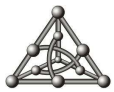
ANÁLISE DE NUTRIENTES UTILIZANDO REDES METABÓLICAS

Paulo Vieira Milreu

Dissertação de Mestrado apresentada ao
Departamento de Computação e Estatística do
Centro de Ciências Exatas e Tecnologia da
Universidade Federal de Mato Grosso do Sul



Orientador: Fábio Henrique Viduani Martinez

O autor teve apoio financeiro do CNPq durante o desenvolvimento deste trabalho.

Setembro de 2008

ANÁLISE DE NUTRIENTES UTILIZANDO REDES METABÓLICAS

Este exemplar corresponde à redação
final da dissertação devidamente corrigida
e defendida por Paulo Vieira Milreu
e aprovada pela comissão julgadora.

Campo Grande/MS, 12 de setembro de 2008.

Banca Examinadora:

- Prof. Dr. Said Sadique Adi (DCT-UFMS)
- Profa. Dra. Maria Emília Machado Telles Walter (CIC-UNB)
- Prof. Dr. Fábio Henrique Viduani Martinez (orientador) (DCT-UFMS)

Agradecimentos

Este trabalho de mestrado só teve início graças ao apoio e incentivo recebidos da minha companheira, Adriana. Portanto, nada mais justo do que agora, ao seu final, agradecê-la tanto pelo incentivo inicial quanto pela sustentação desde então, pelas cobranças, por me ajudar a manter o foco e a sempre acreditar que sim, seria possível. Muito obrigado, Dri!

Agradeço também ao meu outro ponto de sustentação neste processo, o meu querido amigo e sábio orientador Fábio. Obrigado pelo trabalho em conjunto, pela paciência, apoio, orientação e carinho demonstrados durante estes quase 3 anos... "mega-inverno"incluído.

Finalmente, aproveito para agradecer ao Ludovic, à Marie-France, ao Joe, à minha mãe, minha família e a todos que me ajudaram durante esses últimos dois anos de trabalho! Muito obrigado.

E espero o mesmo apoio de todos no doutorado...

Resumo

Neste trabalho apresentamos uma introdução sobre as redes metabólicas de organismos, seus principais conceitos biológicos, as possíveis formas de modelagem destas redes em estruturas de dados, uma formalização matemática de três dos principais problemas relacionados à análise de nutrientes e um método baseado apenas na topologia da rede para realizar análise de nutrientes através da enumeração de conjuntos minimais de precursores ausentes, que são suficientes para se produzir compostos essenciais definidos como alvo.

Palavras-chave: metabolismo, rede metabólica, análise de nutrientes.

Abstract

In this work we present an introduction to metabolic networks of organisms, its main biological concepts, different data structures that can be used to model these networks and a formal definition of three of the main problems related to nutrient analysis. Finally, this work presents a method based on the topology of the network to enumerate all the precursor sets of nutrients that are enough to synthetize some compounds defined as the targets of the network.

Keywords: metabolism, metabolic network, nutrient analysis.

Sumário

Capítulo 1

Introdução

O foco deste trabalho é o estudo do metabolismo celular e das reações químicas que definem estes metabolismos. Um dos tópicos de estudo relacionados às redes metabólicas é a análise de nutrientes. Algumas das motivações que movem este tipo de análise envolvem desde determinar um meio de crescimento de tamanho mínimo para um organismo ou uma célula, isto é, um conjunto mínimo de elementos químicos que propiciam a vida ou o crescimento de um organismo, até o estudo da relação de hospedeiro e parasita através da análise das capacidades metabólicas de um e de outro e da interdependência existente entre eles, causada quando compostos químicos essenciais são sintetizados apenas por um deles.

O interesse do presente trabalho é voltado para a análise de nutrientes através do rastreio de precursores, isto é, a partir de uma descrição dos compostos químicos produzidos pelas células e das reações químicas que produzem esses compostos, analisar qual os conjuntos de elementos químicos que devem estar disponíveis para que uma célula seja capaz de crescer e se reproduzir. Esse conjunto de reações químicas acrescidos de informações sobre as enzimas que catalisam essas reações constitui uma rede metabólica ou as vias metabólicas de um organismo. A rede mundial de computadores disponibiliza informações de redes metabólicas de vários organismos, dentre eles o *Escherichia coli*, que possui grande parte de suas vias metabólicas disponíveis para acesso na página da enciclopédia da *E. coli* [8].

A análise de nutrientes através do rastreio de precursores é apenas um dentre vários problemas associados ao estudo de redes metabólicas, sendo que uma variação desse problema consiste em encontrar conjuntos minimais de reações – e não de compostos químicos – necessárias para que o organismo cresça e se reproduza [4]. Outro problema destacado é a comparação de redes metabólicas para gerar, por exemplo, a filogenia entre organismos a partir das semelhanças entre suas redes metabólicas ou analisar semelhanças funcionais entre enzimas ou reações de dois organismos [31] [12] . Uma das técnicas de comparação é o alinhamento de redes metabólicas, analogamente ao que é feito com alinhamento de seqüências de DNA, existindo tanto técnicas de alinhamento de pares de redes metabólicas [1] [26] quanto técnicas que se destinam ao alinhamento de múltiplas redes metabólicas [33] [11]. Um problema também relacionado à análise de nutrientes é a análise de viabilidade de linhagens mutantes, que consiste em introduzir mudanças nas reações químicas de uma rede metabólica de um organismo e, através de simulações, analisar se essa variação implicaria na morte dessa hipotética linhagem modificada [35].

Um algoritmo de análise de nutrientes por rastreio de precursores deve ser capaz de processar a estrutura de uma rede metabólica e, dada uma lista de compostos definidos como essenciais, isto

é, que devem ser sintetizados por esta rede, avaliar ou gerar um conjunto mínimo de compostos que seja capaz de sintetizar esta lista. Esse conjunto mínimo de compostos químicos a partir do qual a célula é capaz de obter os seus nutrientes essenciais é chamado de meio de crescimento de tamanho mínimo e consiste na identificação do menor conjunto de compostos capaz de fazer com que a rede metabólica produza os compostos essenciais. Outro problema relacionado é o de identificar todos os meios de crescimento minimais – e não apenas o meio de crescimento mínimo. Um meio de crescimento é minimal quando a remoção de qualquer um dos seus elementos faz com que nem todos os compostos essenciais sejam produzidos. Nesse trabalho, estamos interessados em realizar a análise de nutrientes para identificar todos os conjuntos minimais de precursores – chamados compostos ausentes – na produção destes compostos identificados como essenciais. O termo compostos ausentes é usado no sentido de que caso os compostos ausentes identificados pelo método de análise de nutrientes fossem acrescidos ao conjunto inicial de compostos disponíveis, os compostos essenciais seriam todos produzidos.

Romero e Karp estão entre os pioneiros nas pesquisas em análises de nutrientes. Em [27], eles apresentam um estudo de análise de nutrientes em duas etapas visando detectar inconsistências nas informações metabólicas presentes na base de dados do ECOCYC. Na primeira das etapas do método de Romero e Karp, chamada de *forward propagation*, computam-se todos os compostos que uma rede metabólica pode produzir se alguns outros compostos químicos estiverem disponíveis ao organismo. O objetivo é descobrir se a rede metabólica do organismo pode sintetizar determinados compostos essenciais, como aminoácidos ou paredes celulares, a partir dos compostos indicados como disponíveis. A segunda etapa do método é necessária apenas quando pelo menos um dos compostos destacados como essenciais – ou alvos – não é produzido pela rede metabólica em sua fase de *forward propagation* e consiste em identificar quais compostos devem ser adicionados à lista dos compostos disponíveis, para que todos os compostos alvos sejam sintetizados.

Em um trabalho mais recente, Handorf, Ebenhöh e Heinrich [15] definem o conceito de escopo de um conjunto de compostos químicos, chamados sementes, como sendo o conjunto de compostos químicos gerados por uma rede metabólica a partir dos compostos pertencentes ao conjunto de sementes. O algoritmo proposto para se computar o escopo de um conjunto de sementes tem uma fase de expansão, na qual as reações que dependam apenas das sementes são disparadas e os seus produtos são adicionados ao conjunto inicial de sementes, sendo que essa expansão se dá até que o conjunto de sementes não possa ser aumentado. Este conjunto de compostos resultante no final do processo de expansão é chamado de escopo do conjunto inicial de sementes. Este método pode ser encarado como uma adaptação do algoritmo de *forward propagation* proposto por Romero e Karp em [27]. Os principais resultados em termos de análise de nutrientes obtidos em [15] são de análise estrutural da rede metabólica, através da comparação dos escopos produzidos por diferentes combinações de sementes.

Em [14], Handorf *et al.* concentram-se no problema de encontrar um conjunto minimal de precursores que sintetize um determinado composto alvo. A proposta do trabalho apresentado em [14] é encontrar um conjunto de compostos sementes que seja minimal e cujo escopo contenha um determinado composto escolhido como alvo.

De certa forma, o nosso trabalho [6] é uma extensão dos métodos de Romero e Karp e também de Handorf *et al.*, porém, em nossa abordagem de análise de nutrientes procuramos tratar o problema das dependências cíclicas na sintetização de compostos, problema este que embora relativamente freqüente nas redes metabólicas dos organismos, não foi claramente abordada em [27] e não foi tratada em [15] ou [14].

Além dessa contribuição, procuramos atacar outros problemas como a enumeração de todos os conjuntos minimais de precursores necessários à sintetização de um ou mais compostos alvos, sendo que a solução deste problema considerando-se mais de um composto alvo simultaneamente é uma outra contribuição do presente trabalho. Alguns dos resultados obtidos até aqui com a aplicação de nosso método de análise de nutrientes [6] envolvem a identificação de falhas em informações metabólicas de bases de dados como o ECOCYC, assim como haviam feito Romero e Karp, e também uma análise mais abrangente da relação entre parasita e hospedeiro, a partir da análise dos nutrientes compartilhados entre eles [6]. Finalmente, outra contribuição é a formalização das definições dos principais problemas abordados, além de apresentar análise da complexidade dos problemas estudados e análise dos algoritmos apresentados.

A seqüência deste trabalho está organizada da seguinte maneira. O capítulo 2 apresenta as principais definições biológicas referentes ao metabolismo celular, além de apresentar algumas das principais bases de dados metabólicas disponíveis para acesso através da *internet*. O capítulo 3 apresenta as diferentes formas de se modelar uma rede metabólica em estruturas de dados computacionais, incluindo a modelagem através de grafos bipartidos de compostos e reações, estrutura adotada no restante do trabalho. O capítulo 4 aprofunda o estudo sobre análise de nutrientes utilizando-se conjuntos de precursores ausentes, trazendo uma definição formal dos problemas, discussão de suas complexidades e a apresentação de um algoritmo para enumerar todos os conjuntos minimais de precursores de um metabólito alvo, para uma rede metabólica. O capítulo 5 apresenta resultados biológicos obtidos com o método de análise de nutrientes proposto neste trabalho e, finalmente, o capítulo 6 conclui o trabalho, demonstrando os principais benefícios trazidos pelo método proposto e apontando pontos de melhoria e de discussão futuras.

O trabalho contém ainda o apêndice A, contendo uma especificação técnica sobre a ferramenta de análise de nutrientes construída, destacando o diagrama de classes com o detalhamento de cada uma delas, além de um breve tutorial sobre a forma de utilização da aplicação. O apêndice B traz um glossário com os principais termos biológicos utilizados no trabalho.

Capítulo 2

Biologia das Redes e Vias Metabólicas

Este capítulo apresenta na seção 2.1 os principais conceitos e termos relativos à biologia das redes metabólicas e na seção 2.2 alguns detalhes adicionais sobre o processo de reconstrução de uma rede metabólica para um novo organismo, que é o processo através do qual se remonta a rede metabólica de um novo organismo através de seus dados genômicos e da rede metabólica de um outro organismo de referência. Muitas dessas informações estão amplamente disponíveis para acesso através da rede mundial de computadores. Alguns dos principais bancos de dados de informações sobre redes e vias metabólicas são apresentados na seção 2.3.

2.1 Aspectos Biológicos

O conteúdo deste capítulo foi elaborado através de conceitos expostos em [13], [20] e, principalmente, em [21] e [28].

O metabolismo celular é, fundamentalmente, a matéria biológica relacionada a este trabalho. Assim, este capítulo tem por objetivo apresentar os principais conceitos biológicos relacionados ao estudo empreendido, além de estabelecer um vocabulário de termos fundamentais.

Por **metabolismo** compreende-se todo o conjunto de transformações que as substâncias químicas sofrem no interior das células dos organismos. Essas transformações são chamadas de **reações químicas**. O metabolismo pode ser visto como um conjunto de reações químicas intracelulares. Uma **reação química** é a transformação de um conjunto de substâncias químicas, chamadas de **reagentes** ou **substratos**, gerando um conjunto de outras substâncias químicas, chamadas de **produtos da reação**. Chamam-se **metabólitos** as substâncias químicas participantes de reações químicas que compõem o metabolismo de algum organismo, podendo ser tanto reagentes quanto produtos de reações.

Existem dois tipos possíveis de metabolismo: o anabolismo e o catabolismo. O **anabolismo** compreende a síntese ou formação de compostos através do uso de energia e do consumo de reagentes, enquanto o **catabolismo** é a degradação, ou "quebra" de compostos, liberando energia e produzindo compostos que servem de reagentes para outras reações. Tanto um quanto outro se dão através de reações químicas.

A velocidade em que as reações ocorrem nas células é um fator determinante para a sustentação da vida dessas células. Entretanto, as maneiras usuais pelas quais os pesquisadores estimulam, ou catalisam, as taxas de reações, tais como a elevação da temperatura, o aumento da pressão, o aumento da concentração de um dos reagentes ou a adição de ácidos e bases, não ocorrem naturalmente em nível celular. As **enzimas** são os catalisadores das reações químicas das células, aumentando as taxas de reações em várias ordens de magnitude. Geralmente, uma enzima catalisa um pequeno número de reações químicas e para que estas reações ocorram é necessária a participação desta enzima específica.

Desta forma, as enzimas desempenham o papel de incentivar a conversão dos substratos em produtos. Esse processo ocorre da seguinte forma: inicialmente ocorre a associação da enzima E com os substratos S, produzindo um complexo intermediário enzima-substrato ES: E+S $\rightleftharpoons$ ES. A seguir, esse complexo ES passa por um estágio de transição chamado de **catálise**, que é denotado por EA, transformando-se em um complexo enzima-produto EP: ES $\rightleftharpoons$ EA $\rightleftharpoons$ EP. Finalmente, ocorre a quebra do complexo enzima-produto EP, liberando as enzimas e os produtos resultantes da reação: EP $\rightleftharpoons$ E + P, conforme exemplificado a seguir:

$$\mathrm{E} + \mathrm{S} \rightleftharpoons \mathrm{ES} \rightleftharpoons \mathrm{EA} \rightleftharpoons \mathrm{EP} \rightleftharpoons \mathrm{E} + \mathrm{P}.$$

Algumas pequenas moléculas especiais, chamadas **co-fatores**, são essenciais para a ação das enzimas. Elas se encaixam nas enzimas melhorando ou piorando a atividade da enzima. No primeiro caso, são chamadas de **ativadores** da enzima enquanto no segundo caso são chamadas de **inibidores**. Para que uma reação química ocorra, então, algumas condições devem ser satisfeitas:

(*i*) todos os reagentes devem estar presentes na célula, em quantidades necessárias para que a reação ocorra;

(*ii*) a enzima que catalisa a reação deve estar presente;

(*iii*) Co-fatores que interferem na atividade da enzima devem ou não podem estar presentes, dependendo de serem ativadores ou inibidores;

Outro fator determinante para a ocorrência das reações químicas é que, para que os substratos e enzimas estejam presentes e em quantidades necessárias à reação, pode ocorrer tanto a síntese desses constituintes pela própria célula quanto a absorção desses compostos. As enzimas são proteínas ou complexos de proteínas sintetizadas pela própria célula enquanto outros compostos químicos necessários às reações podem ser sintetizados ou absorvidos pela célula, através da membrana celular, por meio de reações químicas denominadas **reações de transporte**.

As atividades enzimáticas são classificadas e catalogadas pelo *International Union of Biochemistry and Molecular Biology* (IUBMB). Cada enzima é classificada por um **EC Number**, que é um código hierarquicamente definido através de 4 números, separados por pontos, no formato X.Y.W.Z, cada número refinando progressivamente a ação enzimática realizada. O primeiro número, X, define o tipo de reação catalisada, os dois números seguintes, Y e W, indicam a classe e subclasse da enzima enquanto Z, o último número, é um identificador seqüencial da enzima dentro desta classificação hierárquica. O sítio `http://www.chem.qmul.ac.uk/iubmb/enzyme/rules.html` traz mais informações e detalhes sobre as regras de classificação e nomenclatura de enzimas de acordo com as reações que elas catalisam.

O metabolismo da célula permite que ela mantenha-se viva, cresça e se reproduza. Para que essas fases da vida celular ocorram é necessária a presença contínua de fontes de energia que servirão como substratos iniciais para uma série de reações encadeadas que terão como produto final os constituintes celulares, propiciando tanto a garantia da vida celular quanto o crescimento e a reprodução da célula.

A principal fonte de energia para a vida são os raios solares. Alguns organismos são capazes de extrair energia diretamente dessa fonte, através do processo chamado **fotossíntese**. Durante a fotossíntese, a energia solar é transformada e armazenada em alguma forma de energia química como o trifosfato de adenosina (ATP), a nicotinamida adenina dinucleotídeo (NAD) ou ainda a nicotinamida adenina dinucleotídeo fosfato (NADP). Organismos incapazes de realizar fotossíntese obtêm energia de maneira secundária, alimentando-se de organismos que possuem algum tipo de energia química armazenada.

ATP e NADP são biomoléculas energéticas pois armazenam energia quimicamente e funcionam como fontes de energia para o metabolismo celular, de tal forma que quando reagem com outras moléculas no interior das células, a energia liberada é utilizada em outros processos intracelulares.

Uma **reação exergônica** é tal que a energia dos produtos da reação é menor do que a energia existente nos reagentes, fazendo com que haja a liberação de energia livre. Uma **reação endergônica**, por outro lado, é aquela na qual a energia dos produtos é maior do que a energia dos reagentes. Uma reação exergônica ocorre espontaneamente enquanto as reações endergônicas precisam de energia livre disponível para ocorrerem.

Por exemplo, a reação Glucose + $P_i \rightleftharpoons$ Glucose-6-fosfato é endergônica enquanto a reação ATP $\rightleftharpoons$ ADP + P_i, que representa a quebra do trifosfato de adenosina, é uma reação exergônica. Essas duas reações compartilham um produto intermediário, o fosfato inorgânico denotado por P_i, sendo que a energia necessária para que a primeira reação ocorra pode ser obtida da energia liberada pela segunda reação. Essas duas reações podem ser encadeadas na forma

$$\text{Glucose} + \text{ATP} \rightleftharpoons \text{ADP} + \text{Glucose-6-fosfato}$$

omitindo o produto intermediário dos dois lados da reação. O acoplamento de reações exergônicas e endergônicas por meio de compostos intermediários em comum é central para a troca de energia em seres vivos. A quebra de ATP, exibida acima, é uma reação exergônica que possibilita a ocorrência de várias reações endergônicas nas células.

O metabolismo de um organismo é também chamado de **rede metabólica** ou de **vias metabólicas do organismo**. Essas outras denominações procuram deixar mais evidente o fato de que as reações ocorridas nas células estão inter-relacionadas, visto que os produtos de uma reação geralmente servem como substratos para outras reações. A figura 2.1 traz parte da representação da via metabólica do organismo *E. coli*, disponibilizada na ferramenta web ECOCYC.

Várias vias metabólicas são apresentadas na figura 2.1, sendo que símbolos como círculos, retângulos ou triângulos simbolizam compostos ou enzimas enquanto as linhas são as reações químicas. A ferramenta ECOCYC possibilita a visualização de mais detalhes de cada uma das vias, como exemplificado na figura 2.2, que traz o detalhamento da via metabólica responsável pela síntese do glicogênio.

Os arranjos dessas vias podem variar, podendo ser **lineares** (*pathways*), **cíclicos** ou **em camadas**, além de poderem possuir ramificações. Nas **vias lineares**, cada reação, catalisada

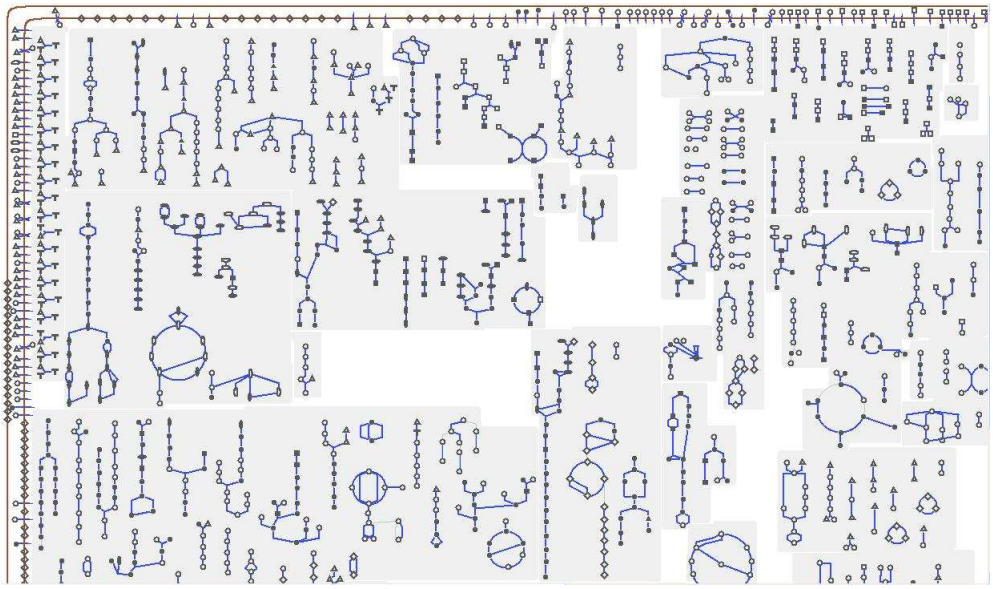


Figura 2.1: Parte da rede metabólica do *E. coli* extraído do ECOCYC.

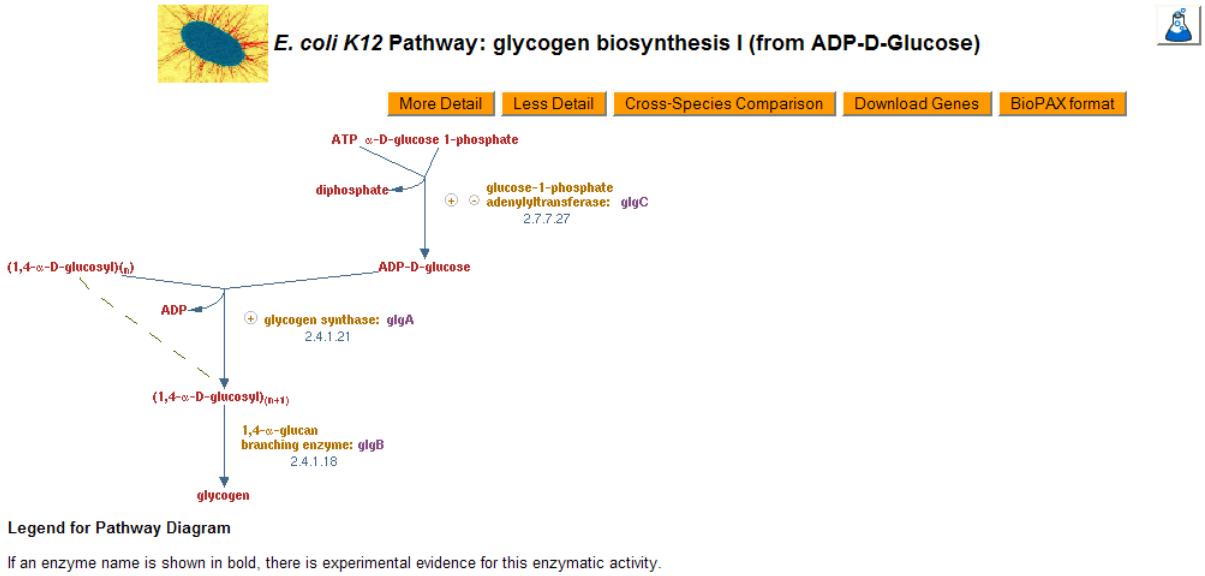


Figura 2.2: Via metabólica de síntese do gligocênio, no *E. coli*, extraído do ECOCYC.

por uma enzima específica, consome substratos e gera produtos que, por sua vez servem como substratos para outra reação. Esse esquema pode ser visualizado na figura 2.3, em que um substrato A sofre uma seqüência de reações até produzir um produto intermediário P, que por sua vez também sofre uma série de reações até gerar o produto final da reação X. Note que a representação das redes metabólicas adotada nas figuras deste capítulo é apenas esquemática, sendo que as discussões acerca da representação e modelagem de redes metabólicas em estruturas de dados computacionais se dará apenas no capítulo 3.

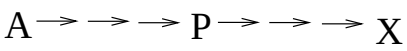


Figura 2.3: Exemplo de via metabólica linear.

Ao conjunto de reações que produzem substratos para outras reações, isto é, que não produzem produtos finais da via metabólica, tais como paredes celulares, organelas, etc, dá-se o nome de **metabolismo intermediário**.

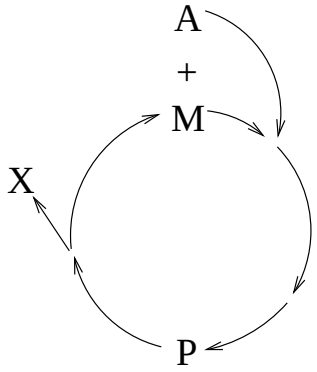


Figura 2.4: Exemplo de via metabólica cíclica.

A figura 2.4 apresenta uma **via metabólica cíclica** e a figura 2.5 demonstra uma **via metabólica em camadas**.

No exemplo da figura 2.4, a reação de dois compostos A e M é iniciada, gerando produtos intermediários P e X e obtendo como produto final o composto M, que desta forma pode acumular-se e iniciar uma nova reação com A, repetindo-se o ciclo. A concentração do composto M desempenha uma função catalítica.

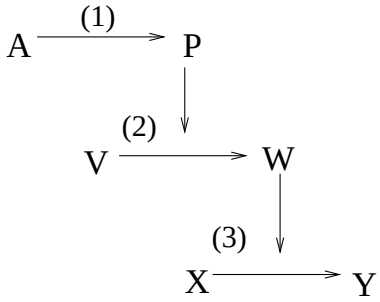


Figura 2.5: Exemplo de via metabólica em camadas.

Em uma via metabólica em camadas, como a apresentada na figura 2.5, a dependência entre as vias se dá sem que o produto das reações de uma via sejam consumidos pelas reações das outras vias, diferentemente do que ocorre nas vias lineares. No exemplo apresentado, a concentração do composto P, produzido pela primeira reação, estimula a segunda reação a ocorrer, fazendo com que seja produzido o composto W que por sua vez desencadeia a produção do composto Y.

Além das diversas configurações topológicas das vias, outra característica importante é a **reversibilidade** das reações, isto é, apesar de ter sido exemplificada uma direção na qual as reações ocorrem, geralmente elas também podem ocorrer no sentido inverso, com os produtos fazendo o papel de reagentes e os reagentes de produtos.

São chamados **nutrientes** os elementos ou compostos químicos essenciais ao metabolismo de um organismo, formando a fonte de sustentação da vida. Um **conjunto de nutrientes** representa os elementos químicos suficientes para a manutenção e reprodução da vida do organismo. O conjunto de nutrientes de um organismo pode diferir do conjunto de nutrientes de outro organismo, já que eles podem possuir metabolismos diferentes.

A partir dos conceitos expostos, pode-se citar o seguinte trecho de [20], que expõe um problema biológico intimamente relacionado ao trabalho ora proposto: "O fato de um organismo ser obrigado a manter equilíbrio energético para sobreviver, mesmo que tenha chegado à maturação, leva-nos à conclusão de que o organismo precisa, ao mesmo tempo, manter um equilíbrio de substâncias. Ele absorve alimentos para obter energia, fazer reparos e manter-se, e excreta os produtos residuais. Portanto o organismo precisa excretar tantos átomos de cada tipo quantos tenha ingerido, ou perderá seu equilíbrio. Tornar-se-á mais gordo ou mais magro, dependendo da direção em que o seu equilíbrio é desregulado. O conceito da existência de um equilíbrio de substâncias leva-nos diretamente a considerar *nutrientes e necessidades nutritivas*. Quais as necessidades mínimas para o equilíbrio alterado de reprodução, crescimento e desenvolvimento?"

2.2 Reconstrução de uma Rede Metabólica

O processo de reconstrução de uma rede metabólica consiste em inferir, a partir de uma rede metabólica de referência e do genoma de um organismo, as reações e vias metabólicas presentes neste organismo. Embora recentemente o processo de comparação genômica tenha sido amplamente automatizado, ainda requer participação manual de especialistas para validar e "lapidar" o trabalho de modelagem de uma nova rede metabólica. O tópico de reconstrução de uma rede metabólica é um tópico de estudo vasto, com diversos desmembramentos e problemas específicos, que fogem ao escopo deste trabalho. A seguir, será apresentada apenas uma introdução ao assunto, com as idéias e problemas fundamentais relacionados ao processo.

Grosso modo, o processo de reconstrução da rede metabólica de um organismo parte do genoma deste organismo, identificando-se a partir daí os genes que sintetizam proteínas e, destas proteínas, quais são enzimas. O passo seguinte é identificar as funções catalíticas de cada enzima sintetizada pelo organismo, sabendo-se assim as reações que devem estar presentes no organismo. Evidentemente, a qualidade deste processo depende da qualidade das anotações do genoma do organismo, dos dados de referência sobre as funções desempenhadas pelas enzimas e também da relação de reações que cada enzima catalisa. Outro fator importante é a distância taxonômica entre o organismo cuja rede está sendo reconstruída e o organismo de referência utilizado.

Vias metabólicas são conjuntos de reações que correspondem a alguma função metabólica. A glicólise por exemplo é uma via metabólica que ocorre na maioria dos organismos. Organismos de referência devem conter as reações comuns a organismos similares mas também reações específicas. O processo envolve então partir deste conjunto de referência e marcar suas reações como presentes no organismo cuja rede está sendo construída, à medida que se constata que uma enzima que catalisa esta reação é sintetizada pelo organismo. Esta técnica, porém, não encontra novas vias metabólicas, não existentes no organismo de referência.

Para realizar a reconstrução de uma rede metabólica já existem várias ferramentas que automatizam grande parte do esforço, localizando genes e identificando as enzimas, cruzando informações em bancos de dados específicos para descobrir as funções destas enzimas, obtendo os seus *EC numbers* e a partir disso reconstruindo a rede metabólica do organismo. Duas das ferramentas mais utilizadas para este propósito são o KAAS – *KEGG Automatic Annotation Server* [17] – e o PATHOLOGIC, a ferramenta de predição que acompanha o pacote BIOCYC PATHWAY TOOLS [25], sendo que a última contém também o módulo PATHWAY HUNTER TOOL, que implementa técnicas que procuram localizar novas vias metabólicas.

O processo automático de reconstrução de redes metabólicas geralmente deixa algumas lacunas a serem preenchidas, tais como reações que não são identificadas pelo processo automático e, por não aparecerem na rede reconstruída, causam "buracos" nas vias metabólicas. A ausência dessas reações pode ser explicada por baixa similaridade da seqüência que corresponde ao gene com as seqüências que codificam a enzima associada a esta reação, no organismo de referência, ou ainda pelo fato de, eventualmente, os produtos da reação terem sido gerados por vias alternativas ou absorvidas do ambiente, dentre outras explicações possíveis. Outro problema ainda pendente ao final do processo automático de reconstrução é a correta indicação quanto à reversibilidade das reações.

2.3 Bancos de Dados de Redes Metabólicas

Nesta seção, serão apresentados alguns dos principais bancos de dados com informações sobre o metabolismo e as vias metabólicas de organismos, disponíveis para acesso público através da rede mundial de computadores.

- KEGG – **Kyoto Encyclopedia of Genes and Genomes** [18].

  O KEGG é um banco de dados contendo informações sobre vias lineares, além de informações específicas sobre compostos, enzimas e reações, dentre outras. As vias metabólicas contidas na base de dados são apresentadas em formato gráfico, com *hiperlinks* para facilitar a navegação entre vias relacionadas. Outra facilidade é a obtenção de dados adicionais sobre as enzimas que catalisam as reações apresentadas. Além da apresentação das vias lineares, o banco oferece recursos para obter esses dados em formato XML;

  Disponível em: `http://www.genome.jp/kegg/kegg2.html`

- ECOCYC – **Encyclopedia of Escherichia coli K-12 Genes and Metabolism** [8]

  Trata-se de um projeto de compilação de informações sobre a bactéria *E. coli*. O ECOCYC é um banco de dados que armazena informações sobre os genes, as reações e o metabolismo do organismo. Esses dados podem ser obtidos em alguns formatos pré-determinados, incluindo nesta lista o FASTA e outros formatos textuais;

  Disponível em: `http://www.ecocyc.org`

- METACYC – **Encyclopedia of Metabolic Pathways** [23]

  Trata-se de um banco de dados contendo mais de 900 redes metabólicas, com garantia de não-redundância das informações e de dados previamente analisados e anotados por especialistas. Além das redes metabólicas, há informações sobre os compostos e as reações;

  Disponível em: `http://www.metacyc.org`

- EMP – **Enzymes and Metabolic Pathways** [9]

  O projeto EMP oferece um banco de dados com informações sobre enzimas e vias metabólicas, permitindo consultas interativas ao banco e obtenção de figuras das vias metabólicas em formato SVG e PNG;

  Disponível em: `http://www.empproject.com`

- TUMOR METABOLOME [34]

  Banco de dados com informações sobre o metabolismo de células com tumores, exemplificando as principais características destas em comparação com as células normais.

  Disponível em: `http://metabolic-database.com`

Devido à descentralização das informações dispostas em várias bases de dados separadas e independentes, existem algumas iniciativas que visam oferecer acesso centralizado aos dados, através da integração dos dados de bases de dados distribuídas, tais como:

- BIOCYC – **Database Collection** [2]

  O BIOCYC é uma coleção de mais de 260 bancos de dados de vias metabólicas, incluindo o METACYC e o ECOCYC;

  Disponível em: `http://biocyc.org`

- BIOSILICO – **An Integrated Metabolic Database System** [3]

  Trata-se de um sistema que permite buscas e análises de dados de vias metabólicas, acessando diversas bases de dados como o ECOCYC e o METACYC, dentre outros.

  Disponível em: `http://biosilico.kaist.ac.kr:8017/biochemdb/index.jsp`

Capítulo 3

Modelagem de Redes Metabólicas

Este capítulo visa apresentar algumas das formas mais comuns de modelagem de uma rede metabólica por meio de estruturas de dados computacionais. A seção 3.1 traz uma visão geral do problema de modelagem de redes metabólicas. As seções 3.2, 3.3, 3.4 e 3.5 apresentam modelagens através de grafos de reações, grafos de compostos, grafos de enzimas e grafos bipartidos, respectivamente. A seção 3.6 apresenta a modelagem de uma rede metabólica através de um hipergrafo. Todas as modelagens apresentadas até a seção 3.6 são qualitativas, não tratando detalhadamente as quantidades de cada reagente e produto das reações da rede metabólica. Uma modelagem quantitativa da rede, através de modos elementares, é apresentada na seção 3.7.

3.1 Representação de Redes Metabólicas

Qual a estrutura de dados mais adequada para representar uma rede metabólica? Levando-se em conta que uma rede metabólica é composta por um conjunto de reações químicas, o problema pode ser inicialmente restrito à melhor estrutura para representar uma reação. Grafos de reações têm sido os principais representantes para atingir esse objetivo, sendo que outras estruturas de dados como grafos de compostos ou grafos bipartidos, com uma partição representando reações e a outra os compostos envolvidos nestas reações, também cumprem esse papel, bem como hipergrafos com hiperarestas representando as reações da rede metabólica.

Apesar da variedade de modelagens, ainda não há uma unanimidade a favor de uma delas e a escolha se dá de acordo com a adequação da modelagem ao problema a ser resolvido. Um dos principais problemas relativos a essas modelagens é a sua falta de precisão, que poderia ser resolvida em uma modelagem através de hipergrafos. As seções seguintes apresentarão com mais detalhes essas variações de modelagem para uma mesma rede metabólica, discutindo com mais detalhes suas diferenças e realçando suas vantagens e desvantagens.

Além da modelagem especificamente da estrutura de uma rede metabólica, alguns trabalhos, como [16], apresentam uma abordagem de modelagem de toda a dinâmica intra-celular do organismo *E. coli* através de modelos de inteligência artificial como os agentes tomadores de decisão, utilizando o modelo Crença, Desejo e Intenção ou BDI, do inglês *Belief, Desire and Intention*. O propósito é caracterizar cada propriedade química da célula como um estado diferente do modelo, medindo as concentrações de substâncias dentro das células que ocasionaram a mudança de estado.

Um dos principais problemas da representação acima descrita é que as simulações por agente consideram um tempo discreto, não capturando completamente os processos dinâmicos, contínuos, do mundo real. Uma abordagem de modelagem híbrida, qualitativa e quantitativa, discreta e contínua, é apresentada e discutida em [22].

Contudo, as características dinâmicas da célula e os problemas de modelagem decorrentes delas não são alvo de estudo deste trabalho e, desta forma, passamos a nos concentrar apenas nas diferentes modelagens estritamente das redes metabólicas dos organismos.

3.2 Grafo de Reações

Um **grafo** é uma estrutura formada por dois tipos de objetos: **vértices** e **arestas**. Cada aresta é um par não-ordenado de vértices, ou seja, um conjunto com exatamente dois vértices. [10]. Assim, formalmente, um grafo G é um par (V, E), sendo V um conjunto finito de vértices e E um conjunto finito de arestas, portanto um subconjunto de V^2. Um grafo é dito ser **orientado** quando as suas arestas são pares ordenados de vértices, neste caso, para um par ordenado de vértices (v_i, v_j), diz-se que a aresta vai de v_i em direção a v_j.

Modelar uma rede metabólica utilizando um grafo é equivalente, portanto, a associar entidades biológicas relacionadas às redes metabólicas aos conjuntos de vértices e arestas de um grafo.

Em um **grafo de reações**, cada reação de uma rede metabólica é representada como um vértice de um grafo orientado, sendo que existirá uma aresta de uma reação R_i para uma reação R_j se há algum composto sintetizado por R_i que seja consumido por R_j.

A principal característica, e vantagem, dessa modelagem é a sua simplicidade e a capacidade de capturar uma relação direta entre reações, permitindo a busca e o reconhecimento de padrões, chamados *motifs*, entre as reações, provavelmente relacionados a vias metabólicas utilizadas pelo organismo. Contudo, esta simplicidade traz consigo imprecisão e ambigüidade, já que, dentre outros problemas, redes metabólicas diferentes como as apresentadas na tabela 3.1, levam ao mesmo grafo de reações, exibido na figura 3.1.

Tabela 3.1: Redes metabólicas distintas que levam ao mesmo grafo de reações.

Rede 1	**Rede 2**
Reação 1: $A \rightarrow B$	$A + B \rightarrow C$
Reação 2: $B \rightarrow C$	$C + B \rightarrow D$
Reação 3: $C \rightarrow D$	$A + D \rightarrow E$

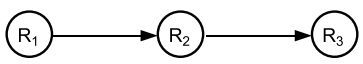


Figura 3.1: —Grafo de reações gerado para as redes da tabela 3.1.

3.3 Grafo de Compostos

Uma outra forma de representação de uma rede metabólica é através de **grafos de compostos**, nos quais associa-se cada composto presente na rede com um vértice de um grafo orientado, sendo que existirá uma aresta de um composto C_i para um composto C_j caso exista uma ou mais reações que tenham C_i como um de seus substratos e C_j como um de seus produtos.

Da mesma forma que ocorre com os grafos de reações, os grafos de compostos são simples mas também imprecisos e ambíguos. A tabela 3.2 apresenta duas redes metabólicas diferentes que levam ao mesmo grafo de compostos, exibido na figura 3.2.

Tabela 3.2: Redes metabólicas distintas que levam ao mesmo grafo de compostos.

Rede 1	**Rede 2**
Reação 1: $A \rightarrow C$	$A + B \rightarrow C$
Reação 2: $B \rightarrow C$	$C \rightarrow D$
Reação 3: $C \rightarrow D$	

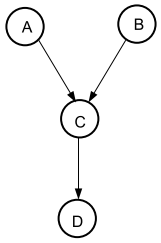


Figura 3.2: —Grafo de compostos gerado para as redes da tabela 3.2.

3.4 Grafo de Enzimas

Nos **grafos de enzimas**, os vértices do grafo são enzimas e uma aresta liga duas enzimas se elas catalisam reações que compartilham pelo menos um composto, seja ele substrato ou produto.

Como reações diferentes podem ser catalisadas pela mesma enzima, elas podem criar atalhos entre compostos distantes na rede metabólica, por exemplo unindo duas vias metabólicas bastante separadas na rede metabólica, caso uma mesma enzima catalise reações distintas, presentes nestas duas vias. Em casos assim, a própria estrutura da rede metabólica não é adequadamente mapeada, mas mesmo assim redes de enzimas podem ser úteis, caso o interesse esteja restrito à relação entre as enzimas.

Uma maneira alternativa de introduzir as enzimas à modelagem de redes metabólicas é incluir rótulos nas arestas dos grafos de compostos e dos grafos de reações, preservando assim a estrutura da rede e proporcionando informação relativa à enzima que catalisa as reações.

3.5 Grafo Bipartido

Nas opções de modelagem apresentadas até aqui, sempre associa-se uma determinada entidade biológica (reações, compostos, enzimas) ao conjunto de vértices de um grafo. Em todas elas, contudo, ocorrem problemas de imprecisão e ambigüidade que fazem com que redes metabólicas distintas sejam modeladas por grafos iguais. A utilização de um **grafo bipartido**, no qual o conjunto de vértices pode conter tanto reações quanto compostos surge, então, para tentar resolver esses problemas.

Na modelagem através de um grafo orientado bipartido, os compostos e as reações são vértices do grafo, e há uma aresta de um composto S para uma reação R se S for substrato de R e há uma aresta de uma reação R para um composto P, caso P seja um produto de R. Desta forma, não há arestas entre dois compostos e nem entre duas reações, formando assim as duas partições do grafo.

As figuras 3.3 e 3.4 trazem as modelagens em grafo bipartido das 4 redes metabólicas apresentadas nas tabelas 3.1 e 3.2. Nas figuras, as elipses representam os compostos e os retângulos representam as reações.

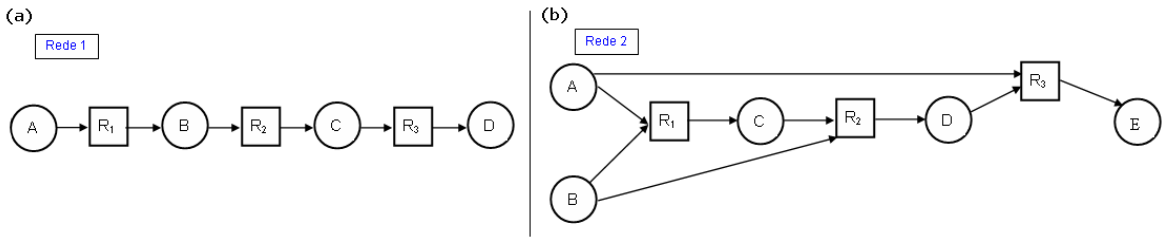


Figura 3.3: —Grafo bipartido gerado para as redes da tabela 3.1.

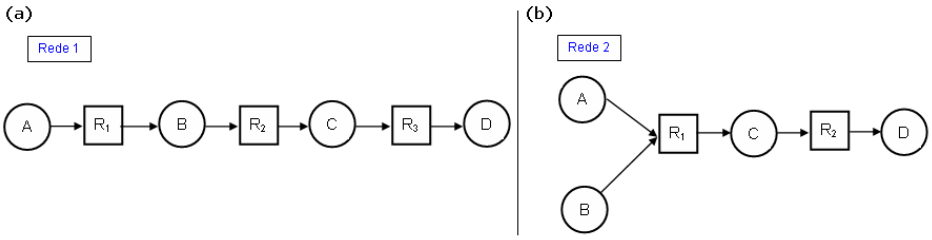


Figura 3.4: —Grafo bipartido gerado para as redes da tabela 3.2.

Ainda que esta modelagem resolva problemas de ambigüidade, ela não contempla alguns aspectos importantes das redes metabólicas, como a estoquiometria – que será devidamente apresentada na seção 3.7 – dos compostos envolvidos nas reações. Entretanto, esse problema pode ser facilmente resolvido incluindo-se uma anotação nas arestas que indique as concentrações de cada substrato e dos produtos resultantes para cada uma das reações.

A modelagem de redes metabólicas através de grafos bipartidos é adotada neste trabalho tanto para os resultados teóricos, apresentados a partir do próximo capítulo, quanto para a contrução da ferramenta e obtenção dos resultados práticos.

3.6 Hipergrafo

Em um hipergrafo também temos um conjunto de vértices, mas passamos a ter um conjunto de **hiperarestas**, sendo que uma hiperaresta é um conjunto de dois ou mais vértices.

A modelagem de redes metabólicas através de **hipergrafos** também resolve os problemas de ambigüidade existentes nos grafos de compostos e de reações, além de contribuir com uma representação mais exata de uma reação como sendo uma hiperaresta, que envolva todos os compostos participantes dela, tanto substratos como produtos. Com isso, a modelagem por hipergrafos tem a vantagem de ser mais precisa que todas as demais anteriormente apresentadas. As figuras 3.5 e 3.6 trazem as modelagens em hipergrafo das 4 redes metabólicas apresentadas nas tabelas 3.1 e 3.2.

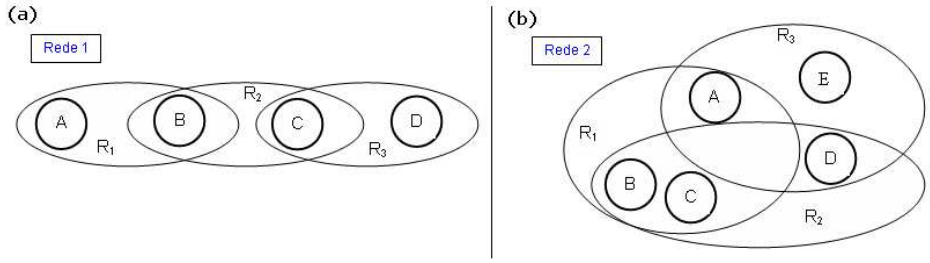


Figura 3.5: —Hipergrafo gerado para as redes da tabela 3.1.

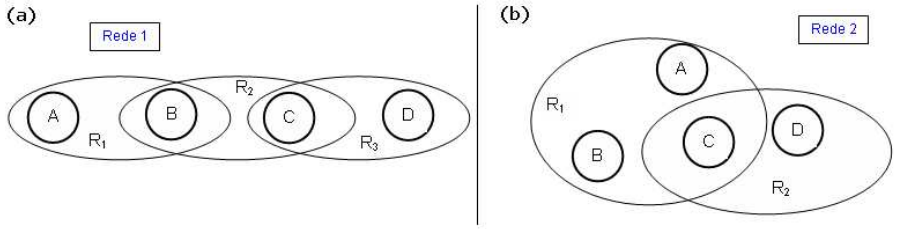


Figura 3.6: —Hipergrafo gerado para as redes da tabela 3.2.

Contudo, para que se obtenha uma correta representação e segmentação entre substratos e produtos de uma reação, é necessário que adote-se o uso de hiperarestas orientadas, no sentido dos substratos para os produtos da reação, tal como ocorria na modelagem através de grafos bipartidos.

Note-se ainda que, apesar de estruturalmente mais apropriada para representar as reações, a estoquiometria não é tão facilmente incorporada a um hipergrafo, usando-se anotações nas hiperarestas, como no caso dos grafos bipartidos. Da mesma forma, a representação em computador de grafos bipartidos e, conseqüentemente, os algoritmos aplicados a grafos bipartidos, tendem a ser menos complexos do que com hipergrafos, justificando-se assim a escolha da modelagem através de grafos bipartidos no restante deste trabalho.

3.7 Modos Elementares

As modelagens de redes metabólicas utilizando-se grafos, descritas nas seções anteriores, não levam em consideração a estoquiometria das reações – mesmo que elas sejam indicadas, eventualmente, como anotações de arestas em um grafo bipartido, por exemplo. Além disso, não há qualquer menção a restrições regulatórias ou termodinâmicas. A **estoquiometria** de uma reação são as quantidades necessárias de cada um dos reagentes desta reação e as quantidades produzidas de cada um dos produtos da reação. A seguir apresentamos os conceitos de modo e modo elementar e de como eles podem ser utilizados para modelar redes metabólicas, atendendo a restrições estoquiométricas, termodinâmicas, entre outras. Uma abordagem de modelagem de rede metabólica é dita ser **baseada em restrições** quando ela visa contemplar todas as restrições quantitativas, termodinâmicas, regulatórias, etc. existentes em nível celular. O conteúdo desta seção foi baseado em [5].

Em uma abordagem baseada em restrições, pode-se continuar modelando a rede metabólica como um grafo ou um hipergrafo, mas deve-se adicionar regras para restringir os fluxos permitidos para a rede, através dos dados estoquiométricos, restrições termodinâmicas ou outras restrições quaisquer. As principais regras servem para garantir o estado de equilíbrio (*steady-state*), para o qual todo composto produzido deve ser consumido, e as restrições termodinâmicas, que ditam que reações irreversíveis só podem ocorrer na direção apropriada.

Aplicando-se essas restrições à uma rede metabólica, apenas fluxos possíveis, isto é, que atendam às restrições definidas, são de interesse. Um fluxo admissível, chamado **modo**, em uma rede metabólica na qual definiram-se restrições é um conjunto de reações que ocorrem determinadas vezes e em determinada seqüência, transformando substratos em produtos, de forma que todos os compostos envolvidos fiquem balanceados – atendam às restrições estoquiométricas – e que as reações irreversíveis ocorram sempre na direção apropriada. Um problema relacionado a essa definição de modo é a análise de fluxo buscando-se um modo que otimize uma função objetivo, como a biomassa ou a produção de ATP.

Para os casos em que não há uma função objetivo a ser otimizada, todos os modos são igualmente interessantes e um problema interessante passa a ser encontrar um conjunto de **modos elementares**, isto é, modos que tem a propriedade de não conterem nenhum outro modo. O conjunto de modos elementares é, desta forma, um conjunto gerador de todos os modos da rede metabólica.

Modos elementares representam uma formalização da definição biológica de via metabólica, já que um modo pode ser visto como um conjunto de enzimas – ou de reações que essas enzimas catalisam – que operam juntamente em estado de equilíbrio para produzir determinados produtos e um modo é elementar se a remoção de uma das enzimas faz com que os produtos não sejam mais produzidos.

Uma **matriz estoquiométrica** S de uma rede metabólica é uma matriz com n linhas e m colunas, n sendo o número de metabólitos e m o número de reações. A posição $S(i, j)$ da matriz S tem valor k se a reação j produz k unidades do metabólito i, neste caso dizemos que i é a saída da reação j. A posição $S(i, j)$ da matriz S tem valor $-k$ se a reação j consome k unidades do metabólito i, neste caso dizemos que i é entrada da reação j. A posição $S(i, j)$ da matriz S tem valor 0 caso o metabólito i não esteja envolvido na reação j. O valor k é o coeficiente estoquiométrico do metabólito i na reação j. Uma matriz estoquiométrica resume a estrutura de uma rede metabólica.

Considere que o conjunto de reações de uma rede metabólica seja dividido em dois: O conjunto $\mathcal{R}$ contendo todas as reações que são reversíveis e o conjunto $\mathcal{I}$ contendo as reações irreversíveis.

Um **vetor de fluxo** – ou distribuição de fluxo – v é um vetor de dimensão m do espaço de reações $\mathcal{R}^m$, no qual o elemento v_i descreve o fluxo na i-ésima reação. Um vetor de fluxo é um modo se:

(i) $Sv = 0$;

(ii) $v_i > 0, \forall i \in \mathcal{I}$.

Chamamos de $R(v)$ o conjunto de reações que participam do fluxo, ou seja, $R(v) = \{j \mid v_j \neq 0\}$. Desta forma, um modo v é dito ser um **modo elementar** caso não haja um outro modo w tal que $R(w) \subset R(v)$. Assim, podemos utilizar álgebra linear para verificar se o vetor v é um modo e depois verificar se o modo é elementar.

A abordagem utilizando-se modos elementares é matematicamente mais precisa e permite uma análise quantitativa da rede metabólica, em detrimento da análise estritamente qualitativa proporcionada pela modelagem através de um grafo bipartido, por exemplo, através da qual obtém-se uma modelagem estrutural sem, contudo, incluir quaisquer informações regulatórias ou regras relacionadas às quantidades dos compostos envolvidos nas reações.

Todavia, é computacionalmente inviável avaliar todos os modos elementares existentes para uma rede metabólica, a fim de obter-se uma enumeração dos conjuntos de nutrientes possíveis para a sintetização de determinados compostos. O que procura-se fazer, devido a essas restrições de desempenho, é realizar uma estratégia mista, em que obtém-se uma série de conjuntos de reações candidatas a serem modos elementares a partir de uma análise qualitativa, fazendo-se em um segundo passo a validação quantitativa destes conjuntos candidatos, avaliando-se então se são modos elementares.

Capítulo 4

Análise de Nutrientes por Conjuntos de Precursores

Uma vez que a rede metabólica de um organismo tenha sido reconstruída, uma das principais questões que podem ser elaboradas é: como são sintetizados os metabólitos essenciais para este organismo? Esta questão pode também ser formulada assim: quais são os metabólitos que o organismo precisa obter do ambiente para produzir tais metabólitos essenciais?

Em [35] é apresentado o conceito de **acessibilidade sintética** que é o número total de reações necessárias para transformar um dado conjunto de substratos e nutrientes em um determinado conjunto de produtos. No trabalho citado é apresentado um estudo do efeito de alterações feitas em S, produzindo uma hipotética rede metabólica mutante. A análise verifica se as alterações podem afetar adversamente o crescimento, indicando assim a viabilidade ou não dessas variações mutantes.

Dá-se o nome de **propagação para frente** (*forward propagation*) [27] ao processo através do qual descobre-se os compostos sintetizados a partir de um conjunto inicial de metabólitos. Pode-se comparar o conjunto de compostos produzidos por uma propagação para frente a um conjunto de metabólitos essenciais, obtendo-se assim a informação se este conjunto inicial é suficiente para sintetizar os metabólitos essenciais. Caso não seja, o problema do rastreio de precursores ausentes (*backtracking*) é definido como a tarefa de descobrir quais são os nutrientes que devem ser adicionados a essa determinada configuração inicial a fim de que determinados compostos sejam gerados. Esses nutrientes ausentes são chamados de **precursores** dos compostos essenciais definidos. Um algoritmo para resolver tanto o problema da propagação para frente quanto o rastreio de precursores ausentes é apresentado em [27]. O algoritmo foi executado sobre a base de dados ECOCYC [8] e proporciona uma análise qualitativa das redes metabólicas baseando-se na conectividade da rede, isto é, procurando responder quais são os compostos gerados pela rede metabólica a partir de um conjunto de compostos fornecidos como entrada ou ainda indicando os nutrientes ausentes que, caso estivessem presentes, fariam com que os produtos da rede fossem de fato produzidos.

Em [4] é apresentada uma análise de nutrientes para o organismo *E. coli*, com uma modelagem de balanceamento de fluxo para a rede metabólica do organismo, tratando diferentes taxas de crescimento esperadas para o *E. coli* como uma problema de otimização de fluxos em redes. O objetivo do estudo é responder à pergunta: qual o conjunto mínimo de genes capaz de permitir a replicação e o crescimento do organismo? O modelo apresentado considera

uma rede metabólica composta de 454 metabólitos e 720 reações. O crescimento é quantificado adicionando-se uma reação adicional que "suga" os compostos da biomassa do *E. coli*, isto é, uma reação que consuma os produtos equivalentes a um outro organismo, de tal forma que a ativação dessa reação é equivalente à produção dos produtos suficientes e necessários ao crescimento e reprodução do organismo. Aos genes são associadas reações, de acordo com a(s) enzima(s) expressada(s) pelo gene, e o problema é modelado como um problema de minimização do número de reações metabólicas ativadas para se atingir um determinado nível pré-fixado de crescimento.

Neste capítulo serão apresentados com mais detalhes os estudos efetuados a partir de [27], relativos aos problemas de análise de nutrientes por conjuntos de precursores. A seção 4.1 aprofunda-se na definição dos conceitos da análise de nutrientes enquanto a seção 4.2 traz a definição formal dos principais problemas relacionados e a seção 4.4 apresenta um algoritmo proposto para resolver um dos problemas apresentados, o de enumeração de conjuntos minimais de nutrientes ausentes, ou precursores.

4.1 Conceitos de Análise de Nutrientes

Alguns estudos têm produzido bancos de dados integrados de genomas e redes metabólicas, através da predição das vias metabólicas de um organismo a partir de seu genoma. Uma dessas iniciativas produziu o banco de dados ECOCYC [8], especializado em informações sobre o organismo *E. coli*. Esse banco de dados contém o genoma completo do organismo, além dos produtos conhecidos dos genes e de um conjunto de todas as reações metabólicas e reações de transporte conhecidas do *E. coli*. O principal objetivo deste e de outros projetos similares, do ponto de vista biológico, é compilar, estruturar e tornar públicas todas as informações conhecidas sobre os organismos estudados, em um ambiente integrado e que permita o estudo, análise e a anotação de novas informações vindas de experimentos e pesquisas relacionadas. Do ponto de vista computacional, a existência desses dados disponíveis de maneira estruturada propiciam uma análise computacional dos dados e, assim, a criação de ferramentas de biologia computacional para gerar novas informações baseadas nestes dados.

No ECOCYC, as informações metabólicas são representadas por estruturas tais como compostos metabólicos, vias lineares e suas reações e as enzimas que catalisam essas reações. As informações genéticas são representadas por estruturas tais como os cromossomos, os genes e os produtos dos genes. A relação primária entre informação metabólica e informação genética se dá entre os produtos dos genes que são enzimas que catalisam reações metabólicas.

A análise empregada em [27] é uma análise metabólica qualitativa, que depende da conectividade da rede metabólica para estimar as necessidades nutricionais de um microorganismo tal qual o *E. coli*. **Microorganismo** é a designação comum a diversos seres pertencentes às categorias de protozoários, bactérias, fungos, algas e vírus, que podem ser vistos apenas com auxílio de microscópios. A abordagem utilizada neste trabalho também pode ser aplicada a outros microorganismos para os quais a informação metabólica seja conhecida.

Pode-se fazer um organismo se desenvolver em um **meio de cultivo** (*culture medium*), a partir do qual ele possa obter nutrientes, isto é, os compostos químicos necessários para crescer e se reproduzir, como por exemplo carbono, fósforo, nitrogênio, potássio, ferro e magnésio, dentre outros. Um **meio de cultivo definido** é tal que todos os compostos químicos que o compõem são bem conhecidos. Quando um meio de cultivo definido contém apenas um conjunto mínimo de nutrientes tais que o microorganismo possa crescer e se desenvolver, ele é chamado de **meio**

de crescimento (de tamanho) mínimo (*minimal growth medium*). Geralmente, os meios de cultivo conhecidos para muitos microorganismos não são meios de cultivo definidos, mas sim meios complexos contendo compostos indefinidos.

Alguns microorganismos possuem vias metabólicas que permitem sintetizar todos os compostos orgânicos a partir de **fontes de carbono** simples, como glucose, lactose ou outros açúcares. Em outros casos, fontes de carbono mais complexas, chamadas de **fatores de crescimento** (*growth factors*), são necessárias para suprir necessidades de compostos orgânicos que os microorganismos não são capazes de sintetizar.

Dessa forma, um **conjunto de nutrientes** para um dado microorganismo consiste em fontes de elementos inorgânicos necessários ao organismo mais uma fonte de carbono e/ou fatores de crescimento tais que o organismo seja capaz de crescer a partir desse conjunto. Um fator de crescimento, quando necessário, geralmente serve também como uma fonte de carbono. Alguns elementos e compostos, chamados de **co-fatores**, são também necessários para que as enzimas do organismo realizem suas funções de catálise.

Em nosso estudo nos concentramos em investigar alguns elementos químicos considerados **compostos essenciais** ou blocos básicos (*building blocks*) usados para sintetizar as macromoléculas essenciais à vida: proteínas, ácidos nucléicos – DNA e RNA – e componentes das membranas celulares. Exemplos de compostos essenciais são os aminoácidos, os lipídios e os sacarídeos, já que é necessário que o microorganismo os produza para garantir seu crescimento e reprodução. Este estudo de compostos que são necessários à sintetização de compostos tidos como essenciais constitui a análise de nutrientes por conjuntos de precursores ausentes, já que o interesse está em identificar os precursores que estão ausentes do meio de cultivo, de forma que a adição deste conjunto de precursores ausentes ao meio de cultivo torne-o um meio de crescimento mínimo.

Romero e Karp [27], com o objetivo de detectar inconsistências no ECOCYC, banco de dados dentro do BIOCYC [2] dedicado à bactéria *E. coli*, usaram uma abordagem que leva em conta a rede completa – e não vias metabólicas isoladas – para encontrar precursores ausentes. Por um processo iterativo chamado *forward propagation*, os autores primeiro computam um conjunto de metabólitos que um conjunto de compostos de entrada são capazes de sintetizar e verificam a presença, neste conjunto resultante, dos metabólitos essenciais, como aminoácidos ou constituintes de paredes celulares. A cada iteração do método de *forward propagation* uma reação é marcada se todos os seus substratos estão disponíveis no conjunto de entrada ou se já haviam sido sintetizados por alguma reação marcada em alguma iteração anterior. O processo pára quando nenhuma nova reação pode ser marcada. Os autores obtêm então uma sub-rede representada pelas reações marcadas e os correspondentes substratos e produtos. O conjunto de metabólitos resultante deste método foi, posteriormente, chamado de **escopo** do conjunto inicial de compostos de entrada [15].

Através de um processo de rastreio dos metabólitos essenciais não contidos no escopo, Romero e Karp geram todos os conjuntos de metabólitos possíveis tais que, caso eles fossem acrescidos aos compostos de entrada para o processo de *forward propagation*, então o escopo deste novo conjunto de entrada conteria todos os compostos essenciais. Os autores definem um metabólito como um precursor potencial caso não seja gerado por nenhuma reação, isto é, caso seja um "ponto de partida" topológico da rede metabólica. O método é, assim, capaz de detectar informação ausente no banco de dados metabólico, mas também pode ser utilizado para obter informação sobre caminhos diferentes para se produzir os compostos indicados como essenciais.

Recentemente, Handorf *et al.* [14] propuseram um método para identificar conjuntos minimais de metabólitos necessários para produzir todos os metabólitos contidos em um conjunto de metabólitos alvos. Cada metabólito que não é um alvo é definido como um precursor potencial e colocado em uma lista. Obviamente, o escopo desta lista contém todos os metabólitos definidos como alvo. O método utiliza então uma abordagem gulosa. A lista de precursores potenciais é ordenada de alguma forma e, começando do topo da lista, cada precursor potencial é sucessivamente removido e o escopo dos metabólitos restantes é computado. Caso não contenha todos os metabólitos alvo, o metabólito removido é devolvido à lista. Quando a lista completa é percorrida, tem-se um conjunto minimal de precursores.

Para se obter todos os conjuntos minimais utilizando-se este método, é necessário testar cada possível permutação da lista de precursores potenciais, o que não é computacionalmente viável. Para se obter uma solução aproximada do espaço completo de soluções, várias permutações aleatórias podem ser geradas. Para reduzir o espaço de soluções, os autores aplicaram algumas heurísticas apoiadas em conhecimentos biológicos, como colocar no início da lista os metabólitos com o maior peso molecular e também colocar ao final da lista metabólitos que sabidamente estão envolvidos em reações de transporte pela membrana do organismo. A posição do metabólito na lista de compostos inicial determina a sua chance de ser mantido no conjunto minimal final devolvido pelo algoritmo, já que os compostos que são avaliados antes têm mais chance de serem removidos. Espera-se que a resposta contenha metabólitos que sejam transportados para dentro da célula, isto é, absorvidos pela célula. Desta forma, compostos quimicamente ricos porém grandes – maior peso molecular – não são bons candidatos, pois dificilmente seriam absorvidos pela membrana celular e devem ser colocados no topo da lista inicial de candidatos a conjunto minimal de precursores, enquanto outros compostos menores e/ou para os quais se sabe que participam de reações de transporte devem ser colocados ao final da lista, aumentando assim a sua chance de fazerem parte do conjunto minimal de precursores.

No método de Romero e Karp [27] brevemente descrito, a definição de precursor potencial é muito restritiva enquanto que no método de Handorf *et al.* [14] é muito ampla.

Neste trabalho propomos um método exato para enumerar conjuntos minimais de precursores que pode lidar com qualquer conjunto de precursores potenciais. Em particular, o usuário da ferramenta produzida a partir de nosso método pode definir seu próprio conjunto precursores. O método também leva em conta o fato de que a maioria das reações são definidas como reversíveis, por conta de falta de informação das concentrações dos metabólitos e das propriedades cinéticas das enzimas. Desta forma, podemos trabalhar tanto com a representação de uma rede metabólica como um grafo bipartido orientado, não-orientado ou misto.

Além disso, o artigo de Romero e Karp [27] não fornece quaisquer detalhes sobre como lidam com ciclos e este é um ponto crucial na análise de redes metabólicas. No método de Handorf [14], o processo usado para computar o escopo é iterativo e as reações podem ser iniciadas somente se todos os seus substratos já estão na sub-rede produzida pelo processo até o momento. O método é incapaz, portanto, de levar em conta metabólitos que não podem ser atingidos por tal processo, tais como os compostos b e d na figura 4.1, apresentada mais adiante na seção 4.2, que são metabólitos típicos de ciclos. Uma outra contribuição deste trabalho é tratar esse problema. Em nosso método apresentamos uma maneira eficiente de lidar com ciclos e a dificuldade que eles apresentam para a computação de conjuntos de precursores.

A enumeração de conjuntos de precursores para um determinado conjunto de metabólitos alvos pode também ser feita primeiro identificando-se todos os modos elementares na rede [30] e depois examinando aqueles que contêm tanto os metabólitos alvos quanto qualquer subconjunto

de precursores potenciais. Modos elementares representam a menor sub-rede que atenda o estado de equilíbrio – a quantidade de cada metabólito produzido dentro do sistema é igual à quantidade consumida. Ao contrário dos métodos de Romero e Karp [27] e de Handorf *et al.* [14], os metabólitos que precisam de sua própria presença na rede para serem produzidos são levados em conta nesta abordagem de manutenção do estado de equilíbrio. A computação de modos elementares requer, contudo, conhecimento da matriz estoquiométrica, ou seja, dos coeficientes estoquiométricos de cada reação da rede. O conjunto de modos elementares encontrado em uma rede depende fortemente desta matriz. Entretanto, com a exceção de algumas poucas redes cujos dados são validados, muitos erros e/ou imprecisões ainda restam nos dados necessários para se gerar as matrizes estoquiométricas dos organismos. A ausência de algumas reações em uma rede, mesmo conhecendo-se a existência destas reações no organismo, um fenômeno freqüente, tem também grande impacto sobre o estado de equilíbrio. Ainda no caso da análise de nutrientes, é interessante levar em conta reações globais ou não tão bem definidas, que se pode obter em estudos metabólicos.

O nosso método não utiliza a matriz estoquiométrica e é livre dos problemas trazidos pelos erros nas anotações estoquiométricas ou por reações que não estejam completamente definidas. Por outro lado, o método apresenta apenas uma análise qualitativa de nutrientes. O conjunto de soluções obtido como resposta por nosso método corresponde a um superconjunto das soluções que são obtidas utilizando-se dados estoquiométricos. Esse superconjunto pode então, *a posteriori*, ser reduzido com uma técnica quantitativa aplicada aos conjuntos minimais obtidos.

4.2 Definição dos Problemas

As definições apresentadas nesta seção estão contidas no artigo [6].

A rede metabólica de um organismo é modelada como um grafo bipartido $G = (V, E)$ com uma bipartição $\{C, R\}$ de V, sendo C um conjunto de metabólitos, também chamados compostos, e R o conjunto de reações da rede. Se todas as reações de uma rede metabólica forem irreversíveis, então o grafo que modela a rede metabólica é um grafo bipartido orientado. Se todas as reações forem reversíveis, a rede poder ser modelada em um grafo bipartido não-orientado. Caso contrário, ela é modelada em um grafo bipartido misto. Um exemplo é dado na figura 4.1. Tanto os grafos não-orientados quanto os grafos mistos requerem uma informação adicional nas arestas - direita ou esquerda, por exemplo - para remover ambigüidades sobre quais metabólitos estão em cada lado da reação.

Grosseiramente, podemos interpretar uma rede metabólica G como um conjunto de metabólitos e um conjunto de reações onde cada reação toma um subconjunto dos metabólitos, chamados **substratos**, e os transforma em outro subconjunto de metabólitos, chamados **produtos** da reação. Neste contexto, uma rede metabólica tem alguns metabólitos disponíveis como reagentes para iniciar algumas reações. Passo a passo, as reações são disparadas, com alguns produtos tomando os papéis de reagentes em novas reações. Dessa forma, um subconjunto de reagentes disponíveis pode gerar um subconjunto de novos produtos alguns passos depois. Uma formalização desse processo é apresentada a seguir, sendo que $\mathcal{P}(\cdot)$ é usado para denotar o conjunto das partes de um conjunto.

Definição 4.1 *Sejam P e A conjuntos de metabólitos em C tais que $P \cap A = \emptyset$. Chamamos P o conjunto de* ***precursores potenciais*** *de G e A o conjunto de* ***metabólitos alvos*** *de G.*

DEFINIÇÃO 4.2 *Seja*

$$\begin{array}{rccc} f\colon & \mathcal{P}(C)\times\mathcal{P}(C\setminus(P\cup A)) & \longrightarrow & \mathcal{P}(C) \\ & (X,X') & \longmapsto & f(X,X')=Y \end{array}$$

uma função tal que, dados subconjuntos $X\in\mathcal{P}(C)$ e $X'\in\mathcal{P}(C\setminus(P\cup A))$, f computa um subconjunto $Y\in\mathcal{P}(C)$ de metabólitos que podem ser alcançados a partir de $X\cup X'$ usando apenas reações que requeiram pelo menos um metabólito de X. Chamamos f de ***função alcance de X e X' em G.***

Desta forma, um composto c pertence a $f(X,X')$ se existir alguma reação r em G que sintetize c tendo como substratos apenas compostos contidos em $X\cup X'$ e, obrigatoriamente, ao menos um substrato contido em X.

DEFINIÇÃO 4.3 *Seja*

$$\begin{array}{rccc} f^i\colon & \mathcal{P}(C)\times\mathcal{P}(C\setminus(P\cup A)) & \longrightarrow & \mathcal{P}(C) \\ & (X,X') & \longmapsto & f^i(X,X') \end{array}$$

uma função recursiva tal que para cada $X\in\mathcal{P}(C)$ e $X'\in\mathcal{P}(C\setminus(P\cup A))$, temos

$$\begin{cases} f^1(X,X')=f(X,X'), \\ f^i(X,X')=f(f^{i-1}(X,X'),X'),\ \textit{para } i>1\,. \end{cases}$$

Chamamos f^i de ***função alcance em i passos de X e X' em G.***

Dizemos que um composto c é alcançável em i passos por um conjunto de compostos inicial X, utilizando-se um conjunto de compostos de apoio X', se c está contido no conjunto resultante do seguinte procedimento, que deve ser repetido i vezes: aplica-se a função alcance no conjunto X e X' e os compostos obtidos como resultado são adicionados ao próprio conjunto X.

DEFINIÇÃO 4.4 *Seja*

$$\begin{array}{rccc} f^*\colon & \mathcal{P}(C)\times\mathcal{P}(C\setminus(P\cup A)) & \longrightarrow & \mathcal{P}(C) \\ & (S,Z) & \longmapsto & f^*(S,Z)=f^k(S,Z) \end{array}$$

uma função tal que $f^k(S,Z)=f^{k+1}(S,Z)$ para algum $k\geqslant 1$, onde $S\subseteq P\subset C$ é um subconjunto do conjunto de precursores potenciais de G e $Z\subseteq C\setminus(P\cup A)$. Chamamos f^* de ***função escopo de S e Z em G.***

Desta forma, a função escopo $f^*(S,Z)$ de um conjunto de metabólitos S com apoio de um conjunto de metabólitos Z representa todos os metabólitos que podem ser sintetizados a partir das reações de uma rede metabólica partindo-se dos metabólitos contidos nos conjuntos S e Z.

Note que a função escopo é monotônica por ser estritamente crescente, visto que se temos conjuntos de compostos $X\supseteq Y$, então $f^*(X,Z)\supseteq f^*(Y,Z)$, para qualquer conjunto $Z\subseteq C\setminus(P\cup A)$. Analogamente, se temos conjuntos de compostos $Z\supseteq Z'$, então $f^*(X,Z)\supseteq f^*(X,Z')$, para qualquer conjunto $X\subseteq P\subset C$.

Para definir que o conjunto S seja um conjunto de precursores de A em G, devemos impor que $f^*(S,Z)$ contenha A e ainda que os compostos contidos em Z sejam auto-regenerados. A idéia de que compostos possam ser auto-regenerados, isto é, participar de vias metabólicas em que são consumidas e geradas, tem respaldo biológico. Em [19] temos uma discussão sobre compostos químicos com essa característica, chamados de replicadores autocatalíticos.

Definição 4.5 *Um conjunto de metabólitos S em uma rede metabólica G é um* ***conjunto de precursores de*** *A* ***em*** *G, com $\emptyset \neq S \subseteq P$ e $S \cap A = \emptyset$, se existir um conjunto $Z \subseteq C \setminus (P \cup A)$ de metabólitos tal que $f^*(S, Z) \supseteq A \cup Z$. Chamamos o conjunto Z de* ***conjunto de metabólitos continuamente disponíveis*** *.*

Definição 4.6 *Um conjunto $S \subseteq P$ é um* ***conjunto minimal de precursores de*** *A* ***em*** *G se S é um conjunto de precursores de A em G e $f^*(S', Z) \not\supseteq A \cup Z$, para cada $S' \subset S$ e algum $Z \subseteq C \setminus (P \cup A)$.*

Definição 4.7 *Um conjunto $S \subseteq P$ é um* ***conjunto mínimo de precursores de*** *A* ***em*** *G se S é um conjunto de precursores de A em G e $|S| \leqslant |S'|$, para cada $S' \subseteq P$ que também seja um conjunto de precursores de A.*

A figura 4.1 ilustra as definições 4.1–4.7.

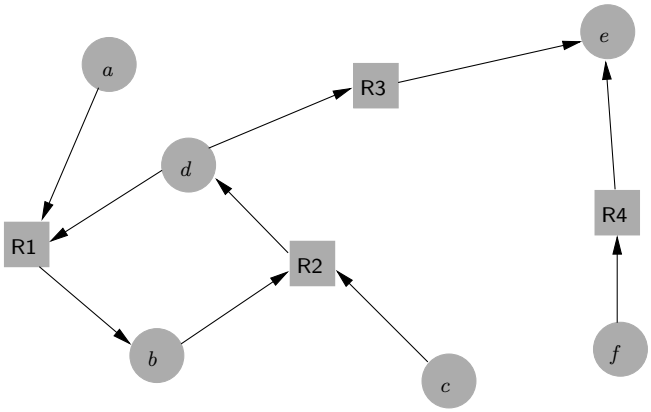


Figura 4.1: Uma rede metabólica G com um conjunto de metabólitos $C = \{a, b, c, d, e, f\}$, conjunto de reações $R = \{\mathsf{R1}, \mathsf{R2}, \mathsf{R3}, \mathsf{R4}\}$, conjunto de precursores potenciais $P = \{a, c, f\}$, conjunto de metabólitos alvos $A = \{e\}$ e conjunto de metabólitos continuamente disponíveis $\{b, d\}$. Veja que, por exemplo, $f(\{a\}, \{d\}) = \{b\}$ e $f(\{f\}, \emptyset) = \{e\}$. Além disso, $f^*(\{a, c\}, \{d\}) = f^3(\{a, c\}, \{d\}) = \{e\} = f^1(\{f\}, \emptyset) = f^*(\{f\}, \emptyset)$. Finalmente, veja que $\{a, c\}$ é um conjunto minimal de precursores de A em G, com metabólitos continuamente disponíveis $Z = \{b\}$ ou $Z = \{d\}$, e $\{f\}$ é um conjunto mínimo de precursores de A em G com $Z = \emptyset$.

Um algoritmo em tempo polinomial que implementa a função escopo f^* é apresentado por Handorf *et al.* em [15], sendo que este algoritmo é uma adaptação do algoritmo de *forward propagation* apresentado por Romero e Karp em [27]. A função f^* devolve o escopo de um conjunto de metabólitos [15], chamados de **sementes**, fornecidos como entrada inicial.

4.3 Formalização dos Problemas e suas Complexidades

As formalizações e análises de complexidade dos problemas apresentados nesta seção estão contidas no artigo [6].

4.3.1 Problemas Analisados

Formalmente, os três problemas de interesse neste trabalho são:

Problema MAL-CP(G, P, A): dados uma rede metabólica $G = (\{C, R\}, E)$, um conjunto $P \subset C$ de precursores potenciais e um conjunto $A \subset C$ de metabólitos alvos, encontrar um conjunto minimal de precursores de A em G.

Problema MIN-CP(G, P, A): dados uma rede metabólica $G = (\{C, R\}, E)$, um conjunto $P \subset C$ de precursores potenciais e um conjunto $A \subset C$ de metabólitos alvos, encontrar um conjunto mínimo de precursores de A em G.

Problema ENUM-MAL-CP(G, P, A): dados uma rede metabólica $G = (\{C, R\}, E)$, um conjunto $P \subset C$ de precursores potenciais e um conjunto $A \subset C$ de metabólitos alvos, encontrar todos os conjuntos minimais de precursores de A em G.

Em MAL-CP deve-se encontrar um conjunto minimal de precursores, em MIN-CP um conjunto mínimo de precursores e em ENUM-MAL-CP enumerar todos os conjuntos de precursores. A seguir apresentamos uma discussão sobre as complexidades dos problemas e na seção 4.4 um algoritmo para resolver ENUM-MAL-CP.

4.3.2 Complexidade do MAL-CP

O problema MAL-CP pode ser resolvido em tempo polinomial, conforme demonstra o algoritmo ingênuo apresentado a seguir.

O algoritmo apresentado toma uma entrada G, P e A do MAL-CPe inicia testando se o maior conjunto de precursores possível – todo o conjunto P – é capaz de sintetizar os compostos do conjunto A, utilizando como candidato a conjunto de metabólitos continuamente disponíveis o conjunto $Z = C \setminus (P \cup A)$. Portanto, o primeiro teste feito na linha 3 valida se há alguma maneira de se sintetizar os compostos alvos. Porém, ainda que este conjunto inicial $S = P$ escolhido seja um candidato à solução, ainda resta averiguar se o conjunto de apoio Z foi auto-regenerado. Este teste é realizado na linha 4. Se Z não tiver sido regenerado, é necessário encontrar um subconjunto Z' de Z que atenda a esta propriedade. Na situação de nenhum subconjunto $Z' \subset Z$ fazer com que o conjunto de precursores potenciais $S = P$ sintetize os compostos contidos em A e também o próprio Z', então não há solução para esta instância do problema e o algoritmo devolve o conjunto vazio como resposta, na linha 9. Para escolher os sub-conjuntos Z', tomamos apenas os compostos que estavam contidos em Z e que foram gerados pelas reações disparadas pelos compostos em S, ou seja, que fazem parte do escopo $f^*(S, Z)$. Podem acontecer três situações cada vez que um novo conjunto Z' é gerado e que um novo escopo é computado, conforme apresentado nas condições da linha 7 do algoritmo. Na primeira condição – $f^*(S, Z) \supseteq A \cup Z$ – o conjunto gerado por $f^*(S, Z)$ contém tanto os compostos alvos quanto o próprio conjunto Z e, assim, o algoritmo deve seguir para as linhas de 10 a 17, nas quais S é transformado em um conjunto minimal. Na segunda condição – $f^*(S, Z) \not\supseteq A$ – o conjunto gerado por $f^*(S, Z)$ já não contém os compostos alvos e, assim, não há conjunto de precursores para este conjunto A e o algoritmo deve devolver o conjunto vazio como resposta. A terceira e última condição ocorre quando $Z = \emptyset$, ou seja, não há mais um novo conjunto $Z' \subset Z$ a ser gerado e o algoritmo

ALGORITMO RESOLVE-MAL-CP(G, P, A)

Entrada: Uma rede metabólica G, um conjunto P de precursores potenciais e um conjunto A de metabólitos alvos;

Saída: Um conjunto minimal de precursores que sintetiza todos os compostos em A.

```
1:  S ← P
2:  Z ← C \ (P ∪ A)
3:  se f*(S, Z) ⊇ A então
4:    se f*(S, Z) ⊉ Z então
5:      repita
6:        Z ← f*(S, Z) ∩ Z
7:      até que f*(S, Z) ⊇ A ∪ Z ou f*(S, Z) ⊉ A ou Z = ∅
8:      se f*(S, Z) ⊉ A ∪ Z então
9:        devolva ∅
10:   Atribua a cor branca a todos os elementos de S
11:   enquanto existir algum elemento de cor branca em S faça
12:     s ← um elemento de cor branca em S
13:     se f*(S − {s}, Z) ⊇ A então
14:       Atribua a cor preta a s
15:     senão
16:       S ← S − {s}
17:   devolva S
18: senão
19:   devolva ∅
```

deve testar, na linha 8, se o conjunto S é capaz de sintetizar os alvos em A sem utilizar nenhum composto continuamente disponível. Caso isso não ocorra, o algoritmo devolve o conjunto vazio como resposta e caso contrário segue para as linhas 10 a 17, a fim de transformar S em um conjunto minimal.

Uma vez definido o conjunto Z de compostos continuamente disponíveis, o algoritmo prossegue tentando eliminar elementos do conjunto S até obter um conjunto minimal. Para tanto, todos os elementos de S são coloridos de branco, na linha 10. Escolhe-se então um elemento qualquer s de S que tenha cor branca e é feito o teste se $f^*(S - \{s\}, Z)$ é um conjunto de precursores de A em G. Se não for, então s não é necessário para a sintetização dos conjuntos em A e é removido de S; caso contrário, s é colorido de preto e mantido em S. O processo é repetido até que não hajam mais elementos brancos em S. O conjunto S resultante deste processo é um conjunto minimal de precursores de A em G.

Como mencionado na seção 4.2, a função f^* pode ser implementada pelo algoritmo *forward propagation* proposto por Romero e Karp [27]. O tempo de execução deste procedimento é $O(V^2)$, onde $V = \{C, R\}$ é o conjunto de vértices de G. No primeiro laço do algoritmo, nas linhas de 5 a 7, a função f^* é chamada $O(Z)$ vezes, no pior caso. No segundo laço do algoritmo, nas linhas de 11 a 16, a função f^* é chamada $O(P)$ vezes, já que na linha 1 o conjunto S é inicializado com o conjunto P. Assim, o tempo de execução do algoritmo RESOLVE-MAL-CP apresentado é $O((P \cdot V^2) + (Z \cdot V^2)) = O(V^3)$.

4.3.3 Complexidade do MIN-CP

Antes de apresentar a complexidade do MIN-CP, apresentamos o algoritmo polinomial CERTIFICA-MIN-CP que valida se um dado conjunto S de tamanho máximo k é um conjunto de precursores de um conjunto de metabólitos alvos A em uma rede metabólica G.

ALGORITMO CERTIFICA-MIN-CP(G, S, A, k)

Entrada: Uma rede metabólica G, um conjunto S de precursores, um conjunto A de metabólitos alvos e um inteiro k;

Saída: **Verdadeiro** caso S seja um conjunto de precursores de A em G e $|S| \leqslant k$. **Falso**, caso contrário.

1: **se** $|S| > k$ **então**
2: **devolva falso**
3: $Z \leftarrow C \setminus S$
4: $Y \leftarrow f^*(S, Z)$
5: **se** $Y \subseteq A \cup Z$ **então**
6: **devolva verdadeiro**
7: **senão**
8: **repita**
9: $Z \leftarrow Z \cap Y$
10: $Y \leftarrow f^*(S, Z)$
11: **até que** $Y \not\supseteq A$ **ou** $Y \supseteq A \cup Z$ **ou** $Z = \emptyset$
12: **se** $Y \supseteq A \cup Z$ **então**
13: **devolva verdadeiro**
14: **senão**
15: **devolva falso**

O algoritmo CERTIFICA-MIN-CP primeiro valida se o conjunto S fornecido como possível solução tem tamanho máximo k e devolve falso caso contrário. Após este passo inicial é computado um conjunto candidato de metabólitos continuamente disponíveis, que é definido como $Z = C \setminus S$, isto é, um conjunto com todos os compostos exceto os candidatos à solução, contidos no conjunto S. O algoritmo prossegue gerando o conjunto $Y := f^*(S, Z)$, isto é, o escopo do conjunto S tendo Z como conjunto de apoio. Se $Y \not\supseteq A$, então S não é um conjunto de precursores de A em G. Caso contrário, existem duas possibilidades. Na primeira, $Y \supseteq A \cup Z$ e o algoritmo termina apontando S como um conjunto de precursores de A em G. Na segunda, $Y \not\supseteq A \cup Z$, isto é, apesar dos compostos contidos em A terem sido gerados, os compostos contidos em Z não o foram, não atendendo assim à restrição de serem auto-regenerados. Como visto na seção 4.2, a função $f^*(S, Z)$ é monotônica e assim para qualquer $Z' \subset Z$, tal que $f^*(S, Z') \supseteq A \cup Z'$ teremos que $Y = f^*(S, Z) \supseteq f^*(S, Z') \supseteq A \cup Z'$ e, assim, se $Y \supseteq A$, como neste caso, então $Z' \subset Y \cap Z$. Desta forma, para encontrar um conjunto Z' é suficiente considerar conjuntos de compostos que estejam contidos em $Z \cap Y$. Intuitivamente, os compostos que devem permanecer em Z são aqueles que foram gerados durante a computação da função escopo, já que apenas estes podem vir a se auto-regenerar, no processo de geração dos compostos alvos.

As linhas de 8 a 11 do algoritmo repetem os passos de gerar novos conjuntos Z – definido como sendo a interseção entre o resultado da computação da função escopo e do próprio conjunto Z no passo anterior – e Y, que é o resultado da função escopo $f^*(S, Z)$. O processo é repetido até que uma das seguintes condições ocorra:

(*i*) $Y \not\supseteq A$;

(*ii*) $Z' = \emptyset$;

(*iii*) $Y \supseteq A \cup Z'$.

Apenas na terceira situação apontamos S como um conjunto de precursores de A em G. Sabendo-se que a função $f^*(S, Z)$ tem tempo quadrático com relação ao número de vértices $|C|$ do grafo que representa a rede metabólica e analisando-se que, no pior caso, serão necessárias $|C|$ iterações, a complexidade do algoritmo de verificação é $O(C^3)$. Com este resultado, podemos então apresentar a seguinte propriedade.

LEMA 4.8 *Dados G, S, A e k, é possível verificar se S é um conjunto de precursores de A em G com tamanho $|S| \leqslant k$ em tempo polinomial.*

PROVA. O algoritmo CERTIFICA-MIN-CP. □

Agora provaremos que a versão de decisão do MIN-CP é um problema NP-completo, através de uma redução do problema HITTING SET, de agora em diante denotado por MIN-HS, para o problema MIN-CP. Informalmente, no problema MIN-HS, temos uma coleção $\mathcal{I}$ de subconjuntos de um conjunto H e queremos encontrar um subconjunto H' de H que intercepte cada conjunto em $\mathcal{I}$. Além disso, queremos que H' tenha no máximo um dado tamanho k. Formalmente,

> **Problema** MIN-HS$(H, \mathcal{I}, k)$: dados um conjunto finito $H = \{1, \ldots, m\}$, uma coleção $\mathcal{I} = \{I_1, \ldots, I_n\}$ de subconjuntos de H, e um inteiro positivo k, encontrar um subconjunto H' de H tal que $|H'| \leqslant k$ e $H' \cap I_j \neq \emptyset$ para cada $I_j \in \mathcal{I}$, $1 \leqslant j \leqslant n$.

TEOREMA 4.9 MIN-CP *é NP-completo.*

PROVA. De acordo com o lema 4.8, MIN-CP está em NP.

Agora mostraremos que MIN-CP é NP-difícil provando que MIN-HS $\leqslant_{\mathrm{P}}$ MIN-CP. Sejam $H, \mathcal{I}$, e k uma instância do MIN-HS tal que $H = \{1, \ldots, m\}$, $\mathcal{I} = \{I_1, \ldots, I_n\}$ com $I_j \subseteq H$ para cada j, $1 \leqslant j \leqslant n$, e k um inteiro positivo. A partir desta instância genérica do MIN-HS, construiremos uma instância particular de MIN-CP. O grafo $G = (V, E)$ é construído como segue. Para cada elemento h em H, adicionamos um vértice h em V. Para cada conjunto I_j em $\mathcal{I}$, adicionamos um vértice I_j em V. Adicionamos também um vértice a em V. Se $h \in I_j$, adicionamos um vértice r_{hj} em V e duas arestas orientadas em E, uma com início em h e fim em r_{hj} e outra com início em r_{hj} e fim em I_j. Adicionamos um vértice r_a em V e uma aresta orientada em E entre I_j e r_a, para cada $j = 1, \ldots, n$. Finalmente, adicionamos uma aresta orientada de r_a para a em E. Perceba que G é um grafo bipartido, com bipartição $V = \{C, R\}$, onde R é o conjunto de vértices r_a e r_{hj}, com $1 \leqslant h \leqslant m$ e $1 \leqslant j \leqslant n$, em V e $C = V \setminus R$. Veja a figura 4.2.

O grafo G pode ser facilmente gerado de uma entrada do MIN-HS em tempo polinomial.

Suponha agora que um conjunto $H' \subset H$ é uma solução para o MIN-HS, com tamanho $|H'| \leqslant k$. Então, cada I_j em $\mathcal{I}$ contém ao menos um elemento de H'. Considerando $S := H'$, temos um conjunto de precursores S tal que $|S| \leqslant k$ e todos os metabólitos I_j são gerados em G e, assim, a reação r_a pode ser disparada e a pode ser sintetizado, já que a reação r_a depende apenas da presença dos compostos I_j que foram sintetizados a partir de S. Desta forma, $S \cup Z$

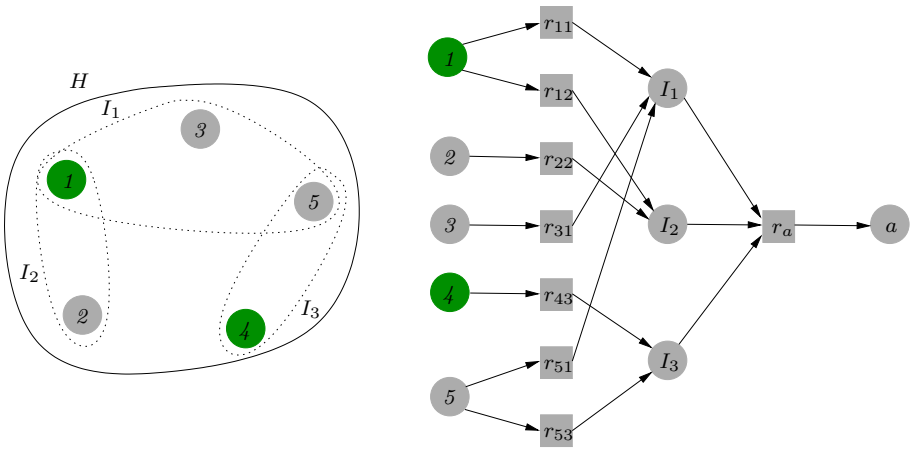


Figura 4.2: Uma transformação em tempo polinomial de uma instância do MIN-HS para uma instância do MIN-CP.

gera a em G, para $Z = \emptyset \subset C \setminus (S \cup \{a\})$. Desta forma, se existe $H' \subset H$ solução de MIN-HS, existe $S = H'$ solução de MAL-CP.

Por outro lado, suponha agora que G tem um conjunto de precursores $S \subset C$ tal que $|S| \leqslant k$ e S gera $a \in V$. Da forma como G é construído, todos os metabólitos I_j, com $1 \leqslant j \leqslant n$, devem ser gerados por S para que a reação r_j seja disparada. Assim, o conjunto de metabólitos S pode disparar todas as reações que geram os metabólitos I_j. Isso significa que $H' := S \subseteq H$ é uma solução para o MIN-HS com $|H'| \leqslant k$. □

4.3.4 Complexidade do ENUM-MAL-CP

O problema de enumerar todos os conjuntos minimais de precursores é exponencial no tamanho da entrada, isto é, no tamanho do conjunto de compostos, tendo em vista que para determinadas instâncias do problema a própria resposta esperada possui tamanho exponencial. O pior caso para o ENUM-MAL-CP ocorre quando a saída deve conter todos os subconjuntos com tamanho $\lfloor p/2 \rfloor$ do conjunto de precursores potenciais P, sendo que $p = |P|$. Neste caso, apenas para mostrar a resposta do problema, devemos gastar $O(p^p)$ passos, por termos exatamente $\binom{p}{\lfloor p/2 \rfloor}$ subconjuntos de tamanho $\lfloor p/2 \rfloor$ de P que são conjuntos minimais de precursores de A em G.

Considere, por exemplo, uma rede metabólica como a apresentada na figura 4.3, que possui um único composto definido como alvo para ser sintetizado. Na rede metabólica da figura, existem n compostos além do composto alvo e $n!$ reações que sintetizam o composto alvo, cada uma delas tomando como substratos um conjunto de compostos diferente, dentro das possibilidades de permutação dos n compostos, por exemplo a reação $R\{1\}$ faz a transformação direta do composto 1 no composto alvo a, enquanto a reação $R\{2, 3\}$ transforma os compostos 2 e 3 no composto alvo a, e assim por diante até a reação $R\{1, 2, ..., n\}$, que tem todos os n compostos como substratos necessários para sintetizar o composto alvo a. Todas essas $n!$ reações representam $n!$ possibilidades de conjuntos de precursores que deverão ser analisadas para se separar os conjuntos não-minimais e produzir a resposta esperada.

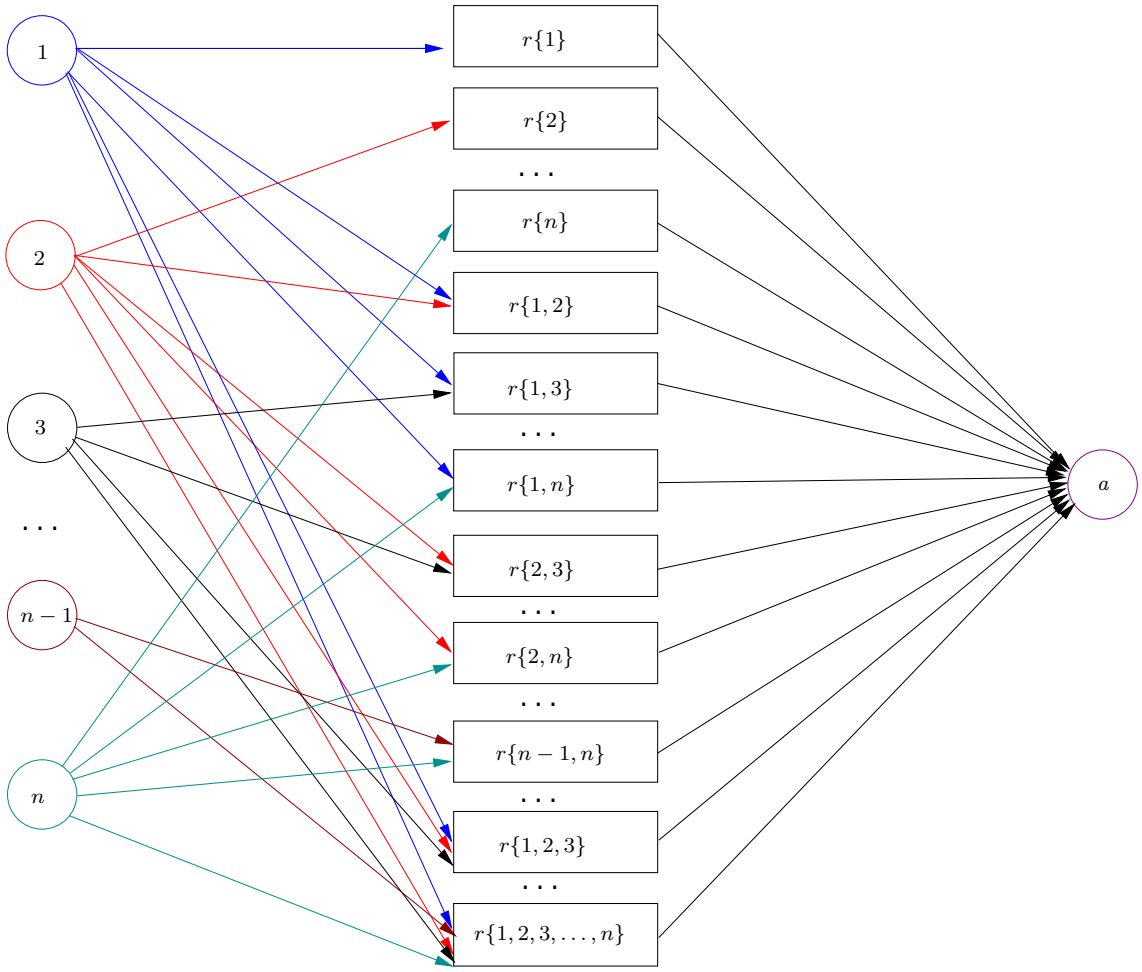


Figura 4.3: Uma rede metabólica com $n!$ reações que sintetizam o alvo.

4.4 Algoritmo para Enumerar Conjuntos Minimais de Precursores

Essa seção apresenta o algoritmo EnumMalCP [6], que foi desenvolvido para enumerar todos os conjuntos minimais de precursores necessários para que uma rede metabólica sintetize um determinado subconjunto de compostos alvo.

Dentre todos os compostos presentes em uma rede metabólica, quais deles são os candidatos a precursores, ou seja, quem pertence ao conjunto de **precursores potenciais**? Como espera-se que os precursores estejam na periferia da rede, definimos os primeiros precursores potenciais como aqueles que não são sintetizados por nenhuma reação ou aqueles que estão envolvidos em uma única reação, que seja reversível (figura 4.4). Precursores desse tipo serão chamados de **precursores potenciais de origem**, representados pelo conjunto S. Porém, no algoritmo EnumMalCP, procuramos não restringir os precursores potenciais apenas ao conjunto de metabólitos de origem, permitindo a definição de compostos internos como precursores potenciais (figura 4.4); chamamos essa classe de precursores potenciais de **precursores potenciais por escolha** e os denotamos pelo conjunto I. Com a intenção de oferecer maior flexibilidade e a possibilidade de eliminar respostas indesejadas - por óbvias, por exemplo, o método contempla

a situação de proibir metabólitos de serem considerados precursores e para isto definimos uma nova classe de metabólitos chamados de **precursores proibidos**, representados pelo conjunto F (figura 4.4). Tanto os precursores potenciais por escolha quanto os precursores proibidos serão entradas para o algoritmo EnumMalCP, enquanto os precursores potenciais de origem podem ser computados automaticamente. Para evitar respostas triviais, desconsideramos como precursores potenciais os compostos contidos no conjunto de metabólitos alvos A, sendo equivalente a considerarmos que eles já fazem parte do conjunto de precursores proibidos F. Desta forma, o conjunto de **precursores potenciais** P pode ser definido como: $P = \big((S \cup I) \setminus F\big) \setminus A$.

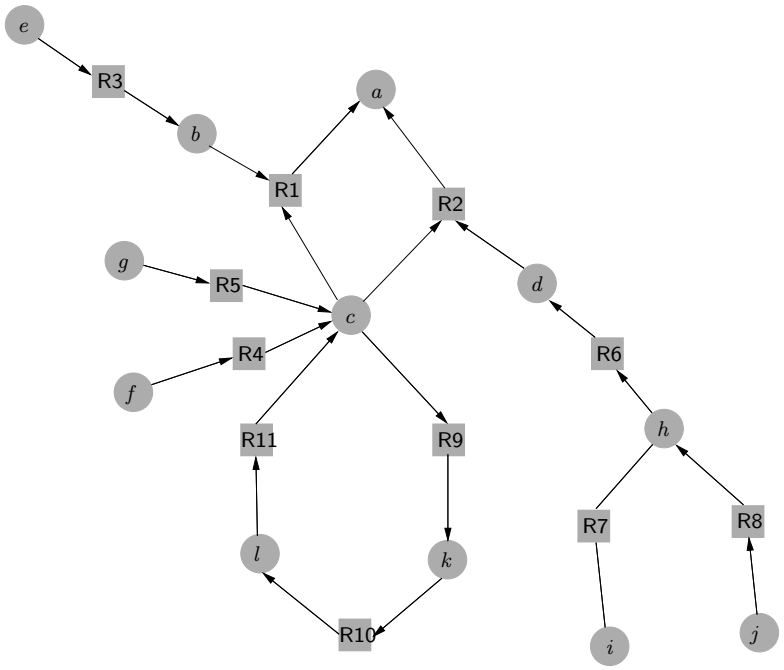


Figura 4.4: O composto a é definido como metabólito alvo. (a) Os precursores potenciais de origem são e, g, f, j e i. O metabólito i pode ser considerado como composto de origem mesmo sendo produzido por R7 devido a estar envolvido em apenas uma reação reversível. Caso não existam precursores potenciais por escolha, as soluções são $\{e, g\}$, $\{e, f\}$, $\{i, g\}$, $\{i, f\}$, $\{j, f\}$ e $\{j, g\}$. (b) Acrescentando l como um precursor potencial os conjuntos $\{e, l\}$, $\{i, l\}$ e $\{j, l\}$ também são soluções. (c) Finalmente, definindo-se g e e como precursores proibidos, as soluções são $\{i, f\}$, $\{j, f\}$, $\{i, l\}$ e $\{j, l\}$.

Antes de apresentar o algoritmo EnumMalCP, faz-se necessário apresentar uma estrutura de dados, chamada **árvore de substituição**, que o algoritmo faz uso para resolver o problema de enumeração de conjuntos minimais de precursores. Cabe ressaltar aqui que esta estrutura de dados foi criada, durante a execução do presente trabalho, com o propósito de servir de apoio à solução do problema enum-mal-CP. Uma árvore de substituições é construída a partir de um metabólito alvo, conforme exibido na figura 4.5, que mostra uma rede metabólica e uma árvore de substituição construída a partir dela.

A idéia para a construção de uma árvore de substituições é que cada metabólito de uma rede metabólica pode ser "substituído" pelos substratos das reações que o sintetizam, ou seja, a presença de um determinado metabólito depende da existência de um conjunto de outros metabólitos.

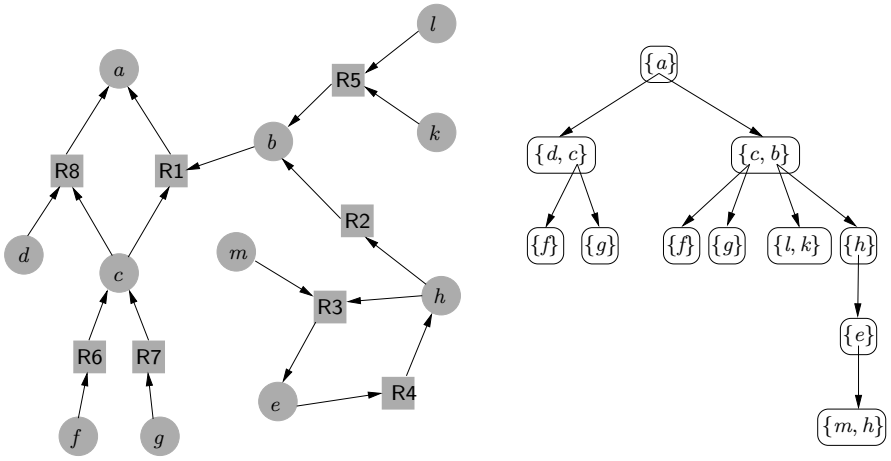


Figura 4.5: Exemplo de uma rede metabólica e de uma árvore de substituição para o metabólito alvo a.

Cada nó de uma árvore de substituições, chamado **nó de substituição**, contém um conjunto de metabólitos e cada metabólito pode ser ramificado de acordo com o número de reações que o sintetizam. A raiz da árvore de substituições conterá o(s) metabólito(s) alvo(s), o segundo nível conterá tantos nós quantas forem as reações que sintetizam o(s) metabólito(s) alvo(s), o nível seguinte conterá os nós representando os substratos necessários à sintetização dos compostos do nível anterior e assim por diante. Chamamos o processo de ramificação de um determinado metabólito em um nó da árvore de **substituição** desse metabólito por seus precursores. O processo de substituições é interrompido quando encontramos um precursor potencial, ou seja, um metabólito que não é mais produzido por nenhuma outra reação e assim não pode mais ser substituído, como por exemplo os metabólitos f, g, l, k e m na figura 4.5, ou um metabólito qualquer definido como precursor potencial. Note ainda que os compostos l e k aparecem dentro de um mesmo nó de substituição, por serem ambos necessários a uma mesma reação – R5 – enquanto os compostos f e g, por exemplo, aparecem sozinhos em nós de substituição, por serem os únicos substratos das reações R6 e R7, respectivamente. Outro caso em que a ramificação da árvore de substituição é interompida ocorre quando em uma mesma ramificação encontramos o mesmo metabólito pela segunda vez, ou seja, identificamos um ciclo, como por exemplo o metabólito h, na figura 4.5. O metabólito encontrado nessa situação não é um precursor potencial, mas sim um candidato a fazer parte do conjunto de metabólitos continuamente disponíveis, que compõem as soluções de conjuntos minimais de precursores (o conjunto Z da definição 4.5). Finalmente, note que nas folhas de uma árvore de substituição existirão apenas precursores potenciais ou metabólitos continuamente disponíveis.

O algoritmo EnumMalCP apresenta o método que, a partir de uma árvore de substituições, visita os nós desta árvore e gera os conjuntos minimais de precursores para os metabólitos alvos. As linhas de 1 a 8 preparam o conjunto de entrada P, que inicialmente contém apenas os metabólitos escolhidos como precursores potenciais, adicionando os metabólitos de origem e removendo os precursores proibidos. Feito isso, faz-se uma iteração para cada metabólito alvo a, construindo-se uma árvore de substituições tendo este metabólito como raiz e, a partir dessa árvore, faz-se uma visitação em suas ramificações para encontrar todos os conjuntos minimais de precursores capazes de sintetizar a. Esse conjunto de soluções para um único metabólito alvo é combinado com os conjuntos de soluções obtidos para cada um dos outros metabólitos alvo, já eliminando-se as soluções não minimais.

ALGORITMO ENUMMALCP(G, P, F, A)

Entrada: Uma rede metabólica $G = ((C, R), E)$, um conjunto $P \subset C$ de precursores potenciais, um conjunto F de precursores proibidos e um conjunto $A \subset C$ de metabólitos alvos;

Saída: Um conjunto X de conjuntos minimais de precursores, que produzam todos os compostos alvos de A.

```
1: para cada c ∈ C faça
2:    RP ← reações que produzem c
3:    se RP = ∅ então
4:       P ← P + c
5:    senão
6:       se |RP| = 1 e r ∈ RP é reversível então
7:          P ← P + c      ▷ Adicionamos a P os metabólitos de origem
8: P ← P \ F      ▷ Removemos de P os precursores proibidos
9: X ← ∅
10: para cada a ∈ A faça
11:    T ← ConstruaÁrvoreSubstituição(G, a, P)
12:    S ← EncontrePrecursores(T, P, a)      ▷ Soluções para o alvo a
13:    se X = ∅ então
14:       X ← S
15:    senão
16:       para cada x ∈ S faça
17:          se x ∉ X então
18:             X ← CartesianoMinimal(X, x)
19: devolva X
```

O processo de obter um conjunto minimal de precursores a partir de uma árvore de substituições construída para um metabólito alvo é detalhado no algoritmo EncontrePrecursores, enquanto o algoritmo CartesianoMinimal é o responsável por combinar ao conjunto de soluções parciais obtidos até um determinado momento o novo conjunto de soluções devolvido pela última chamada EncontrePrecursores, já eliminando neste processo conjuntos não minimais, ou seja, conjuntos de soluções que estejam contidas em outros conjuntos menores de soluções.

Perceba que para obter uma lista com todos os conjuntos minimais de precursores o método não exige que seja computado o escopo do conjunto inicial de compostos de entrada, ou sementes, como ocorria com a chamada ao método de *forward propagation*, como pré-requisito para a determinação dos precursores ausentes, no método proposto por Romero e Karp [27]. Em nosso método, a execução ou não de um método como o *forward propagation* para computar o escopo do conjunto inicial de compostos é opcional e pode ser utilizada com o propósito principal de reduzir o tamanho da rede a ser considerada na computação dos conjuntos minimais de precursores, ao se remover da rede metabólica original todos os compostos e reações ativados pela execução do método de *forward propagation*, já que esses compostos podem ser sintetizados diretamente a partir dos compostos de entrada.

O procedimento recursivo EncontrePrecursores é chamado, pela primeira vez, para encontrar os precursores do metabólito presente na raiz da árvore, isto é, do metabólito alvo para o qual a árvore foi construída. Recursivamente, o algoritmo vai descendo a partir da raiz em direção às folhas. Em algum momento, uma das folhas em um nível n da árvore T conterá um

ALGORITMO ENCONTREPRECURSORES(T, P, m)

Entrada: Uma árvore de substituições T, um conjunto de precursores potenciais P e um metabólito m presente em um dos nós de T;

Saída: Um conjunto minimal de compostos que sintetize m.

```
1: S ← ∅
2: se m está em uma folha de T então
3:    se m ∈ P então
4:       S ← S ∪ {m}
5:    senão
6:       para cada nó n de T que substitui m faça
7:          sn ← ∅
8:          para cada composto c ∈ n faça
9:             sc ← EncontrePrecursores(T, P, c)
10:            CartesianoMinimal(sn, sc)
11:         S ← S ∪ sn
12: devolva S
```

precursor e este será devolvido como sendo o conjunto minimal de precursores para o metabólito que substituiu, isto é, para um dos metabólitos presente no nó de substituição de nível $n-1$, pai da folha de nível n em questão. Esse processo será repetido até que todos os metabólitos presentes neste nó de nível $n-1$ tenham sido analisados e as soluções de cada um deles seja combinada, formando então um conjunto de conjuntos minimais de precursores para um dos metabólitos presentes no nó de substituição de nível $n-2$, pai do nó $n-1$ em questão, e assim sucessivamente até que obtenha-se todos os conjuntos de conjuntos minimais de precursores até a raiz, ou seja, até o metabólito alvo, obtendo-se então a solução para um dos alvos presentes no conjunto A de metabólitos alvos.

ALGORITMO CARTESIANOMINIMAL(N, C)

Entrada: Conjuntos N e C de conjuntos de precursores potenciais;

Saída: Um conjunto minimal de conjuntos de precursores potenciais.

```
1: se N = ∅ então
2:    S ← C
3: senão
4:    S ← N ⊗ C        ▷ Computa o produto cartesiano de N e C
5: Remova os conjuntos não minimais de S
6: devolva S
```

Para exemplificar o método, retornemos ao exemplo de rede metabólica apresentado na figura 4.5. Suponha que queiramos conhecer todos os conjuntos minimais de precursores para o metabólito alvo *a*, sendo que não escolheremos nenhum dos compostos para ser um precursor potencial bem como não proibiremos nenhum deles de sê-lo.

Assim, na execução do algoritmo ENUMMALCP, o conjunto P de precursores potenciais contém apenas os metabólitos de origem, que neste caso são os compostos *d*, *f*, *g*, *l*, *k* e *m*. A árvore de substituição construída na linha 11 do algoritmo ENUMMALCP é idêntica à exibida na figura 4.5 e o resultado final devolvido é o obtido pela execução do procedimento ENCONTREPRECURSORES, já que o conjunto A de metabólitos alvos contém apenas um elemento, o metabólito *a*. Assim, para entendermos o procedimento para este exemplo, devemos acompa-

nhar a execução de EncontrePrecursores, recebendo como entradas a árvore da figura 4.5, o conjunto de precursores P conforme descrito no início deste parágrafo e o metabólito alvo a.

Em um dos primeiros passos do algoritmo EncontrePrecursores, na linha 2, faz-se um teste para verificar se o composto alvo m é uma folha, isto é, se não possui ramificações. Como este não é o caso para o composto a, vamos iterar sobre cada substituição de a existente na árvore. No caso, existem duas substituições, a que leva ao nó com os compostos d e c e a que leva aos compostos c e b. Suponha que o algoritmo seguiu o primeiro caminho. Para cada um dos compostos presentes no nó de substituição, o algoritmo é recursivamente chamado, com o composto em questão fazendo as vezes do parâmetro m. A chamada para o composto d verifica que se trata de um composto folha e, além disso, de um composto que é um precursor potencial. Assim, o conjunto $\{d\}$ é devolvido como sendo o conjunto minimal de precursores necessários para sintetizar o próprio composto d. Esse resultado é armazenado na variável `sc`, que representa as soluções existentes para um composto, e também é utilizado para inicializar a variável `sn`, que contém as soluções para um nó de substituição. É feita então a chamada recursiva passando-se o composto c como o terceiro parâmetro de entrada do algoritmo. Como não se trata de uma folha, serão feitas mais duas chamadas, uma para cada substituição associada a c, que devolverá os conjuntos $\{f\}$ e $\{g\}$ como sendo os conjuntos minimais de precursores necessários para sintetizar o composto c. Assim, faz-se a combinação dos resultados obtidos para o composto d com os resultados obtidos para o composto c, chegando-se aos conjuntos de precursores $\{d,\ f\}$ e $\{d,\ g\}$, que são os conjuntos minimais de precursores para gerar o composto a, levando-se em conta apenas a reação R8. O método fará também a computação e combinação dos resultados obtidos a partir do caminho da reação R1.

O resultado final obtido com este método será o seguinte conjunto de conjuntos minimais de precursores: $\{\{d,\ f\}, \{d,\ g\}, \{f,\ l,\ k\}, \{g,\ l,\ k\}, \{f,\ m\}, \{g,\ m\}\}$. Note, contudo, que nos dois últimos conjuntos de precursores, faz-se necessária a presença do composto h como continuamente disponível, já que é necessária a presença inicial deste composto em complemento ao composto m para que a reação R3 seja disparada e, assim, produza o composto e. Perceba ainda que o encadeamento de reações fará com que R4 seja disparada e o composto continuamente disponível h seja regenerado e, assim, que R2 seja disparado gerando o composto b, que é um dos substratos da reação R1, que sintetiza o composto alvo. A necessidade do composto h para dar início ao ciclo de produção do composto b é biologicamente aceitável, tendo em vista que há a dependência apenas de uma quantidade inicial – que pode estar presente na própria composição inicial da célula – já que haverá a regeneração do composto nas reações cíclicas encadeadas. Na verdade, a sintetização de compostos através de metabólitos continuamente disponíveis que se auto-regeneram é bastante comum nas redes metabólicas dos organismos e um tratamento adequado para este problema é uma das contribuições do método proposto.

Capítulo 5

Resultados Práticos

Este capítulo apresenta alguns resultados biológicos obtidos através do uso da ferramenta construída a partir dos algoritmos descritos no capítulo 4. Tanto a construção da ferramenta quanto a análise dos resultados foram feitos em trabalho conjunto com o doutorando Ludovic Cottret, da Université de Lyon 1, esse último sob a orientação da pesquisadora Marie-France Sagot, da mesma instituição. Os mesmos resultados ora apresentados também podem ser encontrados em [6]. O apêndice A traz detalhes acerca da implementação da ferramenta, bem como um diagrama de classes comentado.

A seção 5.1 apresenta um exemplo do tratamento de dependência cíclica, resultante da abordagem proposta por este trabalho. A seção 5.2 traz um estudo comparativo entre os resultados obtidos com nosso método e os resultados obtidos por Romero e Karp [27]. A seção 5.3 faz um estudo sobre a relação de parasitismo entre a bactéria *Carsonella ruddii* e o seu hospedeiro, ao analisar os nutrientes fornecidos pelo hospedeiro à bactéria.

5.1 Tratamento de Dependências Cíclicas na Rede Metabólica

Conforme mencionado no capítulo 4, uma das contribuições do método ora proposto é a detecção e tratamento das dependências cíclicas entre reações e compostos, que ocorrem em abundância nas redes metabólicas dos organismos. Para ilustrar a nossa abordagem para este problema, vamos utilizar um exemplo para uma rede metabólica hipotética, conforme apresentado na figura 5.1.

Antes de comentar sobre a dependência cíclica apresentada na figura, é importante fazer alusão ao aplicativo CYTOSCAPE [7], utilizado para exibir a rede metabólica contida na figura 5.1. Este aplicativo foi amplamente utilizado durante este trabalho, por oferecer diversos recursos de apoio à visualização e manipulação de dados metabólicos, através da importação direta de arquivos no formato *SBML* [29], padrão para representação de redes metabólicas.

No exemplo da figura 5.1, o composto alvo é o composto Z. Note que o conjunto $\{\mathsf{A}, \mathsf{B}, \mathsf{C}\}$ é um conjunto de precursores para Z, já que a presença de A e B é capaz de fazer com que a reação R3 dispare e produza o composto G, que combinado com C propicia o disparo da reação R2, que produz F, que é suficiente, via reação R1, para produzir Z.

Porém, esta solução apresentada não é a única possibilidade para sintetização do composto Z. A reação R2 depende do composto C e também do composto G. Na solução apresentada, o

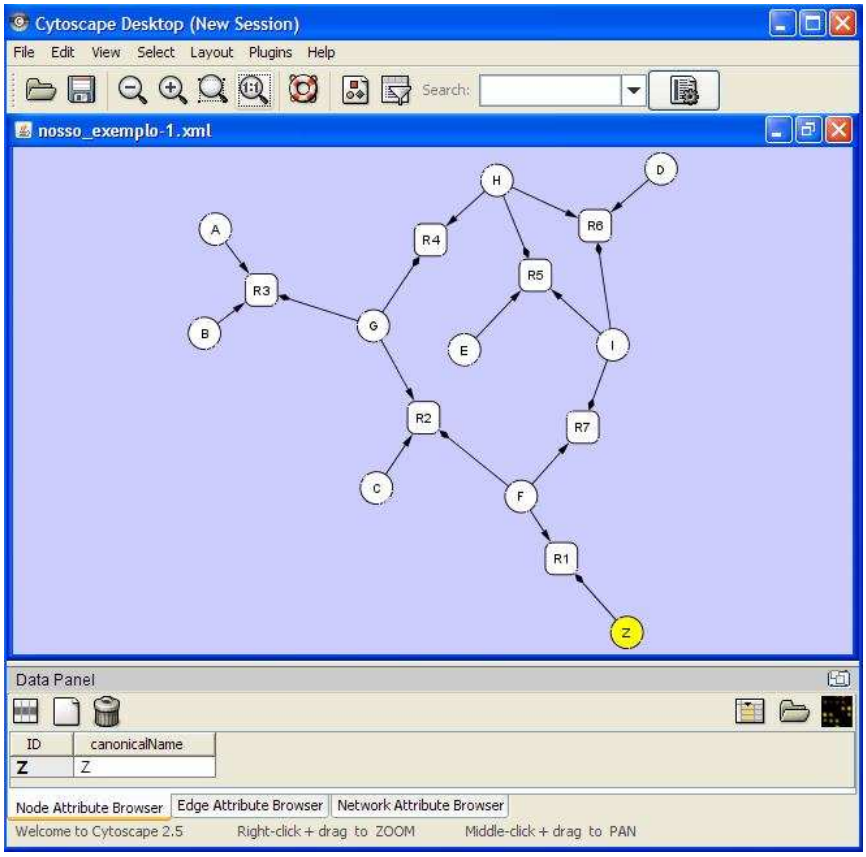


Figura 5.1: Rede metabólica com um exemplo de dependência cíclica.

composto G foi obtido através da reação R3, porém a reação R4 representa uma via alternativa para produção de G. Para tanto, é necessária a presença do composto H, que por sua vez é produzido pela reação R5, que depende dos compostos E e I. O composto E é um precursor topológico enquanto I é gerado pela reação R6, que depende do precursor topológico D e do próprio composto H. Desta forma, identifica-se uma dependência cíclica, já que para sintetizar o composto H é necessária a presença do próprio composto H. Sem considerar o conceito de compostos continuamente disponíveis, o natural seria dizer que todas as reações possuem dependências não satisfeitas, isto é, nem todos os substratos necessários estão disponíveis e, assim, nenhuma nova reação pode ser disparada. Porém, caso seja considerado que o composto H pode ser utilizado para disparar uma via metabólica que sintetize o composto alvo Z, desde que seja capaz de ser regenerado no processo, então passamos a ter uma solução alternativa para a geração do composto alvo Z, que é representada pelo conjunto de precursores $\{C, D, E\}$ e do composto continuamente disponível H.

Os métodos apresentados tanto por Romero e Karp [27] quanto por Handorf et. al [14] não deixam claro se tratam esse problema e como o tratam, caso o façam, apesar de serem aplicados em redes metabólicas de organismos reais que, com certeza, continham diversas incidências de dependências cíclicas entre compostos. Do ponto de vista biológico, esse problema é importante por ser freqüente em redes metabólicas de organismos reais. As soluções alternativas obtidas através do tratamento das dependências cíclicas foram consideradas válidas nas avaliações biológicas dos resultados obtidos.

5.2 Reprodução de Resultados com *Escherichia coli*

Este trabalho deriva diretamente do trabalho realizado por Romero e Karp [27]. Portanto nada mais natural do que iniciar a análise dos resultados de nosso método comparando-os com os obtidos em [27].

O principal objetivo do trabalho de Romero e Karp [27] era identificar inconsistências na base de dados metabólicos ECOCYC [8]. Os autores aplicaram o método *forward propagation* sobre a base de dados do ECOCYC para obter os metabólitos alvos que não foram sintetizados a partir de um conjunto de metabólitos de entrada e de um conjunto de metabólitos de *bootstrap*, isto é, "metabólitos de partida" que poderiam ser utilizados para disparar reações que deles necessitassem. Vale a pena ressaltar que, no método de Romero e Karp, considera-se que esses compostos de *bootstrap* estejam sempre presentes, sem nenhuma restrição de que eles tenham que ser regenerados pela própria atividade metabólica, como em nosso método.

Para tentar reproduzir os resultados e efetuar a comparação, obtivemos os dados da rede metabólica do mesmo organismo utilizados pelos autores de referência, o *Escherichia coli*. A rede metabólica obtida atualmente na base de dados ECOCYC, em sua versão 11.5, contém 897 metabólitos e 879 reações, das quais 104 são irreversíveis. A direção das reações foi definida tanto com informações obtidas no ECOCYC quanto no modelo metabólico MG1655 que contém informação adicional sobre reversibilidade de reações, baseado em restrições termodinâmicas. A figura 5.2 apresenta a rede metabólica do *E. coli* utiliza neste trabalho.

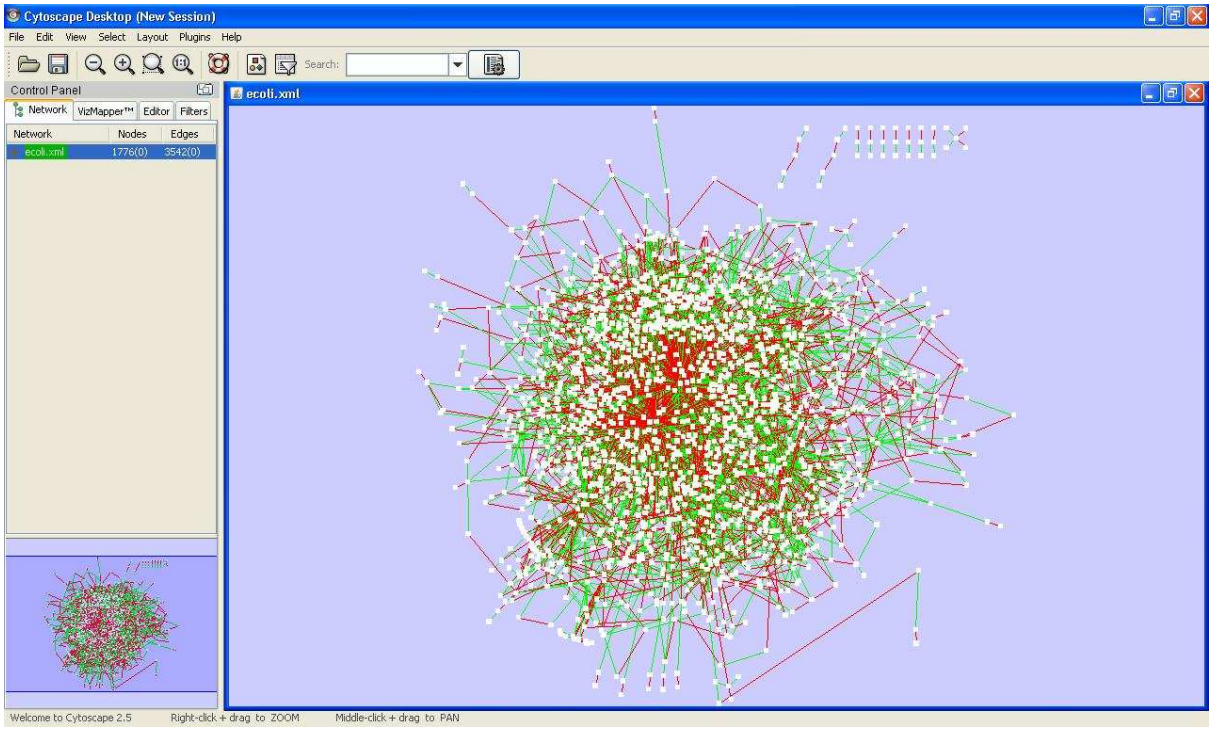


Figura 5.2: Rede metabólica do *E. coli* visualizada no CYTOSCAPE.

Sobre esta rede, aplicamos a nossa implementação do método de *forward propagation*, utilizando os mesmos parâmetros de entrada utilizados em [27]. Como compostos de *bootstrap* definimos os compostos do meio de crescimento mínimo M63, acrescidos de alguns metabólitos

cuja presença na célula é evidente, tal como a coenzima-A, ATP, NAD e oxigênio – já que o meio de crescimento M63 é aeróbio. A lista de compostos utilizados como *bootstrap* é apresentada na tabela 5.1. Como metabólito de entrada para aplicação do método de *forward propagation*, utilizamos apenas glucose. Os compostos alvo foram os 20 aminoácidos.

Compostos de *Bootstrap*
Água
ATP
ADP
Fosfato
Difosfato
NAD
CO_2
NADH
AMP
H+
Coenzima-A
O_2
NAD(P)
NAD(P)H

Tabela 5.1: Compostos de partida – *bootstrap* – definidos para o experimento com o *E. coli*.

O método de *forward propagation* devolveu uma sub-rede com 508 reações e 430 metabólitos, conforme apresentado na figura 5.3. Isso significa que aproximadamente metade dos compostos da rede metabólica podem ser diretamente sintetizados através da injeção de glucose, levando-se em conta a presença dos compostos de *bootstrap* selecionados. Essa sub-rede obtida – o escopo da glucose com os compostos de *bootstrap* selecionados por Romero e Karp – seria ainda maior se houvéssemos adicionado os compostos continuamente disponíveis, isto é, aqueles que têm a capacidade de se auto-regenerar, que são identificados por nosso método. Dentre os 20 aminoácidos definidos como compostos alvo por serem os blocos de construção para a síntese de proteínas, apenas 2 não foram produzidos pelo método de *forward propagation*: lisina e metionina.

Aplicando-se o nosso método para rastreio de precursores ausentes para a lisina, obtém-se 9 conjuntos minimais de precursores ausentes. Dentre eles, um em particular chama atenção por conter somente tetrahidro-dipicolinato (delta1-piperidina-2-6-dicarboxilato). De fato, este metabólito está relacionado à síntese da lisina e, surpreendentemente, esta via metabólica aparece incompleta na base de dados da rede metabólica do *Escherichia coli*. Analisando-se mais atentamente, este metabólito não deveria aparecer como precursor, mas sim como produto da reação 1.3.1.26, porém esta reação aparece como irreversível e na direção inversa do que se conhece quanto a essa via metabólica, ou seja, essa análise revela um erro no ECOCYC. Fazendo com que a reação citada passe a ser definida como reversível, o composto citado passa a aparecer como produto da reação – ao invés de substrato – e tanto a lisina quanto a metionina passam a ser sintetizados pelo método de *forward propagation*.

Apesar de não ser possível reproduzir os mesmos resultados obtidos por Romero e Karp, visto que não há uma cópia da base de dados metabólicos do *E. coli* antes das correções proporcionadas pelos resultados analíticos desde [27], foi possível avaliar que o nosso método de

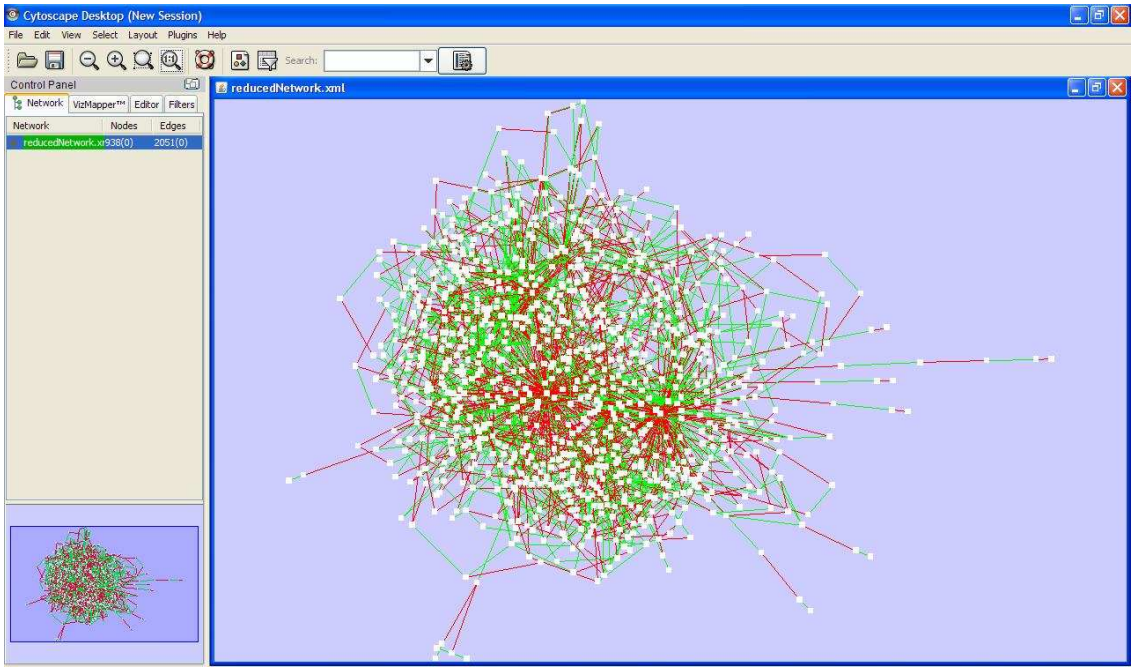


Figura 5.3: Escopo da glucose na rede metabólica do *E. coli*.

análise de nutrientes ausentes também é eficaz para identificar inconsistências em bases de dados metabólicas.

5.3 Análise da Relação de Parasitismo do *Carsonella ruddii*

Uma outra característica para a qual a análise de nutrientes pode ser direcionada é a análise da relação entre parasitas e hospedeiros, ao permitir a identificação dos compostos sintetizados pelo parasita ou obtidos do seu hospedeiro, permitindo assim uma melhor compreensão da relação de dependência entre os dois organismos, de um ponto de vista bioquímico.

A bactéria endosimbiótica *Carsonella ruddii* vive dentro de determinadas células do *psilídeo*, uma espécie de inseto. *C. ruddii* tem a menor rede metabólica conhecida [24], portanto a tendência é de que haja um número de precursores ausentes muito alto para os compostos essenciais da bactéria, os aminoácidos. Estudos recentes [32] demonstram que metade das vias metabólicas dessa bactéria relacionadas à síntese de aminoácidos estão completa ou parcialmente perdidas, por isso é provável que essa bactéria necessite de muitos nutrientes de seu hospedeiro para preencher essas lacunas. Neste sentido, a análise de nutrientes se presta a esclarecer a relação de parasitismo entre os dois organismos através da descoberta de quais são os nutrientes absorvidos pela célula da bactéria a partir das células do inseto hospedeiro.

Para ilustrar, escolhemos procurar pelos precursores ausentes de um desses aminoácidos essenciais, a arginina, cuja via metabólica parece estar completa na rede metabólica da bactéria. A rede metabólica da bactéria contém apenas 130 compostos e 71 reações, 16 delas irreversíveis. A figura 5.4 apresenta a rede metabólica da *C. ruddii*. Para o experimento, consideramos os compostos listados na tabela 5.2 como compostos de partida – *bootstrap*. Foram encontrados 12 conjuntos minimais de precursores ausentes para a arginina, como demonstrado na tabela 5.3.

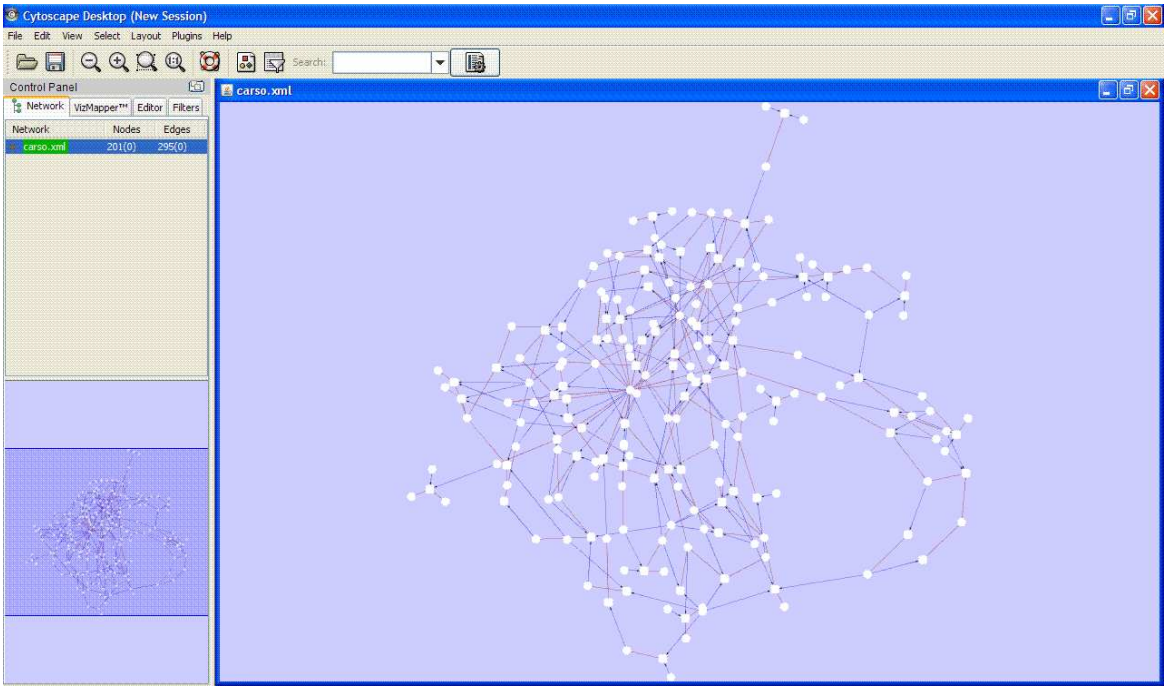


Figura 5.4: Rede metabólica da *C. ruddii* visualizada no CYTOSCAPE.

Uma constatação interessante é o fato dos compostos glutamina, treonina e o íon bicarbonato (HCO3−) estarem presentes em todas as soluções. Glutamina e o íon bicarbonato estão envolvidos na reação 6.3.5.5, que representa um passo essencial na via metabólica de síntese da arginina, como pode ser visto no METACYC [23]. Já o composto treonina merece uma discussão mais aprofundada. O caminho entre a treonina e a arginina inicia com a reação reversível 4.2.3.1 e segue por duas vias metabólicas diferentes, formadas por reações reversíveis, para produzir L-aspartato, um metabólito chave para a síntese da arginina. Um fato interessante é que essas duas vias metabólicas – a biosíntese de treonina a partir de homoserina e a biosíntese de homoserina – são percorridas na direção inversa à que classicamente está indicada na base de dados para essas vias. Naturalmente, isso pode ter ocorrido devido a uma imprecisão na direção das reações mas também pode significar que essas reações podem ser utilizadas na direção inversa à que está indicada nas referências a essas vias metabólicas.

Compostos de *Bootstrap*
Água
ATP
ADP
Fosfato
Amônia (NH3)
CO_2
H+
NADPH
NADP+
Coenzima-A
O_2
NAD(P)
NAD(P)H

Tabela 5.2: Compostos de partida – *bootstrap* – definidos para o experimento com a *C. ruddii*.

Compostos x Conjuntos	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Glutamina	X	X	X	X	X	X	X	X	X	X	X	X
Treonina	X	X	X	X	X	X	X	X	X	X	X	X
Íon Bicarbonato (HCO3−)	X	X	X	X	X	X	X	X	X	X	X	X
Acetato	X	X	X	X		X	X	X	X	X	X	X
2,3-dihidroxi-isovalerato				X		X	X	X	X	X	X	
2,3-dihidroxi-3-metilvalerato	X							X	X			
5-enolpiruvil-shikimato-3-fosfato						X			X		X	
fosfoenolpiruvato				X				X		X		
5,10-metenil-THF	X		X									X
Valina			X							X	X	
2-isopropil-3-oxosuccinato					X							
Piridoxal 5'-fosfato							X					
D-alanina							X					
Piruvato		X										

Tabela 5.3: Conjuntos de precursores ausentes para a arginina no organismo *Carsonella ruddii*.

Capítulo 6

Conclusão

Neste trabalho foi feita uma revisão consistente do tópico de análise de nutrientes utilizando-se conjuntos de precursores, desde uma revisão biológica e de modelagem computacional de redes metabólicas até uma definição matemática dos principais problemas relacionados, discussão de suas complexidades computacionais e a apresentação de algoritmos para identificação de um conjunto minimal de precursores e do primeiro método exato, baseado em topologia de rede, para enumerar todos os conjuntos de precursores para um conjunto de metabólitos alvos. Apesar de se tratar de um problema com complexidade exponencial, o algoritmo desenvolvido pode ser utilizado na prática para as redes metabólicas existentes para os organismos mapeados, produzindo resultados cujos tempos de execução variaram, para os exemplos e testes realizados e apresentados neste trabalho, de poucos segundos a poucas horas. Por exemplo, a busca de precursores para o *E. coli*, que contém 897 metabólitos e 879 reações, para o composto alvo lisina leva cerca de 10 segundos, enquanto a análise de precursores para o mesmo organismo, mas tendo como composto alvo a arginina leva cerca de 10 minutos. Esses resultados foram obtidos utilizando-se uma máquina com processador AMD 64 bits de 2.0GHz com 1MB de memória RAM.

Uma melhoria evidente do método desenvolvido é a maneira formal com que ele lida com as dependências cíclicas entre compostos, fato que é inclusive apontado como uma das inconsistências dos resultados obtidos por Romero e Karp [27]. Para tratar os ciclos, definimos o conceito de metabólitos continuamente disponíveis, que são aqueles que têm a capacidade de se auto-regenerar, uma vez que tenham sido ativados por uma reação. Nossa abordagem tem a vantagem de ser genérica e definida pela própria topologia da rede.

Contudo, as análises feitas até aqui demonstram que alguns conceitos devem ser refinados. Por exemplo, assumir que todos os precursores potenciais estarão em fornecimento infinito pelo ambiente pode não ser válido de um ponto de vista biológico. De fato, alguns nutrientes em determinadas condições podem estar sempre disponíveis no ambiente enquanto outros podem ter um fornecimento limitado.

Outra limitação do método pode ser causada devido ao grande número de reações reversíveis existentes nas redes metabólicas, que podem causar ciclos de auto-regeneração incorretos ou artificiais. Um caso simples, detectado e contornado pelo algoritmo, ocorre na construção da árvore de substituições pois as reações reversíveis fazem com que um produto seja substituído por um substrato e, na substituição seguinte, pelo próprio produto novamente. O contorno é proibir que uma substituição se dê pela própria reação que criou o nó de substituição. Contudo,

outros tipos de ciclos artificias causados por reações reversíveis podem não ser detectados e tratados pelo algoritmo, como o exemplo apresentado na figura 6.1. Neste pequeno exemplo, as reações reversíveis R1 e R3 fazem com que os compostos a e b serão definidos como compostos continuamente disponíveis, durante a construção da árvore de substituições. Desta forma, tendo d como composto alvo, o conjunto de soluções devolvido pelo nosso método é vazio, uma vez que d pode ser substituído por a, b, porém a pode ser substituído por b e vice-versa, devido à reação R1 ser reversível. A solução mais adequada para este exemplo seria o conjunto $\{c\}$.

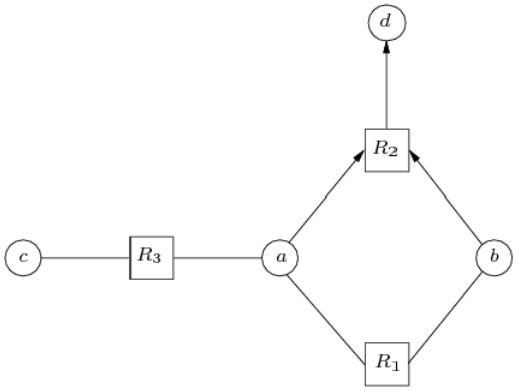


Figura 6.1: Exemplo de um ciclo de auto-regeneração artificial causado por uma reação reversível.

Portanto, o trabalho realizado complementa o trabalho anterior de análise de precursores desenvolvido principalmente por Romero e Karp [27] e Handorf *et al.* [15] [14] e vai além, formalizando e se aprofundando em três dos principais problemas, propondo uma nova abordagem para tratar o mais importante deles, do ponto de vista biológico, que é o problema de conhecer todos os conjuntos minimais de precursores para um conjunto de metabólitos alvos [6].

Algumas alternativas de extensão direta do presente trabalho, do ponto de vista biológico, são oferecer uma melhor abordagem para o problema do ciclo artifical causado por reações reversíveis e para a hipótese dos precursores potenciais terem uma reposição infinita. Neste último caso, uma alternativa interessante e que traria outros benefícios é a introdução de dados estoquiométricos nas reações da rede metabólica, permitindo assim uma análise também quantitativa dos conjuntos de precursores. Finalmente, dados regulatórios que indicam as condições ambientais necessárias à ocorrência das reações também podem ser adicionados, visando dar informação ainda mais confiável sobre os precursores necessários à sintetização dos compostos alvos. Do ponto de vista computacional, refinamentos algorítmicos ou de codificação que visem melhorar o tempo de execução ou o uso de memória são também importantes como trabalhos futuros, principalmente no caso da adição de informações quantitativas ou regulatórias na base de dados metabólicos. Uma outra abordagem interessante é adaptar algoritmos de aproximação existentes para o problema do *Hitting Set* para obter respostas para o problema de encontrar um conjunto mínimo de precursores.

Apêndice A

Apresentação da Ferramenta Web

Esta seção apresenta a ferramenta web para análise de nutrientes desenvolvida durante o trabalho realizado. A seção A.1 apresenta detalhes relativos ao projeto do sistema enquanto a seção A.2 apresenta os meios de utilização da ferramenta web e também as formas para se obter a versão atual do código-fonte, para possíveis extensões.

A.1 Características Técnicas da Ferramenta

Projetamos uma solução orientada a objetos para o problema ENUM-MAL-CP, apresentado no capítulo 4, baseando-nos nos algoritmos apresentados ou sugeridos naquele capítulo. O objetivo é termos uma ferramenta que aja como um "analista de nutrientes", isto é, que forneça respostas sobre quais os conjuntos de precursores necessários para sintetizar um determinado conjunto de compostos alvos, dados uma rede metabólica e alguns compostos de entrada. Sendo assim, batizamos a ferramenta de ***Nutrient Analyst***. A figura A.1 traz um diagrama de classes do projeto *Nutrient Analyst*.

O diagrama não contém todas as classes do projeto, mas sim todas as principais classes definidas para a solução do problema de enumeração de todos os conjuntos minimais de precursores necessários para sintetizar determinados compostos alvos. O projeto foi codificado utilizando-se a linguagem de programação Java. As 8 classes apresentadas no diagrama da figura A.1 contém aproximadamente 2.000 linhas de código.

A classe *NutrientAnalyst* é a responsável por processar a entrada – ler a rede metabólica, os compostos de entrada, os compostos de partida ou *bootstrap* e os compostos alvos – construir uma rede metabólica e obter a lista de precursores para os compostos alvos. Para produzir esta resposta, ela faz uso dos serviços oferecidos pelas demais classes, especialmente da classe *MetabolicNetwork*, que é uma abstração de uma rede metabólica real. Para encontrar precursores ausentes, a classe *MetabolicNetwork* faz uso de uma estrutura de dados especial, chamada árvore de substituições, representada pela classe *ReplacementTree*.

Apresentamos agora uma descrição um pouco mais detalhada do papel desempenhado por cada uma das classes apresentadas na figura A.1 e dos seus principais atributos e métodos. Para efeito de clareza, omitimos os métodos de acesso aos atributos privados das classes – os *getters* e *setters* – e outros métodos necessários apenas para depuração ou exibição de resultados intermediários.

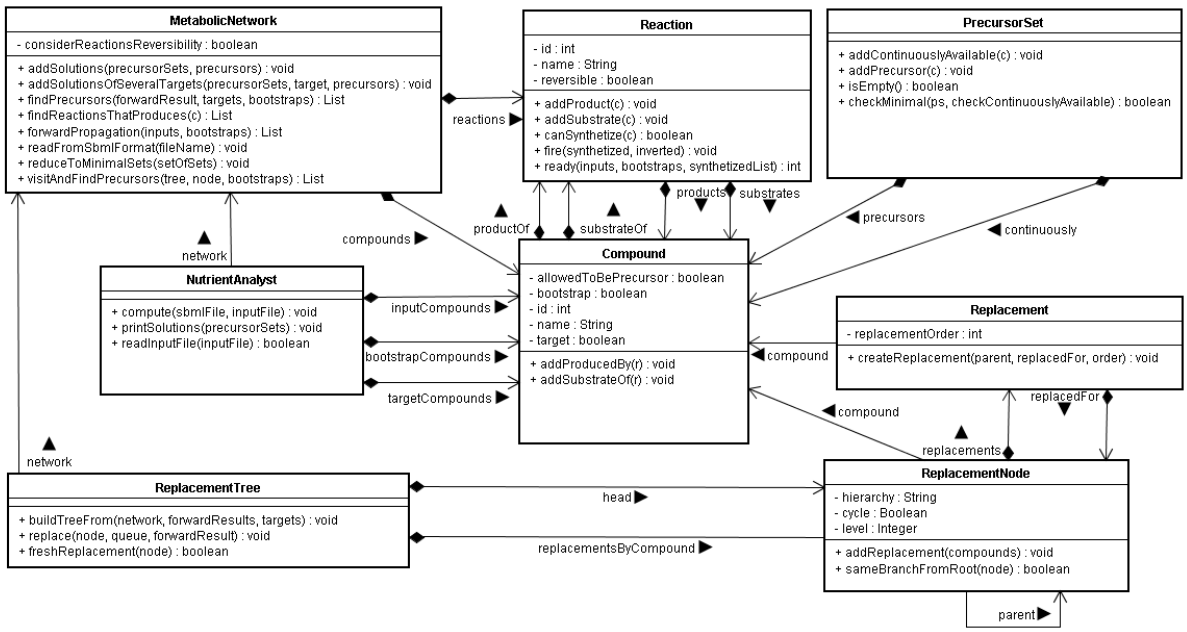


Figura A.1: Diagrama de classes do projeto *Nutrient Analyst*.

A.1.1 Classe *Compound*

A classe *Compound* representa os metabólitos ou compostos químicos, que podem ser tanto substratos ou produtos de uma reação, quanto parte do conjunto inicial de compostos disponíveis em um meio de crescimento, ou ainda compostos alvos de um experimento de análise de nutrientes, etc. A tabela A.1 apresenta os principais atributos e métodos da classe *Compound*.

Atributos	**Estrutura de Dados**	**Tipo de Dados**
allowedToBePrecursor	escalar	booleano
bootstrap	escalar	booleano
id	escalar	cadeia de caracteres
name	escalar	cadeia de caracteres
producedBy	Lista	*Reaction*
substrateOf	Lista	*Reaction*
target	escalar	booleano
Método	**Devolução**	**Parâmetros**
addProducedBy	nenhum	*Reaction*: *r*
addSubstrateOf	nenhum	*Reaction*: *r*

Tabela A.1: Atributos e métodos da classe *Compound*.

Os atributos *allowedToBePrecursor*, *bootstrap*, *target*, *id* e *name* são atribuídos no momento da criação de cada novo objeto, de acordo com os parâmetros de entrada fornecidos e representam, respectivamente, a propriedade de ser um composto proibido ou não, de ser avaliado como precursor, de ser um composto de partida para as reações na fase de propagação para frente da rede, de ser um composto definido como alvo e a identificação e nome do composto. Compostos podem ser desabilitados como precursores caso não se esteja interessado em respostas que o

contenham, como ocorre para compostos sabidamente necessários para a síntese de quase todos os compostos, como água ou ATP.

Os outros atributos são listas que mantêm uma relação do composto com as reações que o sintetizam, mapeados no atributo *producedBy*, ou que o consomem, mapeados no atributo *substrateOf*. O método *addProducedBy* inclui uma reação na lista *producedBy* enquanto o método *addSubstrateOf* inclui uma reação na lista *substrateOf*.

A.1.2 Classe *Reaction*

A classe *Reaction* representa as reações químicas, responsáveis por transformar um conjunto de compostos de entrada em um conjunto de compostos de saída, ou então, um conjunto de substratos em produtos. A tabela A.2 apresenta os principais atributos e métodos da classe *Reaction*.

Atributos	**Estrutura de Dados**	**Tipo de Dados**
id	escalar	cadeia de caracteres
name	escalar	cadeia de caracteres
products	Lista	*Compound*
reversible	escalar	booleano
substrates	Lista	*Compound*
Métodos	**Devolução**	**Parâmetros**
addProduct	nenhum	*Compound*: *c*
addSubstrate	nenhum	*Compound*: *c*
canSynthetize	inteiro	*Compound*: *c*
fire	nenhum	Lista de *Compound*: *synthetized* booleano: *inverted*
ready	inteiro	Lista de *Compound*: *inputList* Lista de *Compound*: *bootstrapList* Lista de *Compound*: *synthetized*

Tabela A.2: Atributos e métodos da classe *Reaction*.

Os atributos *id*, *name* e *reversible* são atribuídos no momento da criação de cada novo objeto, de acordo com os parâmetros de entrada fornecidos e representam, respectivamente, a identificação e nome da reação e a propriedade de ser uma reação reversível ou não, o que determina se os compostos definidos como substratos podem também ser considerados produtos e vice-versa. Os outros atributos são listas que mantêm uma relação da reação com os compostos que ela consome, mapeados no atributo *substrates*, e que produz, mapeados no atributo *products*. O método *addProduct* inclui um composto na lista *products* enquanto o método *addSubstrate* inclui um composto na lista *substrates*.

O método *canSynthetize* verifica se a reação é capaz de sintetizar um composto *c*, passado como parâmetro. O teste efetuado consiste em verificar se *c* está contido na lista de produtos da reação – neste caso, o método devolve o inteiro 1 indicando que a reação sintetiza diretamente o composto *c* – ou se está contido na lista de substratos e a reação está marcada como reversível – neste caso, o método devolve o inteiro 2 indicando que a reação sintetiza indiretamente o composto *c*. Caso nenhuma das duas condições ocorra, o método devolve zero. Este método é utilizado na construção da árvore de substituições para obter todas as reações que sintetizam

determinado composto, determinando assim as ramificações necessárias na árvore. Para mais detalhes, vide a documentação da classe *ReplacementTree*.

Os métodos *ready* e *fire* são utilizados combinadamente na implementação do algoritmo de *forward propagation*, detalhado no método *forwardPropagation* da classe *MetabolicNetwork*. O método *ready* valida se a reação está pronta para ser disparada, ou seja, se todos os seus substratos – ou ainda os seus produtos, caso a reação seja reversível – estão disponíveis no escopo atual de propagação. Este escopo atual é passado como parâmetro para o método, através das listas de compostos *inputList*, *bootstrapList* e *synthetized*, que representam, respectivamente, os compostos iniciais disponíveis no meio de crescimento, os compostos de partida e os compostos produzidos pela rede até o momento. O método devolve o inteiro 1 caso a reação esteja pronta para disparar na direção em que foi definida, isto é, utilizando os compostos do atributo *substrates* como os seus substratos e devolve 2 caso a reação esteja pronta para disparar na direção contrária à que foi definida, isto é, utilizando os compostos do atributo *products* como os seus substratos. Finalmente, o método devolve zero caso a reação não esteja pronta para ser disparada.

O método *fire*, por sua vez, deve ser chamado apenas para reações que estejam prontas para serem disparadas – isto é, primeiramente deve-se validar essa condição através do método *ready* – e adiciona ao conjunto de compostos sintetizados os produtos da própria reação, considerando a direção em que a reação foi disparada, através do parâmetro *inverted*, que sinaliza se os compostos produzidos são os contidos no atributo *products* ou no atributo *substrates*.

A.1.3 Classe *PrecursorSet*

A classe *PrecursorSet* representa uma parte da solução do problema de análise de nutrientes estudado, ou seja, corresponde a um conjunto de precursores acrescidos do conjunto de compostos continuamente disponíveis necessários para que os compostos alvos sejam sintetizados. A tabela A.3 apresenta os principais atributos e métodos da classe *PrecursorSet*.

Atributos	**Estrutura de Dados**	**Tipo de Dados**
precursors	Lista	*Compound*
continuouslyAvailable	Lista	*Compound*
Métodos	**Devolução**	**Parâmetros**
addPrecursor	nenhum	*Compound*: *c*
addContinuouslyAvailable	nenhum	*Compound*: *c*
checkMinimal	booleano	*PrecursorSet*: *ps* booleano: *lookContinuouslyAvailable*
isEmpty	booleano	nenhum

Tabela A.3: Atributos e métodos da classe *PrecursorSet*.

Basicamente, os objetos da classe *PrecursorSet* contêm duas listas de compostos, uma delas para indicar quais são os precursores contidos na solução – o atributo *precursors* – e a outra para guardar os compostos continuamente disponíveis necessários para que os precursores sejam capazes de sintetizar os compostos alvos – o atributo *continuouslyAvailable*. Os métodos *addPrecursor* e *addContinuouslyAvailable* servem apenas para inserir novos compostos nas listas.

O método *isEmpty* devolve verdadeiro quando tanto o conjunto de precursores quanto o conjunto de compostos continuamente disponíveis estiver vazio. O método *checkMinimal*, por

sua vez, devolve verdadeiro caso a lista de precursores e de compostos continuamente disponíveis do objeto estejam todos contidos no conjunto de precursores e de compostos continuamente disponíveis do parâmetro *ps*. Caso o valor do parâmetro *lookContinuouslyAvailable* seja falso, apenas o conteúdo dos conjuntos de precursores serão analisados para validar se o objeto é minimal com relação ao outro.

A.1.4 Classe *MetabolicNetwork*

A classe *MetabolicNetwork* representa uma rede metabólica, com seu conjunto de compostos e reações. A tabela A.4 apresenta os principais atributos e métodos da classe *MetabolicNetwork*.

Atributos	**Estrutura de Dados**	**Tipo de Dados**
compounds	Tabela de Dispersão	*Compound*
considerReactionsReversibility	Escalar	booleano
reactions	Tabela de Dispersão	*Reaction*
Métodos	**Devolução**	**Parâmetros**
addSolutions	nenhum	Lista de *PrecursorSet*: *precursorSets* Lista de *PrecursorSet*: *precursors*
addSolutionsOfSeveralTargets	nenhum	Lista de *PrecursorSet*: *precursorSets* *Compound*: *target* Lista de *PrecursorSet*: *precursors*
findPrecursors	Lista de *PrecursorSet*	Lista de *Compound*: *forwardResult* Lista de *Compound*: *targets* Lista de *Compound*: *bootstraps*
findReactionsThatProduces	Lista de *Reaction*	*Compound*: *c*
forwardPropagation	Lista de *Compound*	Lista de *Compound*: *inputList* Lista de *Compound*: *bootstrapList*
readFromSbmlFormat	nenhuma	Cadeia de caracteres: *fileName*
reduceToMinimalSets	nenhuma	Lista de *PrecursorSet*: *setOfSets*
visitAndfindPrecursors	Lista de *PrecursorSet*	*ReplacementTree*: *tree* *ReplacementNode*: *node* Lista de *Compound*: *bootstraps*

Tabela A.4: Atributos e métodos da classe *MetabolicNetwork*.

Os atributos *compounds* e *reactions* são tabelas de dispersão, indexadas, respectivamente, pelo atributo *id* dos objetos das classes *Compound* e *Reaction*. Assim, a rede metabólica mantém uma lista de todos os seus compostos e suas reações e consegue acessar um objeto específico de forma eficiente, através da tabela de dispersão. O atributo *considerReactionsReversibility* considera se a rede metabólica deve considerar a reversibilidade de suas reações, ao computar o resultado da propagação para frente. Trata-se, portanto, de uma configuração do funcionamento do processamento do método *forwardPropagation*.

O método *readFromSbmlFormat* é o responsável preencher um objeto da classe *MetabolicNetwork*, ao fazer a leitura dos dados contidos em um arquivo que segue o padrão *Systems Biology Markup Language*, que descreve de maneira estruturada, em um arquivo texto devidamente formatado, as informações de uma rede metabólica. Deste arquivo são extraídos os compostos e reações que foram a rede metabólica.

O método *forwardPropagation* devolve o escopo de um conjunto inicial de compostos fornecido como parâmetro de entrada, *inputList*. O algoritmo implementado faz uma iteração pelas reações da rede metabólica, verificando se há reação que está pronta para ser disparada, dados o conjunto inicial de compostos, o conjunto de metabólitos de partida – *bootstrapList* – e os compostos já produzidos até o momento. Para fazer essa validação, o método *ready* da classe *Reaction* é acionado para cada reação da rede metabólica. Para toda reação que a chamada *ready* devolve verdadeiro, o método *fire* da reação é acionado para que os compostos sintetizados pela reação sejam adicionados à lista de compostos já produzidos, que é devolvida ao final do processamento, quando nenhuma nova reação puder ser disparada. É importante notar que o método *ready* avalia se a reação é reversível e, caso seja, se os produtos podem ser utilizados como reagentes para que a reação seja disparada no sentido inverso.

O método *findPrecursors* é um dos principais do projeto, visto que é o responsável por enumerar – e devolver – os conjuntos de precursores necessários para sintetizar um conjunto de compostos alvos, sendo que o algoritmo implementado é muito similar ao pseudo-código apresentado no algoritmo EnumMalCP da seção 4.4, isto é, cria-se uma árvore de substituição – um objeto da classe *ReplacementTree* – para os compostos alvos e para cada composto alvo contido no nó raiz da árvore de substituição chama-se o método recursivo *visitAndFindPrecursors*, que realiza o percurso na direção das folhas da árvore e monta os conjuntos de precursores que sintetizam os alvos. O método *visitAndFindPrecursors* remonta ao algoritmo EncontrePrecursores, também apresentado na seção 4.4, e sua implementação faz uso dos métodos *addSolutions*, que é responsável por incluir uma nova solução parcial encontrada em um conjunto pré-existente de soluções, e *reduceToMinimalSets*, que elimina de uma lista de conjuntos de precursores as soluções não-minimais. A combinação dos métodos *addSolutions* e *reduceToMinimalSets* corresponde ao algoritmo CartesianoMinimal, da seção 4.4.

A.1.5 Classe *ReplacementNode*

A classe *ReplacementNode* representa um nó da árvore de substituições. A tabela A.5 apresenta os principais atributos e métodos da classe *ReplacemenNode*.

Atributos	**Estrutura de Dados**	**Tipo de Dados**
compound	Objeto	*Compound*
cycle	Escalar	booleano
hierarchy	Escalar	cadeia de caracteres
level	Escalar	inteiro
parent	Objeto	*ReplacementNode*
replacements	Lista	*Replacement*
Métodos	**Devolução**	**Parâmetros**
addReplacement	nenhum	Lista de *Compound*: *replacement*
sameBranchFromRoot	booleano	*ReplacementNode*: *node*

Tabela A.5: Atributos e métodos da classe *ReplacementNode*.

Cada nó da árvore de substituições contém um composto que está sendo substituído – o atributo *compound*, que é um objeto da classe *Compound* – e a lista de substituições possíveis para este composto – o atributo *replacements*, que é uma lista de objetos da classe *Replacement*. O atributo *cycle* contém o valor verdadeiro caso o nó constitua um ciclo na árvore de substituições

sendo percorrida. O atributo *level* contém o valor do nível do nó com relação à raiz, que possui nível igual a um. O atributo *parent* referencia o nó ancestral, que foi substituído pelo nó atual. O nível de um novo nó é sempre o nível do nó ancestral *parent* acrescido de um. O atributo *hierarchy* contém uma "identificação"única e hierárquica para cada nó, que é formado pela ordem do composto e da reação que o substituiu. Na prática, a hierarquia é montada a partir da hierarquia do nó ancestral *parent* acrescido do número de ordem desse nó dentro da substituição. Por exemplo, suponha que para uma determinada rede metabólica estejamos interessados apenas em avaliar um composto alvo A contido em um nó de substituição *n*. A hierarquia deste nó *n* será definida como "1". Se houverem, digamos, duas reações que sintetizem A, então o nó *n* conterá duas substituições em sua lista de substituições *replacements*. Se a segunda dessas reações utilizar dois substratos para sintetizar A, digamos *B* e *C*, então os nós criados para cada um destes substratos terá hierarquia "1.2.1"e "1.2.2", respectivamente, indicando que são substituições do primeiro composto alvo, pela segunda reação e o terceiro índice é apenas um número seqüencial do substrato dentro da reação.

O atributo *hierarchy* possui um papel importante na implementação do método *sameBranchFromRoot*, que deve devolver verdadeiro sempre que o nó passado como parâmetro de entrada, *node*, estiver na mesma ramificação do nó objeto. Esse teste é feito pelo método *replace* da classe *ReplacementTree* para verificar se um ciclo foi detectado. O teste é facilitado através da simples comparação dos atributos *hierarchy* dos dois objetos, validando se a hierarquia do objeto começa – é uma subcadeia – da hierarquia de *node*.

Finalmente, o método *addReplacement* toma como parâmetro de entrada uma lista de compostos *replacement* e cria uma nova substituição para o composto, ao criar um novo nó de substituição para cada composto contido em *replacement*, colocando-os como nós filhos do nó sendo substituído. Lembre-se que cada substituição contida no atributo *replacements* representa uma reação que sintetiza o composto referenciado por *compound*, assim, na verdade, os compostos contidos no parâmetro *replacement* são os substratos de cada uma dessas reações.

A.1.6 Classe *ReplacementTree*

A classe *ReplacementTree* representa uma árvore de substituições, estrutura de dados especialmente desenvolvida para este trabalho que representa a seqüência de substituições de determinados compostos, decorrentes das reações que sintetizam estes compostos. A tabela A.6 apresenta os principais atributos e métodos da classe *ReplacementTree*.

O atributo *head* representa a raiz da árvore, é uma lista com todos os compostos alvos, representados como nós da árvore. Cada nó derivado deste nó raiz representa uma substituição possível do composto alvo por compostos capazes de sintetizá-lo. As folhas da árvore de substituição contêm apenas compostos definidos como precursores potenciais ou compostos candidatos a serem compostos continuamente disponíveis, assim identificados pelo próprio algoritmo, quando detecta ciclos na construção da árvore de substituições. O atributo *replacementsByCompound* é uma tabela de dispersão indexada pelo *id* de um objeto da classe *Compound*, que guarda uma lista de objetos da classe *ReplacementNode*, contendo assim todas as substituições já realizadas, na árvore, para cada composto. O atributo *network* é apenas uma referência à rede metabólica para a qual a árvore está sendo construída.

O método *buildTreeFrom* é o responsável por construir a árvore de substituições para a rede metabólica *network*, conhecendo o resultado da fase de propagação para frente e também a lista de compostos alvos através dos parâmetros de entrada *forwardResult* e *targets*, respectivamente.

Atributos	**Estrutura de Dados**	**Tipo de Dados**
head	Lista	*ReplacementNode*
replacementsByCompound	Tabela de Dispersão	Lista de *ReplacementNode*
network	Objeto	*MetabolicNetwork*
Métodos	**Devolução**	**Parâmetros**
buildTreeFrom	nenhum	*MetabolicNetwork*: *network* Lista de *Compound*: *forwardResult* Lista de *Compound*: *targets*
freshReplacement	booleano	*ReplacementNode*: *node*
replace	nenhum	*ReplacementNode*: *node* Lista de *Compound*: *forwardResult* Fila de *ReplacementNode*: *queue* Lista de *Compound*: *forwardResult*

Tabela A.6: Atributos e métodos da classe *ReplacementTree*.

O algoritmo consiste em criar um nó da árvore – chamado de nó de substituição e representado por um objeto da classe *ReplacementNode* – para cada composto definido como alvo. Esses nós criados para os compostos alvos formarão a raiz da árvore e serão também adicionados a uma fila de processamento. Enquanto esta fila contiver elementos, o método *replace* deve ser chamado para realizar a substituição do nó no topo da fila. Esse processo de substituição consiste em criar novos nós, que serão descendentes do nó sendo substituído, um para cada reação que sintetize o composto representado no nó substituído. Desta forma, o papel do método *replace* é o de procurar todas as reações que sintetizem o composto contido no nó de substituição *node*, passado como parâmetro de entrada, e para cada uma delas adicionar uma substituição ao nó *node*, criando um novo nó para abrigar todos os substratos da reação que sintetize o composto contido no nó *node*. O método *replace* não faz a substituição quando o composto contido em *node* for um precursor. O método *replace* possui ainda a função de realimentar a fila de nós de substituições a serem investigadas, adicionando a ela os novos nós criados, desde que se trate de substituições "frescas", o que é validado pelo método *freshReplacement*, que verifica se esta é a primeira substituição deste mesmo composto por esta reação, na mesma ramificação da árvore. Caso não seja, detectou-se um ciclo e o composto contido neste nó de substituição é identificado como pertencente a um ciclo e é um candidato a composto continuamente disponível, sendo que este nó de substituição não é incluído na fila de nós de substituição a serem processados.

A.1.7 Classe *Replacement*

A classe *Replacement* representa uma substituição de um nó da árvore de substituições. Cada nó da árvore de substituições é substituído tantas vezes quantas forem as reações que sintetizam o composto que eles guardam. Cada substituição contém uma lista dos nós de substituição criados, um para cada substrato da reação representada pela substituição. A tabela A.7 apresenta os principais atributos e métodos da classe *Replacement*.

O atributo *compound* contém o objeto da classe *Compound* que está sendo substituído. O atributo *replacementOrder* indica a ordem seqüencial em que esta substituição foi adicionada à lista de substituições do composto *compound*. A lista *replacedFor* contém os nós de substituição criados para a substituição, um para cada substrato da reação representada pela substituição.

Atributos	**Estrutura de Dados**	**Tipo de Dados**
compound	Objeto	*Compound*
replacementOrder	Escalar	inteiro
replacedFor	Lista	*ReplacementNode*
Métodos	**Devolução**	**Parâmetros**
createReplacement	nenhum	*ReplacementNode*: *parent* Lista de *Compound*: *replacedFor* Inteiro: *replacementOrder*

Tabela A.7: Atributos e métodos da classe *Replacement*.

Basicamente, uma substituição não possui comportamentos especiais e o seu único método é o seu construtor, aqui representado pelo método *createReplacement*, responsável por fazer a criação dos objetos necessários a partir dos parâmetros de entrada. Uma substituição é sempre criada através de chamadas ao método *addReplacement* da classe *ReplacementNode*, que por sua vez é invocado através do método *replace* da classe *ReplacementTree*.

A.1.8 A Classe *NutrientAnalyst*

A classe *NutrientAnalyst* representa um analista de nutrientes, ou seja, alguém interessado em realizar experimentos em redes metabólicas, particularmente interessado em realizar propagações para frente seguidas da enumeração de precursores ausentes para determinados compostos alvos. Podemos dizer que esta é a classe principal do projeto, concentrando nela os algoritmos responsáveis por "ligar as pontas" das demais classes, cujas responsabilidades são estruturais ou de prover os serviços necessários às ações da classe *NutrientAnalyst*. A tabela A.8 apresenta os principais atributos e métodos da classe *NutrientAnalyst*.

Atributos	**Estrutura de Dados**	**Tipo de Dados**
bootstrapCompounds	Lista	*Compound*
inputCompounds	Lista	*Compound*
targetCompounds	Lista	*Compound*
network	Objeto	*MetabolicNetwork*
Métodos	**Devolução**	**Parâmetros**
compute	nenhum	Cadeia de caracteres: *sbmlFile* Cadeia de caracteres: *inputFile*
printSolutions	nenhum	Lista de *PrecursorSet*: *precursorSets*
readInputFile	nenhum	Cadeia de caracteres: *inputFile*

Tabela A.8: Atributos e métodos da classe *NutrientAnalyst*.

O método *compute* é o responsável por realizar um experimento sobre uma rede metabólica de entrada – definida no formato *SBML* e contida em um arquivo texto endereçado pelo parâmetro de entrada *sbmlFile* – a partir das informações de entrada, que são uma lista de compostos de entrada, uma lista de compostos de partida e uma lista de compostos alvos, todas contidos no arquivo texto endereçado pelo parâmetro de entrada *inputFile*, um arquivo *XML* criado para descrever a entrada de um experimento. A seção A.2 apresenta os formatos destes arquivos com mais detalhes.

O método *compute* faz então a carga dos dados destes arquivos, preenchendo os atributos *bootstrapCompounds*, *inputCompounds* e *targetCompounds*, a partir dos dados contidos no arquivo de entrada e também criando o atributo *network*, através de uma chamada ao método *readFromSbmlFormat*, da classe *MetabolicNetwork*. Uma vez conhecidos a rede metabólica, os compostos de entrada e de partida e os compostos alvos, o experimento tem início, através da chamada ao método *forwardPropagation* do objeto *network* e, em seguida, do método *findPrecursors* do mesmo objeto. O método *printSolutions* é utilizado para exibir os resultados obtidos pela chamada a *findPrecursors*.

A.2 Como Utilizar

A.2.1 Arquivos de Entrada

Para fazer uso do aplicativo *NutrientAnalyst*, é necessário fornecer como entrada dois arquivos, um descrevendo a rede metabólica e outro descrevendo os parâmetros de entrada para o experimento a ser realizado sobre os dados desta rede.

A rede metabólica é representada pelo formato *Systems Biology Markup Language* (*SBML*) [29], que é um arquivo de marcações – *tags* – com formato *XML*, contendo seções para compostos – *species*, no formato *SBML* – e reações, além dos compartimentos intra-celulares em que os compostos são encontrados, dados regulatórios sobre a ocorrência das reações, dados estoquiométricos, anotações diversas, etc. Uma especificação completa e detalhada sobre o formato *SBML* pode ser encontrada em [29]. O interpretador implementado para o projeto *NutrientAnalyst* faz a leitura apenas das seções *listofSpecies* e *listOfReactions*, para construir o objeto da classe *MetabolicNetwork*. Todas as demais seções do arquivo de entrada são ignoradas no momento do processamento do arquivo. A figura A.2 apresenta um exemplo do conteúdo de um arquivo *SBML*, para uma rede metabólica fictícia.

O arquivo de entrada foi definido especificamente para o projeto *NutrientAnalyst*, mas mantém o formato *XML* e contém uma estrutura bastante similar à do formato *SBML*. Basicamente, o arquivo de entrada possui quatro seções principais, cada uma delas contendo uma lista de compostos. As seções são *input-compounds*, *bootstrap-compounds*, *precursor-compounds* e *target-compounds*, a primeira contendo os compostos disponíveis no meio de crescimento, a segunda os compostos de partida, a terceira os compostos definidos arbitrariamente como precursores e a quarta os compostos alvos. Um exemplo de um arquivo de entrada é apresentado na figura A.3. Para realizar experimentos com a ferramenta, o usuário deverá construir os seus próprios arquivos de entrada, nos moldes do arquivo apresentado na figura A.3.

A.2.2 Versão *Offline*

Para fazer uso do aplicativo *NutrientAnalyst* o interessado pode fazer o seu download a partir do sítio de biologia computacional da UFMS.

Para se realizar um experimento é necessário possuir os arquivos de entrada – a definição da rede metabólica no formato *SBML* e os parâmetros de entrada no formato *XML* – conforme apresentados na seção A.2.1. Para se obter uma enumeração dos precursores ausentes para uma determinada rede e compostos alvos, conforme definidos em arquivos chamados *rede.xml* e *entrada.xml*, por exemplo, basta executar na linha de comando:

```
<?xml version="1.0"?>
<sbml xmlns="http://www.sbml.org/sbml/level2" version="1" level="2" xmlns:html="http://www.w3.org/1999/xhtml">
<model id="ecoli_metabolic_network" name="Escherichia coli">
<listOfCompartments>
  <compartment id="cytoplasm"/>
</listOfCompartments>
<listOfSpecies>
    <species id="GLUCURONATE" name="glucuronate" initialAmount="0" compartment="cytoplasm" boundaryCondition="false"/>
    <species id="L__45__XYLULOSE" name="L-xylulose" initialAmount="0" compartment="cytoplasm" boundaryCondition="false"/>
    <species id="CARBON__45__MONOXIDE" name="CO" initialAmount="0" compartment="cytoplasm" boundaryCondition="false"/>
  </listOfSpecies>
<listOfReactions>
  <reaction id="O__45__SUCCINYLBENZOATE__45__COA__45__SYN__45__RXN" name="NA" reversible="true">
    <notes>
      <html:p>GENE_ASSOCIATION: ( EG11532 )</html:p>
      <html:p>PROTEIN_ASSOCIATION: ( O-SUCCINYLBENZOATE-COA-SYN-MONOMER )</html:p>
      <html:p>SUBSYSTEM: menaquinone biosynthesis</html:p>
      <html:p>SUBSYSTEM: superpathway of chorismate</html:p>
      <html:p>PROTEIN_CLASS: NA</html:p>
    </notes>
    <listOfReactants>
      <speciesReference species="SUCCINYL__45__OH__45__CYCLOHEXADIENE__45__COOH" stoichiometry="1"/>
    </listOfReactants>
    <listOfProducts>
      <speciesReference species="O__45__SUCCINYLBENZOATE" stoichiometry="1"/>
      <speciesReference species="WATER" stoichiometry="1"/>
    </listOfProducts>
  </reaction>
</listOfReactions>
</model>
</sbml>
```

Figura A.2: Exemplo de um arquivo de entrada contendo uma rede metabólica no formato *SBML*.

```
$./nutrientAnalyst rede.xml entrada.xml
```

A.2.3 Versão *Online*

Uma versão *web* do aplicativo *NutrientAnalyst* também está disponível no sítio de biologia computacional da UFMS. A mesma observação feita para a versão *offline* na seção A.2.2, sobre os arquivos de entrada, é válida para a versão *online* da ferramenta.

A tela através da qual o usuário faz a carga dos arquivos de entrada é apresentada na figura A.4 e a apresentação do resultado do processamento é apresentado na figura A.5.

```
<?xml version="1.0"?>
<inputs date="2008-01-24" comment="input file designed to search for the precursors in the whole network of E. coli">
  <input-for-model id="ecoli_metabolic_network" name="E. coli" />
  <input-compounds>
    <species id="GLC-6-P" comment="glucose" />
  </input-compounds>
  <bootstrap-compounds>
    <species id="WATER" comment="WATER (1)" />
    <species id="ATP" comment="ATP (2)" />
    <species id="ADP" comment="ADP (3)" />
    <species id="|Pi|"  comment="Phosphate inorganique (4)" />
    <species id="PPI" comment="Diphosphate (5)" />
    <species id="NAD" comment="NAD+ (6)" />
    <species id="CARBON-DIOXIDE"  comment="CO2 (7)" />
    <species id="NADH"  comment="NADH (8)" />
    <species id="AMP" comment="AMP (9)" />
    <species id="PROTON"  comment="H+ (10)" />
    <species id="CO-A"  comment="Coenzyme A" />
    <species id="OXYGEN-MOLECULE" comment="O2" />
  </bootstrap-compounds>
  <precursor-compounds>
    <species id="GLC-6-P" comment="glucose" />
  </precursor-compounds>
  <target-compounds>
    <species id="L-ALPHA-ALANINE" comment="AA" />
    <species id="LYS" comment="AA e" />
    <species id="L-ASPARTATE" comment="AA" />
  </target-compounds>
</inputs>
```

Figura A.3: Exemplo de um arquivo de entrada contendo os parâmetros para um experimento de análise de nutrientes.

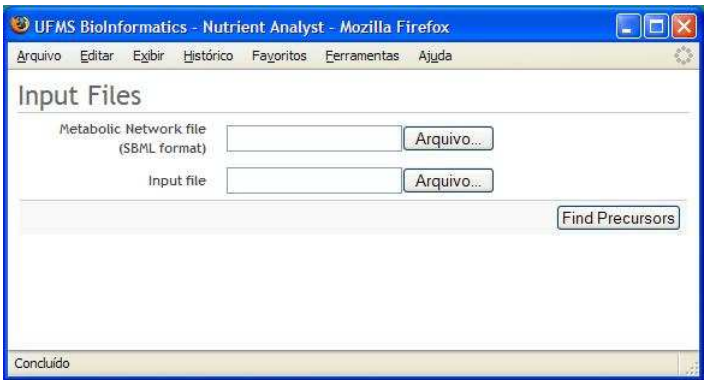


Figura A.4: Exemplo da tela de entrada de dados para a versão online da ferramenta *Nutrient-Analyst*.

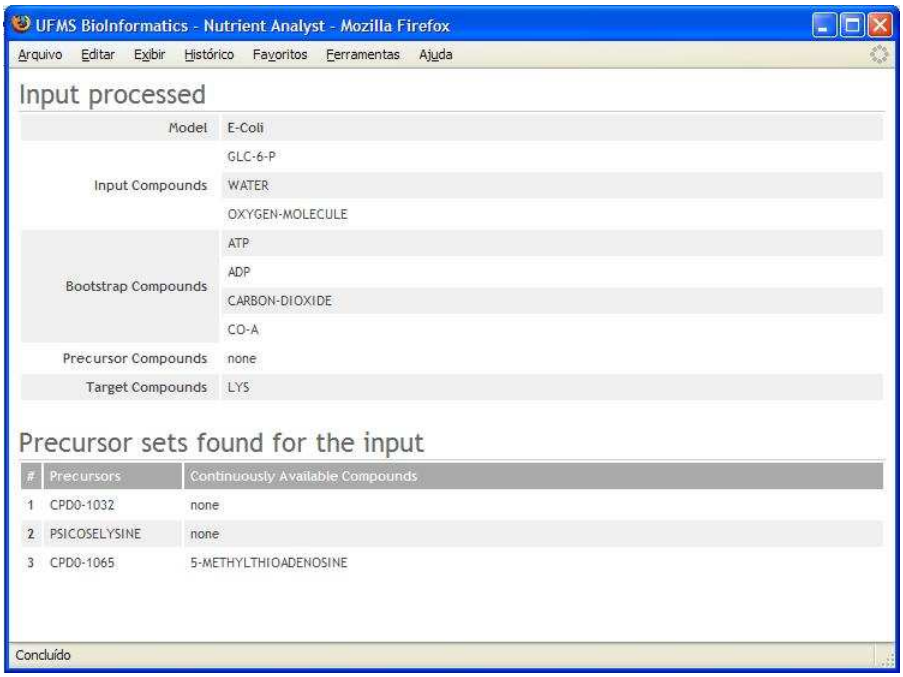


Figura A.5: Exemplo da tela de exibição de resultados da versão online da ferramenta *NutrientAnalyst*.

Apêndice B

Glossário

Bootstrap: vide composto de partida;

Composto (químico): elemento químico participante, como substrato ou produto, das reações químicas que compõem uma rede metabólica;

Composto alvo: em um experimento de análise de nutrientes, trata-se de um composto que se espera obter a partir de um conjunto de compostos iniciais, que formam o meio de crescimento para o organismo cuja rede metabólica é modelada;

Composto de partida (*Bootstrap*): os compostos de partida são aqueles que não compõem o meio de crescimento, mas ainda assim estão disponíveis, na própria composição celular do organismo;

Enzima: elemento químico associado a reações químicas, responsável por acelerar ou inibir estas reações;

Estoquiometria: Estoquiometria de uma reação são as quantidades necessárias de cada um dos reagentes desta reação e as quantidades produzidas de cada um dos produtos da reação;

Meio de crescimento: trata-se do conjunto de compostos químicos disponíveis no meio ambiente para um organismo;

Produtos (de uma reação): compostos gerados por uma reação, a partir de um determinado conjunto de compostos iniciais, chamados substratos;

Reação (química): processo através do qual um conjunto de compostos iniciais, chamados substratos, são transformados em um conjunto de compostos finais, chamados produtos;

Rede metabólica: conjunto de compostos e reações encadeadas;

Substratos: compostos que são transformados, por uma reação, em um conjunto de produtos.

Referências Bibliográficas

[1] J. Berg and M. Lassig. Cross-species analysis of biological networks by Bayesian alignment. *PNAS*, 103(29):10967–10972, 2006.

[2] BioCyc – database collection. Disponível em `http://biocyc.org`.

[3] BioSilico – an integrated metabolic database system.
Disponível em `http://biosilico.kaist.ac.kr:8017/biochemdb/index.jsp`.

[4] A. Burgard, S. Vaidyaraman, and C. Maranas. Minimal reaction sets for *Escherichia coli* metabolism under different growth requirements and uptake environments. *Biotechnology Progress*, 17(5):791–797, 2001.

[5] F. Chierichetti, V. Lacroix, A. Marchetti-Spaccamela, M.-F. Sagot, and L. Stougie. Modes and cuts in metabolic networks: Complexity and algorithms. Unpublished, 2008.

[6] L. Cottret, P. V. Milreu, V. Acuña, A. Marchetti-Spaccamela, F. Viduani Martinez, M.-F. Sagot, and L. Stougie. Enumerating precursor sets of target metabolites in a metabolic network. *Proceedings of the 8th Workshop on Algorithms in Bioinformatics (WABI 2008)*, 2008.

[7] Cytoscape.
Disponível em `http://www.cytoscape.org`.

[8] EcoCyc – encyclopedia of *Escherichia coli* K-12 genes and metabolism.
Disponível em `http://www.ecocyc.org`.

[9] EMP – enzymes and metabolic pathways. Disponível em `http://www.empproject.com`.

[10] P. Feofiloff. Exercícios de teoria dos grafos, 2005.
Disponível em `http://www.ime.usp.br/~pf/grafos-exercicios`.

[11] J. Flannick, A. Novak, B. S. Srinivasan, H. H. McAdams, and S. Batzoglou. Graemlin: General and robust alignment of multiple large interaction networks. *Genome Research*, 16(9):1169–1181, 2006.

[12] C. V. Forst, C. Flamm, I. L. Hofacker, and P. F. Stadler. Algebraic comparison of metabolic networks, phylogenetic inference, and metabolic innovation. *BMC Bioinformatics*, 7:67–78, 2006.

[13] R. Garret and C. Grisham. *Biochemistry*. Saunders College Publishing, 1999.

[14] T. Handorf, N. Christian, O. Ebenhöh, and D. Kahn. An environmental perspective on metabolism. *Journal of Theoretical Biology*, 252:530–537, 2007.

[15] T. Handorf, O. Ebenhöh, and R. Heinrich. Expanding metabolic networks: scope of compounds, robustness and evolution. *Journal of Molecular Evolution*, pages 498–512, 2005.

[16] C. M. Jonker, J. L. Snoep, J. Treur, H. V. Westerhoff, and W. C. A. Wijngaards. BDI-modelling of intracellular dynamics. In A. B. Williams and K. Decker, editors, *Proceedings of the First International Workshop on Bioinformatics and Multi-Agent Systems, BIX-MAS'02*, pages 15–23, 2002.

[17] KAAS – KEGG Automatic Annotation Server.
Disponível em `http://www.genome.jp/kegg/kaas`.

[18] KEGG – Kyoto encyclopedia of genes and genomes.
Disponível em `http://www.genome.jp/kegg/kegg2.html`.

[19] A. Kun, B. Papp, and E. Szathmáry. Computational identification of obligatory autocatalytic replicators embedded in metabolic networks. *Genome Biology*, 9:51, 2008.

[20] J. Larner. *Metabolismo Intermediário e sua Regulação*. Editora Edgard Blucher Ltda, 1974.

[21] A. Lehninger, D. Nelson, and M. Fox. *Principles of Biochemistry*. Sarvier, 4th edition, 2004.

[22] P. Lincoln and A. Tiwari. Symbolic systems biology: hybrid modeling and analysis of biological networks. *7th International Workshop on Hybrid Systems: Computation and Control, LNCS 2993*, pages 660–672, 2004.

[23] METACYC – Encyclopedia of metabolic pathways.
Disponível em `http://www.metacyc.org`.

[24] A. Nakabachi, A. Yamashita, H. Toh, H. Ishikawa, H. E. Dunbar, N. A. Moran, and M. Hattori. The 160-kilobase genome of the bacterial endosymbiont *Carsonella*. *Science*, 314(5797):267, 2006.

[25] BIOCYC PATHWAY TOOLS.
Disponível em `http://bioinformatics.ai.sri.com/ptools/`.

[26] R. Y. Pinter, O. Rokhlenko, E. Yeger-Lotem, and M. Ziv-Ukelson. Alignment of metabolic pathways. *Bioinformatics*, 21(16):3401–3408, 2005.

[27] P. Romero and P. D. Karp. Nutrient-related analysis of pathway/genome databases. In *Proceedings of 6th Pacific Symposium on Biocomputing (PSB 2001)*, pages 470–482, 2001.

[28] M.-F. Sagot. An introduction to metabolic networks and their structural analysis. Unpublished, 2008.

[29] SBML SPECIFICATION.
Disponível em `http://www.sbml.org/Special/specifications/sbml-level-2/version-1/html/sbml-level-2.html`.

[30] S. Schuster, T. Dandekar, and D. Fell. Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends in Biotechnology*, 17(2):53–60, 1999.

[31] R. Sharan and T. Ideker. Modeling cellular machinery through biological network comparison. *Nature Biotechnology*, 24(4):427–433, April 2006.

[32] J. Tamames, R. Gil, A. Latorre, F. S. Peretó, and A. Moya. The frontier between cell and organelle: genome analysis of candidatus *Carsonella ruddii*. *BMC Evolutionary Biology*, 7:181, 2007.

[33] Y. Tohsato, H. Matsuda, and A. Hashimoto. A multiple alignment algorithm for metabolic pathway analysis using enzyme hierarchy. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 376–383. AAAI Press, 2000.

[34] Tumor Metabolome. Disponível em `http://metabolic-database.com`.

[35] Z. Wunderlich and L. A. Mirny. Using the topology of metabolic networks to predict viability of mutant strains. *Biophysical Journal*, 91(6):2304–2311, 2006.