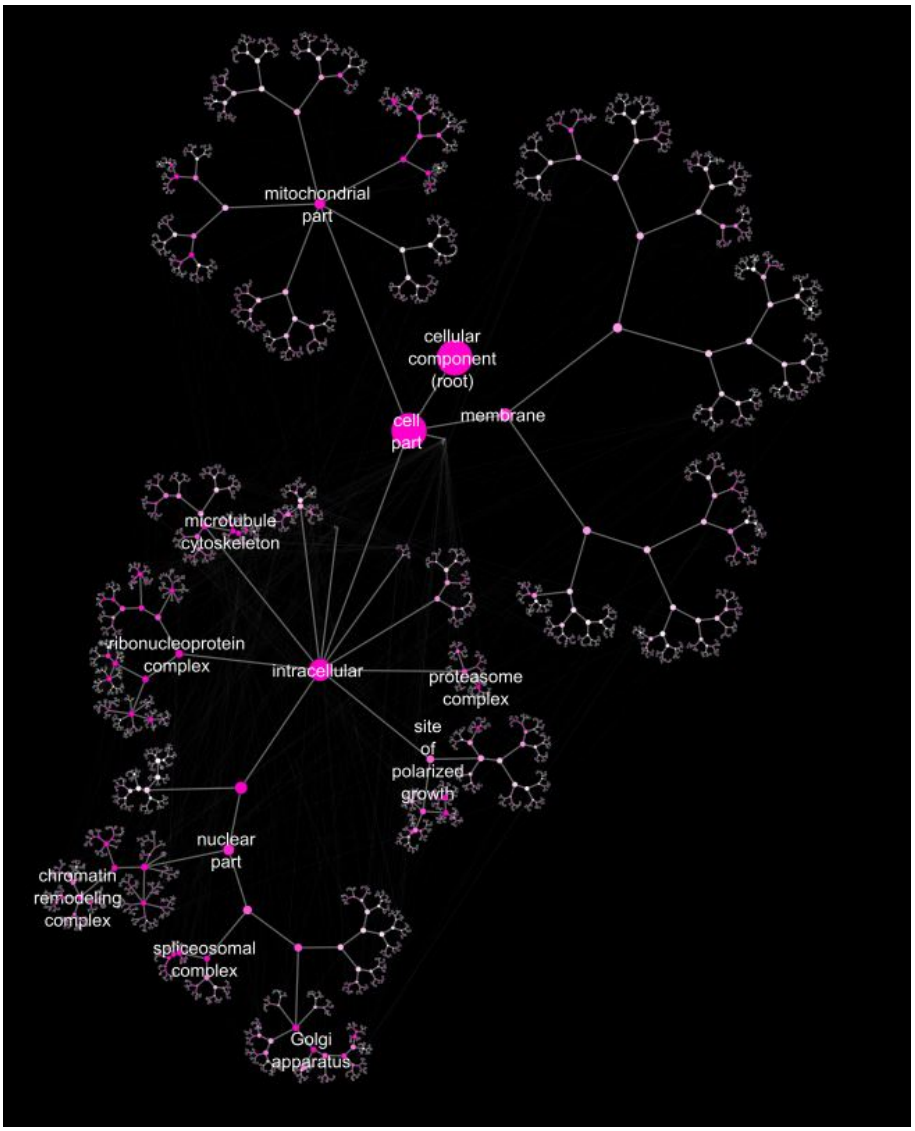

Figura 1 – NeXO

Fonte: Estrutura retirada de http://nexontology.org/ e representada através do Cytoscape.

Dessa forma, a partir da NeXO pôde-se saltar do uso de ontologias para análise de

dados para a utilização de dados na criação e avaliação de ontologias. Um salto, ainda em curso, que pode ter consequências profundas na forma de se fazer biologia.(7) (10) (13)

1.4 CliXO

Motivados pelos resultados promissores da NeXO os autores consideraram diversos métodos de clusterização hierárquica que poderiam ser utilizados para a reconstrução da GO e possivelmente trariam resultados melhores. Os principais pontos considerados foram a possibilidade da utilização de valores discretos nas redes de interação, uma vez que na NeXO foram utilizados *thresholds* para a transformação de redes ponderadas em não ponderadas, pela impossibilidade do uso das primeiras nesse método, o que resulta em perda de informação; e criação de DAGs com relações de parentesco múltiplas permitidas desde um primeiro momento, o que poderia gerar relações mais precisas.(13)

O único algoritmo encontrado pelos autores que realizaria a tarefa desejada é o LocalFitness, esse método cria *clusters* em algum nível da hierarquia por otimização de uma função de *fitness* para cada um dos possíveis *clusters* sobrepostos contruídos a partir de vários nós. A função fitness inclui um parâmetro que é ajustado para encontrar *clusters* nos múltiplos níveis, as partições do grafo que são robustas, ou seja, estáveis em um intervalo considerável do parâmetro, são utilizadas. (13)

Além do LocalFitness também foi apresentado um novo método, denominado Clique Extracted Ontology (CliXO). O método, explicado em detalhes na seção 2.3 é baseado no conceito de cliques, derivado da teoria de grafos. Além de atingir uma imensa precisão na reconstrução da GO utilizando de similaridade semântica ($> 98\%$ de termos identicamente alinhados) como prova de conceito, o método performa melhor que outros algoritmos de clusterização e apresenta resultados similares aos da NeXO no que tange a reconstrução da GO através de redes de interação. O método apresenta, ainda, uma forma explícita de se lidar com o ruído, um dos problemas fundamentais da tarefa. (13)

2 MÉTODOS

2.1 Grafos

Um grafo é constituído por um conjunto de nós (ou vértices) e um conjunto de conexões (ou arestas). Um grafo é dito dirigido se suas conexões apresentam direcionamento, ou seja, podem ser percorridas em apenas um sentido e denominado não dirigido se podem ser percorridas nos dois. Um grafo dirigido é dito acíclico se, partindo de um nó e caminhando no sentido das ligações é impossível voltar para o mesmo nó, grafos dirigidos acíclicos são abreviados como DAG (Figura 2).(14)

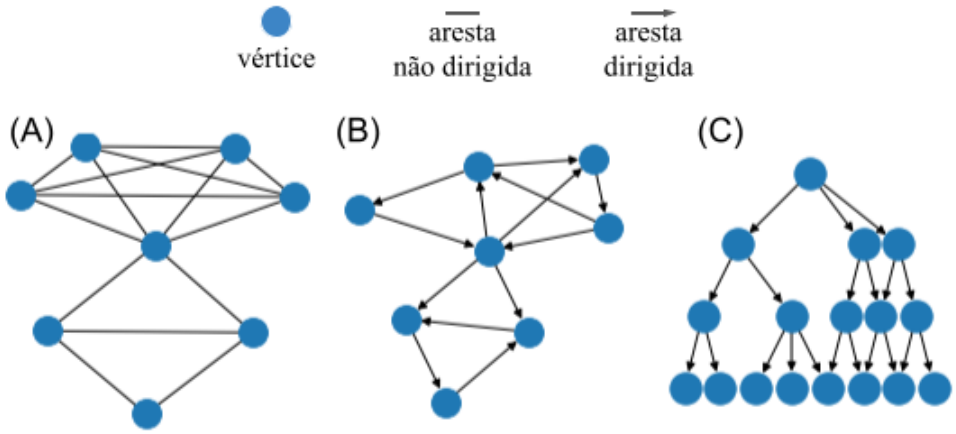

Figura 2 – Exemplos de grafos: (A) um grafo não dirigido; (B) um grafo dirigido; (c) um grafo dirigido acíclico

2.2 Construção da NeXO

Como dito de antemão na seção 1.3 o método para a construção da NeXO baseia-se em três etapas, são elas:

1. Criação de uma árvore binária através de um método probabilístico de clusterização hierárquica
2. Adição de relações entre termos para permitir relações múltiplas de parentesco
3. Alinhamento da ontologia gerada com a GO

2.2.1 Criação da Árvore Binária

O método para a criação da árvore binária possui uma grande complexidade, tendo isso em vista, faz-se adequada uma primeira apresentação dele em um análogo não

hierárquico. A explicação aqui contida é inspirada no artigo no qual a metodologia foi proposta (12) e fundida com os detalhes particulares da NeXO (10).

2.2.1.1 Um modelo plano

Considera-se um grafo G, definido por um conjunto de vértices (V) e arestas (E, do inglês edges). Um modelo plano M define como esses vértices se juntam em grupos, os quais são denotados $[C_1, C_2, ..., C_k]$ para um modelo com K grupos, cada vértice é associado a um e apenas um grupo, os quais são, portanto, disjuntos. Índices i,j são utilizados para grupos, enquanto u e v para vértices.

A contagem de arestas entre grupos e dentro de um mesmo grupo pode ser contabilizada, respectivamente, por $e_{ij} = \sum_{u \in i, v \in j} e_{uv}$ e $e_{ii} = \sum_{u<v \in i} e_{uv}$, sendo e_{uv} 1 para um vértice entre u e v e 0 para a ausência (presença de um buraco). Já a contagem de pares de vértices totais é, na mesma ordem, $t_{ij} = n_i n_j$ e $t_{ii} = n_i(n_i - 1)/2$, sendo n_i o número total de vértices no grupo i. Por conseguinte, a contagem de buracos entre i e j, iguais ou diferentes, é $h_{ij} = t_{ij} - e_{ij}$.

Para um dado par de grupos i e j, as arestas e_{ij} são modeladas a parir de t_{ij} ensaios de Bernoulli com parâmetro θ_{ij}. A probabilidade das arestas observadas, condicionadas a t_{ij} é(15):

$$P(\theta_{ij}) = \theta_{ij}^{e_{ij}}(1 - \theta_{ij})^{h_{ij}} \tag{2.1}$$

O valor de maior verossimilhança (P_{ij}^{ML}) é obtido pela utilização da estimativa de máxima verossimilhança de θ_{ij}, $\hat{\theta} = e_{ij}/t_{ij}$, com uma probabilidade a priori (prior) uniforme. Uma probabilidade verdadeiramente bayesiana (P_{ij}^{FB}) é obtida através do processo de marginalização, marginalizando sobre o parâmetro θ_{ij}, novamente com um prior uniforme (15):

$$\begin{aligned} P_{ij}^{ML} &\equiv e_{ij}^{e_{ij}} h_{ij}^{h_{ij}} / t_{ij}^{tij} \\ P_{ij}^{FB} &\equiv Beta(e_{ij} + 1, h_{ij} + 1) \end{aligned} \tag{2.2}$$

Dessa forma, para um modelo plano, com $K(K+1)/2$ parâmetros, a verossimilhança e probabilidade bayesiana são(15):

$$\begin{aligned} L(M) &= \prod_{i \leq j} P_{ij}^{ML} \\ P(M) &= \prod_{i \leq j} P_{ij}^{FB} \end{aligned} \tag{2.3}$$

2.2.1.2 Generalização Para um Modelo Hierárquico

Um grafo randômico hierárquico (HRG, do inglês hierarchical random graph), é um dendograma em que cada nó tem uma probabilidade associada (p_r), a qual corresponde com a probabilidade de um vértice entre as suas subárvores filhas estarem ligados na rede sendo representada. Esse modelo permite a recriação de uma rede com propriedades topológicas relacionadas a original, o que da margem para utilizar uma abordagem de máxima verossimilhança para detectar a estrutura hierárquica. (11)

Essa abordagem pode ser utilizada para extender a noção do modelo plano M (subseção 2.2.1.1) para um modelo hierárquico T. Utilizando o mesmo modelo probabilístico introduzido naquela seção, define-se $e_{c1,c2}$ e $h_{c1,c2}$ como sendo, respectivamente, o número de arestas e buracos entre as subárvores $c1$ e $c2$ de um nó. De maneira similar à Equação 2.3, fazendo $M \equiv T$, a verossimilhança $L(T)$ de um dado modelo T e a probabilidade $P(T)$ da rede dado o modelo são:

$$\begin{aligned} L(T) &= \prod_{c1,c2 \in nós(T)} P^{ML}_{c1,c2} \\ P(T) &= \prod_{c1,c2 \in nós(T)} P^{FB}_{c1,c2} \end{aligned} \tag{2.4}$$

Onde $P^{ML}_{c1,c2} = e^{e_{c1,c2}} h^{h_{c1,c2}}_{c1,c2} / t^{tc1,c2}_{c1,c2}$ e $P^{FB}_{c1,c2} = Beta(e_{c1,c2}+1, h_{c1,c2}+1)$. Sendo $Beta$ a função de mesmo nome, pela definição dela fica claro que a probabilidade $P(M)$ da rede dado o modelo é maximizada se as relações entre os nós de C1 e C2 forem coerentes, isso é, forem predominantemente buracos ou arestas.

Para um conjunto de redes R a serem representadas por um único modelo T, $P(T)$ e $R(T)$ são dados pelo produto das probabilidades e verossimilhanças das $r \in R$ redes individuais:

$$\begin{aligned} P(T) &= \prod_{r \in R} P_r(T) \\ L(T) &= \prod_{r \in R} L_r(T) \end{aligned} \tag{2.5}$$

Fazendo uma otimização de máxima verossimilhança sobre o conjunto de dados a serem utilizados para a ciação da ontologia, encontra-se uma árvore binária, a qual é utilizada para uma primeira representação hierárquica da rede.

2.2.2 Generalização para relações múltiplas de parentesco

Embora essa árvore binária sirva para uma primeira representação da rede, ela não da conta de toda a complexidade intrínseca da estrutura hierárquica dos genes (seção 1.3). Dessa forma, modifica-se o modelo T para permitir relações de parentesco múltiplas.

Primeiramente, calcula-se o quanto cada nó de T contribui para a pontuação geral do modelo, nós que não contribuem são removidos e seu nó parental é conectado diretamente a cada um de seus nós filhos. A contribuição de um nó p é calculada pela razão entre a probabilidade dos dados sob a árvore original e sob uma árvore atualizada, na qual p é substituído por cada um de seus filhos $(c1, c2, ..., cn)$:

$$\lambda_{c1,c2,...,cn} = \prod_{s=1}^{K} \frac{P_{p,s}}{P_{c1,s} P_{c2,s} \cdots P_{cn,s}} \tag{2.6}$$

Onde s representa cada um dos K nós irmãos de p e as probabilidades são calculadas como descrito anteriormente (Equação 2.4). O nó p é removido se tanto $\lambda_{c1,c2,...,cn} < 1$ como a densidade de interação $(D_{ij} \equiv e_{ij}/t_{ij})$ entre $c1, c2, ..., cn$ não for maior que entre p e seus irmãos. Com a remoção de nós, são criadas relações de parentesco múltiplo.

Além disso, a fim de permitir a existência de nós com mais de um nó parental, é montada uma heurística. Começando das folhas, consideram-se todos os pares de nós (c,p) tais que o número de genes associados a c é menor do que o de associados a p. O nó p é identificado como um parente adicional de c se:

1. Os nós c e p não estão no mesmo caminho.

2. O padrão de interações entre os genes associados e a c e associados a p é denso $(e_{pc}/t_{pc} > 0.3)$.

3. O conjunto de genes associados a p unidos com os associados a c formam um cluster denso $(e_{p \cup c, p \cup c}/t_{p \cup c, p \cup c} \geq \frac{1}{2} e_{p,p}/t_{p,p})$

2.2.3 Alinhamento de Ontologias

Como citado anteriormente (seção 1.3), até essa etapa os termos da NeXO são apenas *tags*, os quais só possuirão um real significado caso a estrutura hierárquica seja analisada. O alinhamento de ontologias permite não apenas aproveitar a anotação existente da GO para essa tarefa como também comparar as ontologias e encontrar possíveis erros na GO. A técnica de alinhamento utilizada para a confecção da NeXO é inspirada por um método denominado ASMOV34, criado para o alinhamento de ontologias semânticas. (16)

Dado duas ontologias, O1 com n1 termos e 02 com n2 temos, um alinhamento de ontologias A é um mapeamento de termos entre as ontologias de forma que cada termo em 01 é mapeado para no máximo um termo em 02 e vice-versa. O mapeamento entre termos no alinhamento é avaliado usando uma função pontuação que considera a similaridade dos conjuntos de genes associados aos termos (similaridade intrínseca entre termos) e a posição relativa dos termos na hierarquia (similaridade relacional).

O alinhamento é um processo iterativo, com cada iteração k produzindo um alinhamento A_k. O processo começa com o cálculo de uma matriz T_k de dimensões $n1 \times n2$, onde $0 \leq T_k(i,j) \leq 1$ representa a similaridade do termo $i \in O1$ com $j \in O2$. T_k é contabilizado da seguinte maneira:

$$T_k(i,j) = \begin{cases} I(i,j), \text{para } k = 0 \\ 0.75I(i,j) + 0.25R_k(i,j), \text{para } k > 0 \end{cases} \tag{2.7}$$

Onde $I(i,j)$, a similaridade intrínseca, é precomputada, uma vez que não depende de k, e é dada pelo Índice Jaccard:

$$I(i,j) \equiv \frac{x_i \cap x_j}{x_i \cup x_j} \tag{2.8}$$

Já $R_k(i,j)$, a similaridade relacional, é calculada pela semelhança entre os conjuntos de termos que são parentes de i e j (P_i, P_j) e entre os que são filhos dos mesmos (C_i, C_j):

$$R_k(i,j) = \begin{cases} \dfrac{S(P_i,P_j) + S(C_i,C_j)}{2}, \text{para nós internos} \\ S(C_i,C_j), \text{para raiz} \end{cases} \tag{2.9}$$

Onde as similaridades (s) entre conjuntos são calculadas utilizando A_{k-1} da seguinte forma:

$$\begin{aligned} S(X,Y) &= \frac{SOS}{|X| + |Y| - SOS} \\ SOS &= \sum_{(x,y) \in L} T_{k-1}(x,y) \end{aligned} \tag{2.10}$$

L é o alinhamento local de X com Y, determinado pela escolha de pares (x,y) com o maior T_{k-1}, assegurando-se que nenhum elemento de X ou Y participe em mais de um par. Tendo em vista o que foi apresentado nessa seção, o alinhamento A_k é encontrado através do seguinte algoritmo míope:

1. Inicializar o A_k sem correspondência entre termos; Calcular T_k; inicializar L como uma lista ordenada de pares de termos (i,j) em ordem decrescente de $T_k(i,j)$

2. Selecionar o primeiro par (i,j) de L

3. Confirmar se (i,j) conflita com algum dos pares já contidos em A_k. Dois pares, (i,j) e (i',j'), conflitam se:

   (a) $i = i' \vee j = j'$

(b) $((i$ é descendente de $i') \wedge (j$ é ancestral de $j')) \vee$
$((i$ é ancestral de $i') \wedge (j$ é descendente de $j'))$

4. Se não houver conflito, adicionar (i, j) a A_k

5. Se todos os termos de O1 ou O2 estiverem mapeados, ou se todos os pares tiverem uma similaridade abaixo de um valor limite (definido como 0.01), então A_k está completo. Senão, retornar para o passo 2.

6. Se $A_k = A_i, i \in (0, ..., k)$, terminar, caso contrário, reiniciar para a próxima iteração $(k = k + 1)$

Com o fim do algoritmo, a pontuação $(S_k(t))$ de cada um dos termos alinhados é calculada como:

$$S_k(t) = \begin{cases} T_k(t, A_k(t)), \text{para termos mapeados por } A_k \\ 0, \text{para termos não mapeados por } A_k \end{cases} \tag{2.11}$$

A taxa de alinhamentos falsos é calculada como:

$$FDR(t) = \frac{\frac{1}{n}\sum_{i=1}^{n} N_{R_i}(t)}{N(t)} \tag{2.12}$$

Onde N_{R_i} é o número de termos em um alinnhamento feito por permutação randômica que possuem uma pontuação de alinhamento $\geq$ t. Um limiar de pontuação de alinhamento mínima para o mesmo valer é 0.1 e a mesma é reduzida com o tamanho dos grupos de forma a manter $FDR < 10\%$ para todos os tamanhos de grupos.

2.3 Construção da CliXO

2.3.1 Cliques

A construção da CliXO é baseada em um conceito da teoria de grafos denominado clique. Um clique de um grafo não orientado U é definido como um subconjunto de seus vértices em que todos os pares estão conectados entre sí. Um clique é dito maximal quando é impossível adicionar outro vértice e encontrar um clique maior, ou seja, o clique maximal não é subconjunto de nenhum outro clique.

2.3.2 Definições

A metodologia e explicação sobre ela foram baseadas no artigo em que a CliXO foi apresentada (13).

Define-se o grafo que representa a ontologia como um DAG ponderado $G = [T, N, E, w, r]$ com as seguintes propriedades:

1. G tem dois tipos de nós, terminais (T; sem nós filhos) e não terminais (N), os quais também são denominados termos. G tem uma única raiz (r), a partir da qual todos os nós podem ser alcançados.
2. $|G|$:= número de nós em N
3. Entre qualquer par de nós $(a, b) \in T$, $w(a, b)$ denota o menor caminho entre a e b em G.
4. **Propriedade Ultramétrica**: um nó $u \in N$ tem distância constante ($w(u)$) para todos os seus descendentes terminais ($L(u)$).
5. **Propriedade de Testemunha**: para um nó $u \in N$, e um nó $v(\in T \wedge \notin L(u))$, $\exists b \in L(u)|w(a, b) > 2w(u)$.
6. $\forall(a, b) \in T \exists$ um menor ancestral comum ($lca(a, b)$) $|(w(a, b) = 2w(lca(a, b)))$

2.3.3 Formalização do problema

Para otimizar-se a resolução de um problema, faz-se útil primeiro entender com detalhes do que se trata o problema a ser resolvido. Com a seguinte formalização, o entendimento sobre o algoritmo e sua motivação é facilitada.

2.3.3.1 Caso Perfeito

Entrada - Um conjunto de nós terminais T e uma matriz M de distância entre os pares de nós, (uma matriz similaridade pode ser convertida para uma distância fazendo M =constante - $M_{similaridade}$)

Saída - G, com T como o conjunto de nós terminais e $\forall(a, b) \in T, w(a, b) = M(a, b)$

Como raramente as distâncias de entrada satisfazem as distâncias da ontologia perfeitamente, no caso imperfeito calcula-se a ontologia que melhor representa M

2.3.3.2 Caso imperfeito

Entrada - Um conjunto de nós terminais T, uma matriz M de distância entre os pares de nós e um parâmetro de ruído α fornecido pelo usuário.

Saída - G, com T como o conjunto de nós terminais que maximiza |G| enquanto satifaz as seguintes condições:

1. $\forall(a, b) \in T, w(a, b) \geq M(a, b)$
2. $\forall u \in N, \forall a(\in T \wedge \notin L(u)), \exists b \in L(u)|(M(a, b) > 2w(u) + \alpha)$
3. $\forall u \in N, \exists(a, b) \in L(u)|2w(u) + \alpha < w(v), \forall v$ com $(a, b) \in L(v), v \neq u$
4. $\forall u \in N, (a, b) \in L(u), w(u) = 2max(M(a, b))$

2.3.4 O Algoritmo CliXO

2.3.4.1 Caso perfeito

Considera-se um grafo U não dirigido, com nós $\in T$ e sem arestas. Sendo S o ordenamento de todos os pares de nós (a,b) em ordem decrescente de distância (M(a,b)), encontra-se a ontologia pela seguinte heurística:

Input : Ordenamento dos pares de nós (S)
Output : Nós não terminais de G (C_G)
$C_g \leftarrow \{\}$
while $S \neq \{\}$ **do**
  $(a, b) \leftarrow S[0]$
  $t \leftarrow M(a, b)$
  while $M(a, b) = t$ **do**
    $(a, b) \leftarrow Pop(S)$
    Adicionar aresta (a,b) a U
  end
  $C_{cur} \leftarrow$ conjunto de cliques maximais em U
  $C_G \leftarrow (C_g \cup C_{cur})$
end

A cada saída, os cliques maximais entre os nós terminais correspondem a um nó sendo criado no grafo da ontologia, gerando a hierarquia. (Figura 3,Figura 4)

	B	C	D	E	F	G	H
A	0.75	0.25	0.25	0.25	0.0	0.0	0.0
	B	0.25	0.25	0.25	0.0	0.0	0.0
		C	0.6	0.6	0.0	0.0	0.0
			D	0.6	0.0	0.0	0.0
				E	0.6	0.5	0.0
					F	0.75	0.5
						G	0.6

Figura 3 – Exemplo de matriz de distâncias

2.3.4.2 Caso imperfeito

No caso imperfeito a única mudança no algoritmo é, ao invés de adicionar todos os cliques de C_{cur} para C_g a cada vez que um novo valor de *threshold* é atingido, adiciona-se a C_G apenas cliques $C \in C_{cur}$ para os quais $max_{a,b \in C}(M(a, b)) < t - \alpha$.

O algorimo, construído dessa maneira, consegue distinguir perfeitamente entre sinal e ruído quando $\alpha < s \wedge \alpha > n(G)$, sendo s a menor distância entre qualquer par de nós conectados na ontologia e n(g) determinado pelo seguinte procedimento. Para

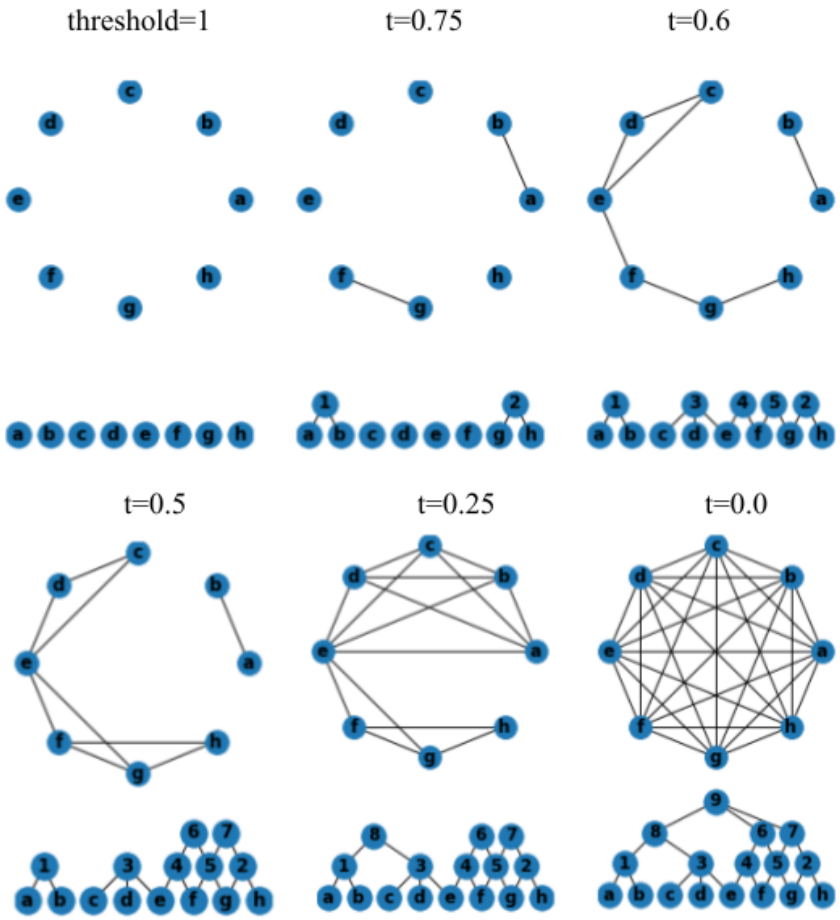

Figura 4 – Exemplo de construção da hierarquia pelo método CliXO a partir da matriz da Figura 3; acima as iterações de U e abaixo as iterações da hierarquia

Fonte: Elaborado pelo autor a partir do NetworkX.

cada termo $u \in G$, ordena-se pelo valor todos os $M(a, b)$, onde $a, b \in u$, $n(u)$, para um nó não terminal $u \in N$, é a diferença máxima entre valores adjacentes na lista, por fim, $n(G) = max_{u \in N} n(u)$.

Não é possível determinar $n(G)$ sem saber a verdadeira estrutura da ontologia, entretanto é um conceito útil para entender o papel de α. $n(G)$ pode ser estimado, contudo, como 2x o erro padrão das distâncias medidas e α pode ser estimado com esse valor.

Nem sempre é possível definir α de tal forma que $n(G) < \alpha < s$, porque às vezes $n(G) > s$. Nesse caso, se $\alpha < n(G)$, criar-se-á termos estranhos que são um subconjunto de termos 'reais'. Por outro lado, se $\alpha > s$, então qualquer nó filho que for próximo a seu nó parental $(2(w(p) - w(c)) < \alpha))$ não será incluído na ontologia. Uma estratégia para definir o valor de α de maneira satisfatória é analisar uma parte conhecida da ontologia, se muitos termos estranhos estão sendo criados, então provavelmente α está baixo, se vários termos estão colapsados, então α está alto.

2.3.4.3 Falsos negativos

Em dados experimentais conexões faltando (medições em que $M(a,b) << w(a,b)$ na ontologia 'real') são uma realidade. Para cada falso negativo em um termo de tamanho k, dois termos de tamanho k-1 são criados na ontologia gerada para CliXO. Essa realidade gera uma ontologia a qual possui muito mais termos do que a 'real'.

Para se contornar isso, desenvolve-se um método em que um parâmetro definido pelo usuário $0 < \beta \leq 1$ dita como a matriz M deve ser editada para corrigir falsos negativos. Dois cliques C_i e C_j são considerados altamente sobrepostos pelo algoritmo se $\forall a \in C_i \cup C_j$, $\frac{|N(a) \cap (C_i \cup C_j)|}{|C_i \cup C_j| - 1} \geq \beta$.

Modifica-se o algoritmo CliXO, de forma a, antes de um clique maximal $C_i \in C_{cur}$ ser adicionado a C_G, averiguar-se todos os outros $C_j \in C_{cur}$ em busca de alta sobreposição com C_i. Para qualquer C_j altamente sobreposto com C_i, procura-se outro $C_k \in C_cur$ que seja altamente sobreposto com C_j. Para todos os pares de cliques C_i, C_j, $M(a,b) \leftarrow max(w(Ci), w(C_j))$ para todos os $(a,b) \in (C_i \cup C_j)$. Adiciona-se então as conexões ajustadas em U e atualiza-se C_{cur}.

2.3.4.4 Alinhamento de Ontologias

Assim como na NeXO a ontologia precisa ser alinhada com a GO. Isso é feito pela mesma metodologia(subseção 2.2.3).

3 RESULTADOS E DISCUSSÃO

No artigo em que a CliXO foi proposta, os autores, utilizando similaridade semântica de Resnik (17) extraída da GO de processo biológico (GO BP), geraram ontologias a partir de diferentes métodos, e averiguaram o quanto elas foram capazes de reconstruir da GO. A CliXO reconstruiu $>$ 98% dos termos da GO BP identicamente e 100% dos termos construídos foram alinhados. Já o algoritmo de Local Fitness alinhou-se a 5% dos termos e obteve 30% de alinhamento nos termos criados. Os métodos hierárquicos testados (single linkage, complete linkage, ward, UPGMA e WPGMA) reconstroem 35-80% dos termos da GO (20% de reconstrução de termos identicos), entretanto os mesmos são forçados, por construção, a criar uma árvore binária, danificando sua precisão (<35%). Por fim, o método NeXO foi utilizado para reconstrução da GO BP com o emprego de *thresholds*, o mesmo reconstruiu 40% da Ontologia e obteve precisão de 70% nos termos gerados. (13)

No intuito de avalia-los no que tange a reconstrução da GO a partir de dados, os autores submeteram diferentes métodos aos mesmos bancos de dados de levedura (interações gênicas, perfil de expressões gênicas e YeastNet v3). Através dos resultados, os autores concluíram que os métodos NeXO e CliXO possuem uma clara vantagem sobre os outros, apresentando resultados similares quando com parâmetros otimizados. Todavia, o método NeXO demonstrou ser muito mais sensível a parâmetros, com pequenas variações da *threshold* provocando grandes perdas, enquanto o método CliXO apresentou grande robustês em grandes intervalos de seus parâmetros.

Com isso, a partir desse ponto o CliXO parece ter se tornado uma espécie de método "padrão ouro"na literatura do assunto, com todos os outros métodos propostos e artigos que se utilizam de alguma hierarquia criada por dados empregando-o, mesmo assim, devido a brevidade de ambos os métodos não existe, até onde o autor pode averiguar, uma conclusão definitiva da academia sobre a CliXO ser de fato o melhor método e mais pesquisas podem ser necessárias para elucidar a questão.

4 CONCLUSÃO

Ontologias são extremamente importantes para o desenvolvimento das ciências e em especial da biologia. No tangente, em específico, à biologia, desde a sua criação a GO vem tomando papel central no que diz respeito a fonte anotações de genes e planejamento de experimentos: ela representa uma sistematização do conhecimento que se possui sobre o papel de cada um dos produtos gênicos nas mais diversas espécies e, portanto, é fundamental para a integração da biologia.

Nesse contexto, o surgimento de métodos de inferência de ontologia através de dados experimentais vem para solucionar alguns problemas da GO, como subjetividade humana na sistematização do conhecimento e falta de rigor nas anotações (18). Entretanto, para além desse escopo inicial, esses métodos podem levar a muito mais.

Com de uma maneira de extrair as relações hierarquicas intrínsecas dos dados, pode-se estar diante de uma revolução na forma de se fazer biologia, saltando do uso de ontologias para a análise de dados para a utilização de dados na criação e avaliação de ontologias.

De fato, já foi sugerido na literatura uma maneira sistemática de integrar os dados disponíveis sobre determinado sistema, de interações proteína-proteína, genéticas e de coexpressão, para gerar conhecimento sobre a estrutura hierárquica do sistema, com a possibilidade de depois encontrar o melhor experimento o possível para iterar sobre os dados e progredir ainda mais no entendimento do sistema em questão. Tal abordagem foi testada para o sistema de autofagia celular como prova de conceito e diversos novos papéis de produtos gênicos foram confirmados e integrados na GO. (19)

Além disso, diversas outras iniciativas promissoras, baseadas ou inspiradas nesses métodos, estão aparecendo na literatura, como um método semi-supervisionado proposto para integrar sistematicamente a GO com dados experimentais(18) e um método que se utiliza de aprendizado profundo com uma rede neural visível para modelar crescimento celular a partir de dados da GO e da CliXO, servindo de base para a análise dos mecanismos moleculares por trás das relações genótipo-fenótipo *in silico*. (8)

É digno de nota que os métodos aqui trabalhados a princípio não servem apenas para extrair ontologias biológicas a partir de dados, são métodos fundamentais, os quais podem ser utilizados nas mais diversas áreas do conhecimento.

Em suma, os métodos de busca por estruturas hierárquicas em redes de interação biológicas são um método auxiliar de grande utilidade na construção da GO. Entretanto, para além disso, eles podem acabar por revolucionar o jeito com que dados e experimentos são enxergados dentro da biologia, sistematizando e acelerando o processo de elucidação

dos diferentes sistemas celulares.

REFERÊNCIAS

1 CHANDRASEKARAN, B.; JOSEPHSON, J.; BENJAMINS, V. What are ontologies, and why do we need them? **IEEE Intelligent Systems and their Applications**, v. 14, n. 1, p. 20–26, 1999.

2 GUARINO, N. **Formal ontology in information systems: Proceedings of the first international conference (FOIS'98), June 6-8, Trento, Italy**. [*S.l.: s.n.*]: IOS press, 1998. v. 46.

3 ALMEIDA, M. B.; BAX, M. P. Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. **Ciência da informação**, SciELO Brasil, v. 32, n. 3, p. 7–20, 2003.

4 NOY, N. F.; MCGUINNESS, D. L. *et al.* **Ontology development 101: A guide to creating your first ontology**. [*S.l.: s.n.*]: Stanford knowledge systems laboratory technical report KSL-01-05 and ..., 2001.

5 ASHBURNER, M. *et al.* Gene ontology: tool for the unification of biology. **Nature genetics**, Nature Publishing Group, v. 25, n. 1, p. 25–29, 2000.

6 CONSORTIUM, G. O. **Gene Ontology Website**. Disponível em: http://geneontology.org/.

7 DOLINSKI, K.; BOTSTEIN, D. Automating the construction of gene ontologies. **Nature biotechnology**, Nature Publishing Group, v. 31, n. 1, p. 34–35, 2013.

8 MA, J. *et al.* Using deep learning to model the hierarchical structure and function of a cell. **Nature methods**, Nature Publishing Group, v. 15, n. 4, p. 290, 2018.

9 LEONELLI, S. *et al.* How the gene ontology evolves. **BMC bioinformatics**, Springer, v. 12, n. 1, p. 1–7, 2011.

10 DUTKOWSKI, J. *et al.* A gene ontology inferred from molecular networks. **Nature biotechnology**, Nature Publishing Group, v. 31, n. 1, p. 38–45, 2013.

11 CLAUSET, A.; MOORE, C.; NEWMAN, M. E. Hierarchical structure and the prediction of missing links in networks. **Nature**, Nature Publishing Group, v. 453, n. 7191, p. 98–101, 2008.

12 PARK, Y.; BADER, J. S. Resolving the structure of interactomes with hierarchical agglomerative clustering. **BMC bioinformatics**, BioMed Central, v. 12, n. 1, p. 1–10, 2011.

13 KRAMER, M. *et al.* Inferring gene ontologies from pairwise similarity data. **Bioinformatics**, Oxford University Press, v. 30, n. 12, p. i34–i42, 2014.

14 COSTA, L. d. F. *et al.* Characterization of complex networks: A survey of measurements. **Advances in physics**, Taylor & Francis, v. 56, n. 1, p. 167–242, 2007.

15 MYUNG, I. J. Tutorial on maximum likelihood estimation. **Journal of mathematical Psychology**, Elsevier, v. 47, n. 1, p. 90–100, 2003.

16 JEAN-MARY, Y. R.; SHIRONOSHITA, E. P.; KABUKA, M. R. Ontology matching with semantic verification. **Journal of Web Semantics**, Elsevier, v. 7, n. 3, p. 235–251, 2009.

17 RESNIK, P. Using information content to evaluate semantic similarity in a taxonomy. **arXiv preprint cmp-lg/9511007**, 1995.

18 LI, L.; YIP, K. Y. Integrating information in biological ontologies and molecular networks to infer novel terms. **Scientific reports**, Nature Publishing Group, v. 6, n. 1, p. 1–10, 2016.

19 KRAMER, M. H. *et al.* Active interaction mapping reveals the hierarchical organization of autophagy. **Molecular cell**, Elsevier, v. 65, n. 4, p. 761–774, 2017.