



UNIVERSIDADE FEDERAL DO PAMPA
CIÊNCIA DA COMPUTAÇÃO

DOUGLAS PAIM LAUTERT

**REDUÇÃO DE CRUZAMENTO DE ARESTAS EM DESENHOS DE
SISTEMAS BIOLÓGICOS**

Trabalho de Conclusão de Curso

Alegrete
2011

DOUGLAS PAIM LAUTERT

**REDUÇÃO DE CRUZAMENTO DE ARESTAS EM DESENHOS DE SISTEMAS
BIOLÓGICOS**

Trabalho de Conclusão de Curso
apresentado como parte das atividades para
obtenção do título de bacharel em Ciência
da Computação na Universidade Federal do
Pampa.

Orientador: Prof. Me. Marcelo Cezar Pinto

Alegrete

2011

DOUGLAS PAIM LAUTERT

**REDUÇÃO DE CRUZAMENTO DE ARESTAS EM DESENHO DE
SISTEMAS BIOLÓGICOS**

Trabalho de Conclusão de Curso apresentado
como parte das atividades para obtenção do
título de bacharel em Ciência da Computação
na Universidade Federal do Pampa.

Trabalho apresentado e aprovado em: 28 de Junho de 2011.

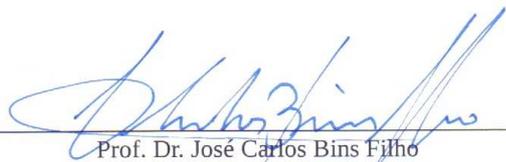
Banca Examinadora:



Prof. Msc. Marcelo Cezar Pinto
Orientador
Ciência da Computação - UNIPAMPA



Prof. Dr. Cleo Zanella Billa
Ciência da Computação - UNIPAMPA



Prof. Dr. José Carlos Bins Filho
Ciência da Computação - UNIPAMPA

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	VI	
LISTA DE FIGURAS	VII	
LISTA DE TABELAS	VIII	
LISTA DE GRÁFICOS	IX	
RESUMO	XI	
1	INTRODUÇÃO	1
1.1	Motivação	1
1.2	Objetivo geral	2
1.2.1	Objetivos específicos	2
1.3	Organização do texto	2
2	BIOLOGIA SISTÊMICA	4
2.1	Sistemas Biológicos	4
2.2	Modelos Biológicos	4
2.3	Linguagens de visualização e Descrição	5
3	DESENHO DE GRAFOS	6
3.1	Conceitos de Grafos	6
3.2	Desenho de Grafos	7
3.2.1	Fundamentos da Representação Gráfica	8
3.2.2	Paradigmas para Desenho de Grafos	10
3.2.2.1	Algoritmos direcionados por força	11
3.2.2.2	Algoritmos direcionados pelo desenho de árvores	11
3.2.2.3	Algoritmos de variações de forças direcionadas	12
3.2.2.4	Algoritmo Fruchterman e Reingold	13
4	SBML	17
4.1	Resumo da estrutura de um SBML	17
5	SBGN	21
5.1	Estrutura SBGN da Linguagem de Descrição de Processos	22
5.1.1	Grupo de Nós Entidade (GNE)	23
5.1.2	Nós Contêineres (NC)	23
5.1.3	Nós de Processo (NO)	24
5.1.4	Operadores Lógicos	25
5.1.5	Arcos Conexos	25
6	GML	27
6.1	Especialização SBML	31
6.2	Especialização SBGN	32
7	SOFTWARES PARA SISTEMAS BIOLÓGICOS	35
7.1	CellDesigner	35
7.2	EPE	36
7.3	Arcadia	36
7.4	VANTED	37
8	DESENVOLVIMENTO DA FERRAMENTA	38
8.1	Requisitos principais do problema a ser trabalhado	38
8.2	Procedimentos Gerais	40
8.2.1	Adequar SBGN	41
8.2.2	Organizar Compartimentos	43

8.2.3	Aplicar Heurística	45
8.3	Implementação da Ferramenta	45
8.3.1	Técnicas e Ferramentas utilizadas	45
8.3.2	Desenvolvimento do SBLgen	46
8.3.2.1	Algoritmo de Adequar SBGN	46
8.3.2.2	Algoritmo de Organizar os Compartimentos	47
8.3.2.3	Algoritmo de Aplicar Heurística	48
9	TESTES	51
9.1	Teste em modelo com quantidade de arestas superiores aos nós	51
9.1.1	Média de redução de cruzamento pela temperatura	52
9.1.2	Redução média de cruzamento em relação à configuração inicial	53
9.2	Teste em modelo com números de nós e arestas similares	54
9.2.1	Média de redução de cruzamento pela temperatura	56
9.2.2	Redução média de cruzamento em relação à configuração inicial	56
10	DISCUSSÃO	58
11	REFERÊNCIAS BIBLIOGRÁFICAS	60

LISTA DE ABREVIATURAS E SIGLAS

SBGN – Systems Biology Graphical Notation

SBML – Systems Biology Markup Language

GNE – Grupo Nós Entidade

NC – Nós Contêineres

NP – Nós Processos

GML – Graph Modelling Language

EPE – Edinburgh Pathway Editor

VANTED – Visualization and Analysis of Networks containing Experimental Data

SBLGen – Systems Biology Layout Generator

XML – eXtensive Markup Language

SBGN-DP – SBGN Descrição de Processos

SBGN-ER – SBGN Entidade Relacionamento

SBGN-AF – SBGN Atividade de Fluxo

LISTA DE FIGURAS

FIGURA 3.1 – Convenções para Representação Gráfica	9
FIGURA 3.2 – Diagrama utilizando algoritmo direcionado por força	11
FIGURA 3.3 – Esquema de uma Força Dirigida para Árvores	13
FIGURA 3.4 – Pseudocódigo do algoritmo de FR	15
FIGURA 4.1 – Lista de elementos componentes de um modelo SBML	18
FIGURA 5.1 – Exemplos de glifos do grupo GNE	23
FIGURA 5.2 – Exemplos de glifos do grupo NC	24
FIGURA 5.3 – Exemplos de glifos do grupo Nós Processo	24
FIGURA 5.4 – Exemplos de glifos do grupo Operadores Lógicos	25
FIGURA 5.5 – Exemplos de glifos do grupo Arcos Conexos	25
FIGURA 6.1 – Representa um esquema de um GML.	28
FIGURA 6.2 – Apresentação de um nós Espécie	29
FIGURA 6.3 – Figura com informações de uma aresta	30
FIGURA 6.4 – Apresentação de um nós Espécie	31
FIGURA 6.5 – Exemplo de nó GNE Química Simples	32
FIGURA 6.6 – Aresta GNE Consumo	33
FIGURA 7.1 – Imagem do CellDesigner	35
FIGURA 7.2 – Imagem do EDE (Edinburgh Pathway Editor)	36
FIGURA 7.3 – Imagem do Arcadia em funcionamento.	36
FIGURA 7.4 – Imagem do programa VANTED	37
FIGURA 8.1 – Padrão de representação de desenho de grafo de sistemas biológicos	39
FIGURA 8.2 – Procedimento Geral da ferramenta	40
FIGURA 8.3 – Representação de um Sistema Biológico	41
FIGURA 8.4 – Representação GML de Sistema Biológico antes de Adequar SBGN	42
FIGURA 8.5- Representação do GML após Adequar SBGN	42
FIGURA 8.6 – Dimensionamento e Posicionamento dos Compartimentos	44
FIGURA 8.7 – Hierarquizar e Inserção Interna no frame <i>Parent</i>	44
FIGURA 8.8– Aplicação do Algoritmo FR Modificado	45
FIGURA 8.9 – Chamadas de Procedimentos do AdequarSBGN	46
FIGURA 8.10 – Procedimento que organiza os compartimentos	47
FIGURA 8.11 – Algoritmo FR modificado	49

LISTA DE TABELAS

Tabela 5.1 – Representação do vocábulo dos conceitos Glifos utilizado no SBGN	22
Tabela 9.1 – Lista de informações gerais de dois diferentes modelos	53
Tabela 9.2 – Lista de informações específica	53
Tabela 9.3 – Lista de informações gerais de três diferentes modelos	56
Tabela 9.4 – Lista de informações específica	56

LISTA DE GRÁFICOS

Gráfico 1 – Gráfico de barras de Redução de Cruzamento Médio ao longo do Esfriamento da Temperatura para os Modelos 18 e 144	52
Gráfico 2 – Redução média de cruzamento dada a condição inicial em relação a redução propriamente dita	53
Gráfico 3 – Gráfico de barras de Redução de Cruzamento Médio ao longo do Esfriamento da Temperatura para os Modelos 202, 3 e 4	56
Gráfico 4 – Redução média de cruzamento dada a condição inicial em relação à redução propriamente dita	57

RESUMO

O trabalho trata da aplicação do algoritmo *Fruchterman e Reingold* da linha de pesquisa de Desenho de Grafo para melhoria de visualizações de dados biológicos advindos de pesquisas realizadas na linha de Biologia Sistêmica. Desenho de Grafo utiliza conceitos de grafos para geração de novos layouts. Enquanto que Biologia Sistêmica estuda sistemas biológicos e as suas interações. A aplicação de Desenho de Grafos em Biologia Sistêmica é uma integração de áreas distintas e serve para que o conhecimento científico se torne mais facilmente compreendido entre os pesquisadores de Biologia Sistêmica, através de diagramas SBGN que tem como principal critério estético a redução de cruzamento de arestas. Assim a ferramenta SBLgen criada, através de modelos biológicos em SBML, cria *layouts* de forma clara e intuitiva para dar suporte aos pesquisadores de Biologia Sistêmica no que diz respeito à Visualização Científica de seus dados.

Palavras-chave: Biologia Sistêmica, Desenho de Grafos, Sistemas Biológicos, Modelos Biológicos, SBML e SBGN.

ABSTRACT

This work applies Fruchterman & Reingold Graph Design area algorithm to improve biological data visualization from Systems Biology research. Graph Design uses Graph concepts to generate new layouts and Systems Biology deals with biological systems and their interactions. The integration of Graph Design and Systems Biology guarantees that scientific knowledge could be easily understood to researchers. The main aesthetic criterion for SBGN diagrams adopted in this work is edge crossing minimization. So, the devised SBLGen tool, acting after VANTED software conversion of SBML models to GML files, creates intuitive and clear layouts to support Systems Biology researchers to visualize its data.

Keywords: Systems Biology, Graph Drawing, Biological Systems, Biological Models, SBML and SBGN.

1 INTRODUÇÃO

Desenho de Grafo é uma linha de pesquisa que utiliza conceitos de grafos para geração de novos layouts para visualização de desenhos. (NISHIZERKI et.al. ,2004) O presente trabalho trata justamente da aplicação de algoritmos de Desenho de Grafo para a melhoria de visualizações de dados biológicos de pesquisas realizadas na linha de Biologia Sistêmica.

Segundo Klipp et al. (2007) Biologia Sistêmica é o estudo das interações entre os componentes de um sistema biológico e como essas interações fazem emergir função e comportamento no sistema. Através da criação de padrões de descrição e modelagem de sistemas, os pesquisadores da área podem descrever, compartilhar e visualizar sistemas biológicos.

Os profissionais da área da biologia estão tão familiarizados com as notações como engenheiros eletricitas estão familiarizados com a notação de esquemas de circuitos. Assim, podem economizar tempo com elas, redirecionando esforços para seus projetos e trabalhos. De mesma forma, na Biologia Sistêmica, usuários de linguagens de descrição utilizam das notações para geração de layout gráfico de sistemas biológicos para obtenção de desenhos gráficos através de linguagens visuais como a SBGN (Systems Biology Graphical Notation). (LE NOVÈRE et al. ,2008)

Assim, o trabalho objetiva a criação de uma ferramenta para melhoria da visualização dos layouts, a partir do pressuposto de que a redução de cruzamentos entre arestas nas representações gráficas dos modelos biológicos torna o desenho mais legível.

1.1 Motivação

Os pesquisadores, da linha de pesquisa Biologia Sistêmica, utilizam de variadas ferramentas de visualização de desenhos gráficos. No entanto, os layouts não são totalmente claros ou de fácil compreensão por não utilizarem critérios estéticos e de restrição de agrupamento de dados biológicos. Portanto, uma visualização de fácil compreensão permitiria aos pesquisadores da área identificar o funcionamento dos sistemas biológicos documentados de forma mais intuitiva e eficiente.

1.2 Objetivo Geral

O objetivo é gerar um layout dos modelos biológicos de forma clara e que possa ser compartilhado. Para isso foram utilizados conceitos de Desenho de Grafos e a linguagem SBGN.

Os modelos biológicos estão disponíveis no formato SBML (ver Capítulo 4), os quais podem ser convertidos para GML (ver Capítulo 6) pelo programa VANTED (Visualization and Analysis of Networks containing Experimental Data)¹. GML é uma notação para representação de grafos, a qual é útil para a conversão para SBGN.

1.2.1 Objetivos Específicos

Como objetivos específicos estão:

1. A criação de uma ferramenta que manipule a saída do VANTED para gerar um arquivo SBGN com um desenho melhor do sistema biológico em estudo;
2. O uso de heurística de Desenho de Grafos para redução do número de cruzamento de arestas em representações gráficas de sistemas biológicos;
3. O uso da informação de compartimentos biológicos provenientes dos modelos SBML para manter a coerência espacial entre os diversos nós dos sistemas.

O presente trabalho reuniu conceitos, ferramentas e algoritmos de diversas áreas, a saber: Desenho de Grafos, Inteligência Artificial e Biologia Sistêmica. De acordo com o conhecimento dos autores, este é o primeiro trabalho que se utiliza da heurística FR (ver subseção 3.2.2.4) para melhorar o desenho de sistemas biológicos, permitindo que este desenho seja armazenado no formato SBGN.

1.3 Organização do Texto

Este texto está organizado como segue.

¹ Programa que suporta SBGN, para diversas edições, validações e traduções de linguagens para biologia sistêmica gerando mapas e diagramas SBGN sempre que necessário para diferentes modelos biológicos.

O capítulo 2 apresenta conceitos básicos sobre Biologia Sistêmica.

O capítulo 3 composto por um pequeno resumo sobre Desenho de Grafo.

O capítulo 4 cita elementos estruturais e descreve funções, características e aplicações da linguagem SBML.

O capítulo 5 cita as linguagens SBGN, além de descrever as estruturas que compõem a linguagem SBGN-DP.

O capítulo 6 sobre o conceito de uma estrutura para representar grafo, chamada GML e suas especializações.

O capítulo 7 exemplifica alguns softwares pra sistemas biológicos.

O capítulo 8 é sobre o Desenvolvimento da Ferramenta SBLgen.

O capítulo 9 lista os testes sobre a ferramenta implementada SBLgen.

E, por fim, no capítulo 10, a discussão final deste trabalho de conclusão.

2 BIOLOGIA SISTÊMICA

A Biologia Sistêmica, segundo Robeva (2010), é uma linha de pesquisa de metodologia interdisciplinar que tem por objetivo o esforço em compreender a dinâmica da vida, ao determinar como os componentes individuais de um sistema vivo interagem no espaço e no tempo para formar redes de sistemas funcionais. Biologia Sistêmica, na visão segundo Kitano (2002), nela há uma intensa e diversificada investigação sobre tópicos relacionados à Bioinformática que tem no controle a criação e modelagem computacional de sistemas biológicos, principalmente relacionados a sistemas biológicos para representatividade de regulação gênica, redes metabólicas, redes bioquímicas, que são comportamentos de sistemas vivos, respectivamente relacionados à, genética, metabolismo e redes de sequência química

Novos desafios de conhecimento de todos os níveis de organização de sistemas biológicos exigem aos pesquisadores em Biologia Sistêmica novos conhecimentos a cerca da vida. Os profissionais da área da SB oferecem, de suas pesquisas de cunho biológico, os mais baixos aos mais altos níveis de detalhamento, o que equivale tanto aos saberes sub-celulares (baixo) quanto aos de níveis celulares, de tecidos, órgãos, ou organização comportamento (alto). Estas estruturas de baixo nível, como as redes bioquímicas, muitas vezes são compostas por uma gama de unidades de outros subsistemas de redes viva, portanto, facilmente expansível o conhecimento, assim, a Biologia Sistêmica oferece um intensa investigação em sistemas biológicos tanto de nível mais baixo ao mais alto (ROBEVA, 2010)

2.1 Sistemas Biológicos

Na biologia, um sistema biológico ou sistema orgânico é um grupo de funções que juntas executam alguma determinada tarefa. Segundo Kitano (2001) alguns sistemas vivos oferecem o estudo em alto nível como no funcionamento de órgãos humanos, por exemplo, quanto em baixo nível como a criação de energia física para um órgão funcionar, por exemplo também.

2.2 Modelos Biológicos

Modelos biológicos são os resultados das modelagens computacionais documentadas de simulações dos sistemas biológicos pesquisados por profissionais da área

da computação/biologia. Os modelos oferecem um padrão de documentação para o compartilhamento de informações entre pesquisadores da área da SB. Os documentos resultantes são frutos de simulações e devem servir para troca ou partilha dos resultados das pesquisas. Um exemplo de padrão de modelagem, é a documentação através da linguagem de descrição SBML, que permite aos pesquisadores intercomunicarem-se através de documentos SBML os seus estudos com maior facilidade e precisão científica (ALON, 2007).

2.3 Linguagens de Visualização e Descrição

Linguagens de visualização que descrevem modelos biológicos que são representados no formato de visual e, linguagem de descrição, na forma de notações matemáticas. Exemplo de linguagem para construção visual é o padrão SBGN (*Systems Biology Graphical Notation*) que é uma convenção para representar sistemas biológicos em desenhos. Os exemplos de linguagem de descrição são as linguagens SBML (*Systems Biology Markup Language*), CellML (*Cell Markup Language*), CML (*Chemical Markup language*), dentre outras. (ALON, 2007)

3 DESENHO DE GRAFOS

Neste capítulo são apresentados os fundamentos de duas linhas de pesquisa relacionadas ao presente trabalho como os tópicos de pesquisa em Teoria de Grafos e Desenho de Grafos .

3.1 Conceitos de Grafos

Grafos são modelagens de cunho matemático que permitem relacionar elementos de dados, como entidades únicas. Essas entidades são identificadas como vértices e a relação entre elas como arestas. Abaixo está uma definição de grafo dado segundo West (1996).

Definição: Um grafo G com n vértices e m arestas consiste de um conjunto de vértices $V(G) = \{v_1, \dots, v_n\}$ e um conjunto de arestas $E(G) = \{e_1, \dots, e_m\}$ onde cada aresta possui dois vértices chamados seus extremos. Será denotado de uv para uma aresta $e = \{u, v\}$. Se o par uv pertencerem $E(G)$, então u e v são adjacentes. Os grafos podem ser classificados pelo seu tamanho, estrutura e tipo de arestas. De acordo com seu tamanho os grafos podem ser (MARSHALL, 2001):

- **Pequeno** - grafos até 100 elementos (vértices e arestas);
- **Médio** - grafos com até 1000 elementos (vértices e arestas);
- **Grande** - grafos com até 10.000 elementos (vértices e arestas);
- **Esparso** - grafos com mais de 10.000 elementos (vértices e arestas).

Os conceitos a seguir foram retiradas de segundo West (1996) e têm relação com a topologia de um grafo, sendo importantes no restante deste texto.

Um grafo é dito finito se o seu conjunto de arestas e vértices é finito. Um *loop* é uma aresta cujos vértices extremos são iguais. Seja uv uma aresta de um grafo G , essa aresta é dita adjacente a u e adjacente a v , também se diz que uv é incidente em u e v .

O grau de um vértice v , escrito $d(v)$, é o número de arestas (somente aquelas que não são loops) adjacentes a v . Arestas paralelas são arestas que têm o mesmo par de vértices adjacentes. Um grafo é dito simples se não possui loops nem arestas paralelas.

Grafo direcionado é aquele onde a aresta é um par ordenado de vértices, ou seja, cada aresta possui uma direção. A aresta uv pertence a $E(G)$ de um grafo direcionado G sai do vértice u e vai para o vértice v . O vértice u é chamado origem e v destino.

A escolha da origem e do destino determina a direção da aresta, assim uv pertence a $E(G)$ é diferente da aresta vu pertence a $E(G)$.

A distância entre dois vértices em um grafo, também chamado distância teórica, é o tamanho do caminho mais curto entre estes vértices.

Um ciclo é quando ao percorrer um grafo pelos cruzamentos de algumas arestas sem voltar à aresta anterior, tendo início em vu e encontre da aresta wv onde w pertence $V(w)$ qualquer e que encontre novamente v .

Esta pesquisa trabalha com grafos finitos simples conexos de tamanho pequeno, que serão identificados a partir de agora somente por grafo direcionado.

3.2 Desenho de grafos

Um Desenho de Grafo para um grafo G é um desenho de G no plano, tal que os vértices são desenhados em lugares distintos onde nenhuma aresta entre si se cruza e as arestas adjacentes também não e duas arestas, caso se cruzem, no máximo uma única vez e no seu interior, nunca nos extremos. Arestas não interceptam vértices, exceto quando incidentes nestes vértices, portanto cruzamento somente pode ser compartilhado por duas arestas. (NISHIZEKI et.al., 2004)

Definição

Desenho de grafo ou *Graph Layout*, um ramo da teoria dos grafos, aplica-se a topologia e geometria para derivar representações bidimensionais de gráficos. Um Desenho de Grafo é basicamente uma representação simples gráfica (geralmente, com cruzamentos em poucas arestas sendo permitidos) geralmente destinada a uma visualização conveniente de certas propriedades de geradores gráficos desenvolvidos. Sendo motivado pelas aplicações como projetos de circuitos lógicos, análise de redes sociais, cartografia e bioinformática, onde muitas das aplicações fazem uso de visualização da informação de dados. (WIKIPEDIA, 2010)

3.2.1 Fundamentos da Representação gráfica

Não existe uma forma objetiva para classificar o melhor jeito de se programar e gerar uma representação visual. Para se considerar um Desenho de Grafo ótimo em algum determinado contexto de aplicação visual, algumas premissas são necessários, como a especificação de parâmetros bem definidos que descrevam a qualidade da representação gráfica. Os principais parâmetros encontrados na literatura são as convenções, restrições e critérios estéticos. Os próximos capítulos abordam, separadamente, cada um deles. (NISHIZEKI et.al., 2004)

Convenções

Uma convenção de representação visual ou gráfica corresponde a uma determinação básica que deve ser satisfeita para um *layout* gráfico adequado. Em determinado tipo de diagrama, por exemplo, pode ser necessário que todas as setas (arestas direcionadas) sejam desenhadas da esquerda para direita. Uma convenção para uma representação gráfica de um aplicativo real pode ser extremamente complexa e envolver vários detalhes para ela. (WIKIPEDIA, 2010)

Algumas convenções bastante conhecidas são ilustradas na Figura 3.1 e descritas como:

- **Poligonal** - Cada aresta do grafo é desenhada como uma cadeia poligonal;
- **Linhas-retas** - Cada aresta do grafo é desenhada como segmentos retos;
- **Ortogonal** - Cada aresta do grafo é desenhada como uma cadeia poligonal de segmentos horizontais e verticais;
- **Grid** - Vértices, cruzamentos de arestas e dobras de arestas possuem coordenadas inteiras;
- **Planar** - Não há cruzamento de arestas.

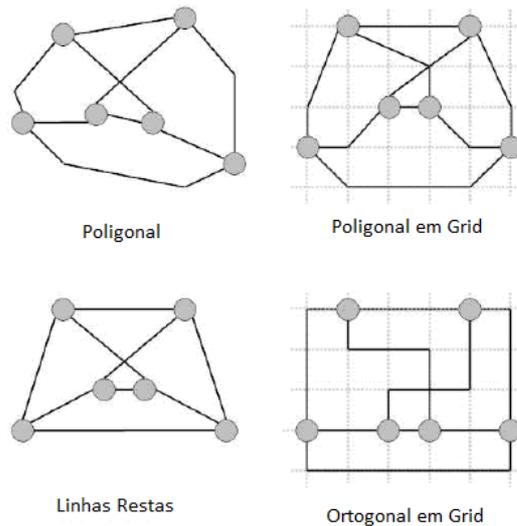


FIGURA 3.1 – Convenções para Representação Gráfica

As convenções linhas-retas e ortogonal são especializações da convenção poligonal, então conforme três casos as arestas são series poligonais. Assim sendo, um Desenho de Grafo pode adotar mais de uma convenção em uma imagem, como a serie poligonal. Os critérios estéticos especificam propriedades de *layout* da representação gráfica que melhoram esteticamente desenho e que devem ser aplicadas ao máximo possível.

Critérios Estéticos

- **Cruzamentos de Arestas** - Os cruzamentos de arestas deixam ininteligível um visual. Desenho Planar é sempre desejado, porém quando não é possível, deve-se minimizar os cruzamentos entre as arestas.
- **Área** - Quando a convenção grid é adotada, as representações gráficas não podem ser reduzidas ilimitadamente. Pois algumas aplicações práticas podem conter grafos grandes os quais precisam ser exibidos inteiramente na representação. Portanto, a área de visualização na tela do computador deve-se ser limitada e expandida pelo usuário.
- **Comprimento Total de Arestas** - Assim como no caso da área, deve-se tentar minimizar a soma total dos comprimentos das arestas quando a convenção grid é adotada.

- **Dobras em Arestas** - Para convenções ortogonais, é importante que o número de dobras por aresta, assim como o número total de dobras, seja minimizado. Com muitas dobras fica difícil de entender o desenho do grafo.
- **Resolução Angular** - Em um Desenho de Grafo que adota a convenção linhas-retas, a maximização do menor ângulo entre duas arestas incidente a um vértice é especialmente relevante. Se o grafo possui vértices com grau elevado e esse critério não for levado em conta, o grafo na tela pode ficar pouco inteligível;
- **Simetria** - A ênfase das simetrias de um grafo na tela também pode ser importante em alguns contextos. Existem modelos matemáticos que definem precisamente o que são simetrias em grafos.

Restrições

Restrições garantem certas propriedades que Enquanto que as convenções e os critérios estéticos estão associados à representação visual como um todo, as restrições se aplicam apenas as coordenadas do grafo (Cruz, 2010) . As restrições usualmente utilizadas são listadas abaixo:

- **Centralização** – O local na tela dado vértice no centro da representação visual;
- **Externalização** – O local na tela dado vértice no limite exterior do desenho;
- **Agrupamento** – O posicionamento de conjunto de vértices próximos uns dos outros;

3.2.2 Paradigmas para Desenho de Grafos

Um algoritmo para Desenho de Grafo levam em consideração as fundamentações gráficas, vista na seção 3.2.1. E são classificados de acordo com alguns paradigmas usados para montar o desenho. Cada paradigma das seções seguintes, objetivam satisfazer uma ou mais fundamentações gráficas. Existem vários paradigmas para algoritmos de desenho de grafos, abaixo são listadas segundo Battista (1998).

3.2.2.1 Algoritmos direcionados por força

Algoritmos com força de base dirigida é uma classe de algoritmos para desenhar grafos de uma maneira agradável esteticamente. Sua finalidade é posicionar os nós de um gráfico em duas dimensões ou três dimensões no espaço da tela ou desenho para que todas as arestas tenham comprimento igual. Os componentes do desenho são posicionados em um estado inicial e as arestas são abstrações de molas, que retraem ou expandem com energia imposta as molas. Quando energia estabiliza, os valores de tamanho das arestas ficam equivalentes e o desenho fica simétrico.

Segundo Aiello e Silveira (2004) forças expansão ou repulsão são calculadas entres todos os pares de componentes, mas essas forças são somente calculadas entre componentes vizinhos.

A Figura 3.2 apresenta o resultado obtido quando este algoritmo é usado para desenhar o diagrama de um sistema de distribuição.

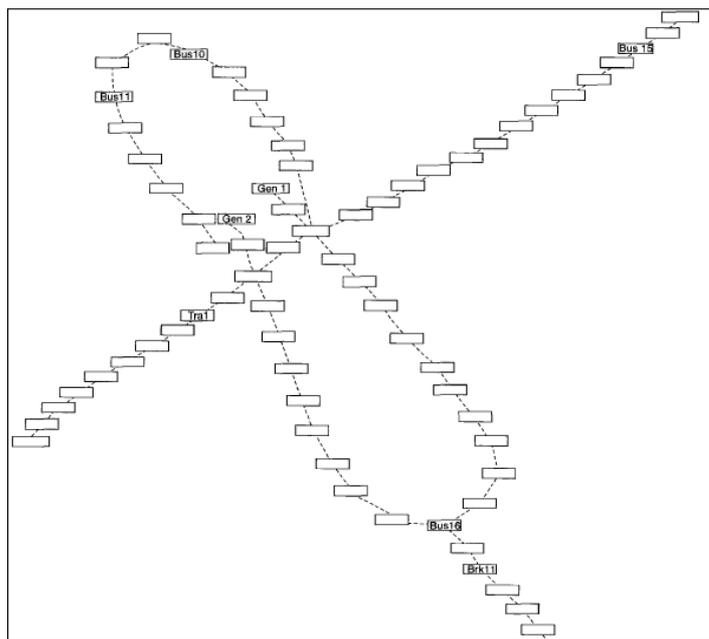


FIGURA 3.2 – Diagrama utilizando algoritmo direcionado por força.

3.2.2.2 Algoritmos direcionados pelo desenho de árvores

Layout de Árvore é um conjunto de algoritmos que apresentam um vértice pai como o vértice que forma uma hierarquia parecida como uma árvore. Este algoritmo é adequada para grafos no formato de árvores. Os filhos de cada vértice da árvore são

desenhados em círculo ao redor do pai para que estes não se sobreponham. (BATTISTA, 1998)

A Figura 3.3 abaixo apresenta o resultado obtido quando este algoritmo é usado para desenhar o diagrama uma árvore de um grafo qualquer.

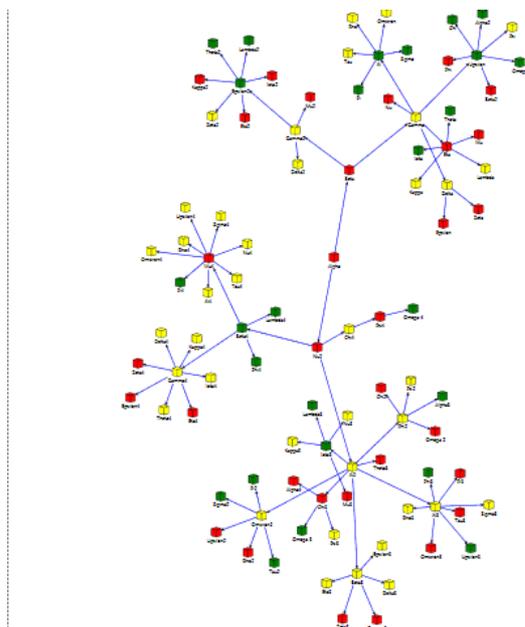


FIGURA 3.3 – Esquema de uma Força Dirigida para Árvores

3.2.2.3 Algoritmos de Variações de Forças direcionadas

As variações de força direcionadas são métodos mais simples e populares de força dirigida que usam forças das *molas* e energias aplicadas nelas.

Os algoritmos que utilizam esta forma de forças são chamados de algoritmos de *Spring*. Os vértices do grafo resultante são vistos como *Springs* que são considerados como *partículas* de energia. Neste método, a força de um vértice v é definida da seguinte:

$$F(v) = \sum f_a(d_{uv}) * \left(\frac{u-v}{d_{uv}}\right) - \sum_{(uv) \in V^2} f_r(d_{uv}) * \left(\frac{u-v}{d_{uv}}\right)$$

Onde d_{uv} é a distância entre u e v e f_a é a fator força atrativa e f_r é o fator de repulsão. Como pode ser visto da seguinte maneira:

$$f_a(d_{uv}) = k_{uv}^{(1)}(d_{uv} - l_{uv}) \text{ e } f_r(d_{uv}) = \frac{k_{uv}^{(2)}}{d_{uv}^2} .$$

Onde l_{uv} é o comprimento natural da mola entre os vértices u e v , $K_{uv}^{(1)}$ é a capacidade não distanciamento das molas no *Spring* entre u e v , e $K_{uv}^{(2)}$ indica a intensidade da força de repulsão entre u e v .

A força geral nas molas está na diferença entre as forças aplicadas entre u e v dada a distância entre esses dois vértices. As forças de repulsão prevêm aos vértices do grafo quando colocados muito próximos um ao outro. Quando longe, elas se atraem como se as molas expandissem.

Os exemplos de layouts abaixo são *Springs*:

- **Kamada e Kawai (Kamada e Kawai, 1989)**: baseado no Spring Embedder, esse método trabalha com um sistema de energia buscando um estado de equilíbrio. Para cada par de vértices adjacentes, o tamanho da mola (aresta) relaxada é proporcional ao caminho mais curto no grafo entre eles. Focaliza a minimização dos cruzamentos, simetria do desenho e uniformização do tamanho das arestas;
- **Fruchterman e Reingold (Fruchterman e Reingold, 1991)**: trabalha com um sistema de atração eletrostática em que a força de atração entre os vértices conectados é balanceada pela força de repulsão entre todos os vértices. A determinação do equilíbrio é baseado em outra heurística chamada simulated annealing.

3.2.2.4 Algoritmo Fruchterman e Reingold (FR)

O algoritmo de Fruchterman-Reingold segundo (Fruchterman e Reingold, 1991) adicionou uma distribuição de distância entre pares de vértices igualitária que respeitam dois critérios estéticos, em que todas as arestas tem tamanhos parecidos e a minimização do cruzamentos .

O FR então trata vértices do grafo como se fossem unidades de partículas atômicas ou corpos celestes, exercendo forças de atração e repulsão entre si .

As forças atrativas e repulsivas são definidas por respectivamente:

$$f_a(d) = d^2/k \text{ e } f_r(d) = -k^2/d$$

Em termos de distância d entre dois vértices e a distância otimizada entre os vértices k define-se como:

$$k = \sqrt{(W * L) / |V|}.$$

As variáveis W e L , correspondem, respectivamente, a largura e altura do lugar onde acontecerá a repulsão e atração. E V é o vetor que contém os vértices de um grafo. Na equação que retorna a k , o módulo de V significa o número de vértices contém a lista de vértices do grafo ou grupo de elementos vértices. O resultado feito da divisão então é aplicado na raiz quadrada e dá o resultado de k que é um fator aplicado nas funções f_r e f_a .

O algoritmo de FR acrescenta a noções de temperatura que podem ser usadas pondo um valor inicial, por exemplo, um décimo da largura do *frame* ou 10% do tamanho de um espaço na tela, e decaindo a zero em uma forma linear inversa.

A temperatura controla o deslocamento de vértices de modo que o layout melhora, por exemplo, quando o número de cruzamentos cai drasticamente, assim seriam ajustado o grafo. O pseudocódigo para o FR fornece mais informações sobre o funcionamento desse algoritmo no Figura 3.4 a seguir:

Algoritmo de FR (V,E,W,L)

```

01 area = W * L; {W e L são largura e altura do frame }
02 G = (V,E) ; {os vértices são colocados aleatoriamente nas posições}
03 k = sqrt(area/|V|);;
04 função fa(x) = início return x2/k fim;
05 função fr(x) = início return k2/x fim;
06 função cool(t) = início return t*0.1 fim;
07 função dif(p1, p2) = início p3.x=p1.x-p2.x; p3.y=p1.y-p2.y; return p3 fim;
08 t = 10000;
09 para i =1 até iterações faça início
10     para cada v em V faça início
11         v.d.x = 0; v.d.y = 0;
12         para cada u em V faça início
13             se (u != v) então início
14                 delta = dif(u.pos, v.pos);
15                 v.d += (delta/|delta|)*fr(|delta|)
16             fim
17         fim
18     fim
19     para cada u,v em E faça início
20         {cada aresta num par ordenado de vértices .v e .u}
21         delta = dif (v.pos, u.pos);
22         v.d -= (delta/|delta|)*fa(|delta|);
23         u.d += (delta/|delta|)*fa(|delta|);
24     fim
25 {limita Maximo de deslocamento pela temperatura t e previne o deslocamento fora do frame}
26     para cada v em V faça início
27         v.pos.x += (v.d.x/|v.d| )*min(v.d.x, t); v.pos.y +=(v.dy/|v.d|)*min(v.d.x, t);
28         v.pos.x = min(W/2, Max(-W/2, v.pos.x));
29         v.pos.y = min(L/2, Max(-L/2, v.pos.y));
30     fim
31     t = cool(t);
32 fim

```

FIGURA 3.4 – Pseudocódigo do algoritmo de FR

No Figura 3.4 acima, as variáveis W , L , V , E, G indicam, respectivamente, largura, altura, lista de vértices, lista de arestas e o grafo que contém V e E .

Enquanto que d e pos indicam, consecutivamente, os vetores (x,y) de deslocamento de um vértice dentro de um frame e posição atual (x,y) de um vértice. Enquanto que $delta$ é o vetor diferença entre a posição (x,y) de dois vértices distintos. As

variáveis com d são referências de vetores posição que guardam o deslocamento para cada vértice que as referenciam.

A variável k é uma variável global que recebe os parâmetros do tamanho da área do frame, e é uma constante que é usada nas funções de atração e repulsão, e serve para delimitar o tamanho da repulsão ou atração.

Em relação funcionamento do algoritmo de FR tem três partes que levam em conta uma quantidade limitada de iterações (linha 09). A primeira parte do funcionamento do algoritmo, leva em conta o '*para*' localizado na linha 10, significando que para lista de vértices são calculadas as forças de repulsão entre os pares diferentes, ou seja, ele busca um vértice e aplica a força para os outros, e assim por diante, guardando num vetor posição dentro d para cada vértice.

Em relação ao segundo passo, do para da linha 19, são calculadas as forças de atração entre vértices, que possuem aresta entre si. O vértice da linha 22, de origem v , desloca-se negativamente em direção ao vértice destino u , que por sua vez, se desloca positivamente em direção ao vértice origem, assim, eles estão se atraindo.

A terceira e última parte, todos os vértices são movimentados e as suas posições dos vértices ficam limitadas ao tamanho da área do frame, com a altura e largura dele L e W (linhas 28 e 29). Além disso, há uma função de resfriamento (função cool), que tem temperatura inicial dez mil (10000) e, reduz para dez por cento (10%) de seu valor a cada iteração. A redução acelerada faz diminuir a atração e repulsão entre os vértices, linha 31.

Ao final do processo da contagem das iterações haverá um desenho simétrico dentro de frame delimitado por uma área com base em L e W , altura e largura.

4 SBML

SBML (Linguagem de Marcação para Biologia Sistêmica) foi um padrão criado no ano dois mil, baseado na estrutura da linguagem XML (Linguagem de Marcação Extensiva). Atualmente a SBML é desenvolvida de forma cooperativa por uma comunidade científica com a ajuda de colaboradores de fóruns de discussões especializados no assunto onde colocam-se em pauta novas aplicações e correções aos recursos da biblioteca open-source chamada *LibSBML*. Tão logo são implementados novos recursos na biblioteca novas versões (*levels*) da linguagem são criadas. (HUCKA et. al., 2003; BOMSTEIN et. al., 2008; HUCKA et. al., 2004)

Mesmo com as novas versões dessa linguagem, as suas características são mantidas como: foco na simulação, armazenamento de dados biológicos e a fácil legibilidade para máquinas, criando um padrão de linguagem que garante formatos de arquivos que possibilitam modelagens sobreviverem além da vida útil do software. (HUCKA et.al. 2003; LE NOVÈRE, 2006)

Os focos ou objetivos finais para a SBML são de ser uma linguagem descritiva de representação para modelos biológicos computacionais com objetivo de servir como uma linguagem franca para software que permitem codificar modelos que possam ser trocados e interpretados por diferentes softwares. (HUCKA et. al., 2004)

O objetivo geral da SBML não é para ser destinada a fornecer detalhes de implementação de algoritmos para programas, nem procedimentos utilizados para processar e analisar os modelos, mas para instanciar as construções em formato padrão. (BOMSTEIN et. al., 2008) Além disso, a linguagem SBML não está ligada a qualquer sistema de software em específico, podendo ser tratada por várias ferramentas como de visualização de dados ou de modelagem. (ALVES et. al., 2006)

A estrutura XML da linguagem SBML promove a fácil distribuição de dados biológicos em um formato padrão, que é uma entrega de uma construção matemática que se torna um quadro semanticamente enriquecido para comunicação do conhecimento que sobrevive além do tempo útil do software SBML. (SBML.org, 2009; HUCKA et.al. 2004; LE NOVÈRE, 2006).

4.1 Resumo da estrutura de um SBML

Uma estrutura SBML é baseada em entidades bioquímicas, e uma dessas, é reação química. Essa entidade é a transformação química de elementos em novos

elementos. A reação pode ser dividida em um número restrito de elementos conceituais de entidades como as espécies reagentes, espécies de produtos, espécies modificadoras, reações, compartimentos, eventos.

Além da entidade reação, elementos matemáticos como as quantidades de diferentes tipos de grandezas, parâmetros, leis da taxa para quantidade, funções, regras, restrições e assinaturas iniciais, são atribuídas no documento. Nas próximas seções serão vistas com mais detalhes implementação e conceito.

O Figura 4.1 abaixo mostra o esquema de listas de elementos de um documento SBML.

-
1. Definição inicial de um modelo
 2. Lista de definições de funções (opcional)
 3. Lista de unidade de definição de Grandeza (opcional)
 4. Lista de tipos de compartimentos (opcional)
 5. Lista de tipos de espécies (opcional)
 6. Lista de compartimentos (opcional)
 7. Lista de espécies (opcional)
 8. Lista de parâmetros (opcional)
 9. Lista de assinaturas iniciais (opcional)
 10. Lista de regras (opcional)
 11. Lista de restrições (opcional)
 12. Lista de reações (opcional)
 13. Lista de eventos (opcional)
 14. Definição final de um modelo
-

FIGURA 4.1 – Lista de elementos componentes de um modelo SBML.

As definições de listas de elementos na Figura 4.1 correspondem à baixo ao nome:

- **Lista de Definições de funções:** Uma lista de funções matemáticas que podem ser utilizadas em todo o resto do modelo. (HUCKA et. al., 2008)

- **Lista de Unidade Definições de Grandeza:** Uma lista de definições de grandezas do nome de uma nova unidade de grandeza de medida, ou uma redefinição de uma unidade padrão existente SBML. Ex.: Grama ou unidade em miligrama. Utilizados em outra parte do modelo, como no valor da grandeza que determina o volume de um elemento. (HUCKA et. al., 2008)
- **Lista de Tipos de Compartimentos:** Uma lista de tipos diferentes de locais (compartimentos) onde as entidades (espécies, objetos ou processos) como as substâncias químicas podem ser localizados. (HUCKA et. al., 2008)
- **Lista de Tipo de Espécies:** Lista de tipos diferentes de entidades (objetos) que podem participar de reações. Exemplos comuns de tipos de espécies incluem íons como Ca^{2+} , tais como moléculas de glicose ou ATP, dentre outras. (HUCKA et. al., 2008)
- **Lista de Compartimentos:** Uma lista de objetos ‘contêineres’ (recipientes) onde as espécies podem ser localizadas. Pode-se um modelo conter vários compartimentos de mesmo tipo e, cada espécie deve estar alocada em um. (HUCKA et. al., 2008)
- **Lista de Espécies (objetos):** Um grupo de entidades, ou grupo de objetos do tipo de ‘espécie’ devem ser alocados a um compartimento específico e tem informações relevantes de elemento químico ou entidade participativa de reações. (HUCKA et. al., 2008)
- **Lista de Parâmetros:** Lista de nomes simbólicos com valores determinados. Em SBML, o parâmetro termo é usado em um sentido genérico para se referir a quantidades de constantes ou variáveis em um modelo. O parâmetro definido no nível superior é dito variável global. Também é possível definir os parâmetros que são locais para uma única reação, por exemplo, ou seja, uma declaração de variáveis visíveis em funções de diferentes componentes listados. (HUCKA et. al., 2008)
- **Lista de Assinaturas Iniciais:** Lista de expressões matemáticas utilizadas para determinar as condições iniciais do modelo. Dessa maneira é condicionada uma expressão matemática inicial para utilização em simulações em software, determinando um início de valor. (HUCKA et. al., 2008)

- **Lista de Regras:** Lista de expressões matemáticas adicionadas ao conjunto de equações construídas com base nas reações definidos em um modelo. (HUCKA et. al., 2008)
- **Lista de Restrições:** Lista de restrições é definida por uma expressão computacional booleana com valor ‘true’ ou ‘false’ preenchidos em campos respectivos a alguns componentes do modelo. Ex.: se uma espécie deve modificar a reação de uma ou mais espécies em um determinado tempo. (HUCKA et. al., 2008)
- **Lista de Reações (transformações):** Uma declaração descrevendo alguma transformação, transporte ou processo de vinculação que pode alterar a quantidade de uma ou mais espécies. Por exemplo, uma reação pode descrever como certas entidades (reagentes) são transformadas em determinadas entidades (produtos). Ou outras entidades alteram (modificadores) interferem na velocidade com que ocorrem as reações, ou seja, as reações associadas às expressões cinéticas (de velocidade). (HUCKA et. al., 2008)
- **Lista de Eventos:** Uma declaração descrevendo uma mudança instantânea, descontínua em um conjunto de variáveis de qualquer tipo (espécie de compartimento de quantidade, tamanho ou valor do parâmetro) quando uma condição de disparo está satisfeita, como uma ‘reação de gatilho’, pode ‘disparar’ outra reação. (HUCKA et. al., 2008)

5 SBGN

SBGN (Systems Biology Markup Language) é uma Notação Gráfica para Biologia Sistêmica, como um padrão UML, mas para sistemas biológicos. Existem diferentes formatos de linguagens visuais. São constituídas de diferentes sentenças e representações intencionais atribuídas de significados na forma de três diferentes tipos:

- **SBGN de Descrição de Processos (SBGN-DP):** linguagem que permite a descrição de todos os processos (reações de uma ou mais espécies, por exemplo) que ocorrem num sistema biológico. O conjunto de todos esses processos constitui uma descrição. Utilizada para mostrar todas as interações moleculares de entidades bioquímicas em uma rede (grafo). (LE NOVÈRE et. al., 2009).
- **SBGN Entidade Relacionamento (SBGN-ER):** linguagem que permite a descrição de todas as relações envolvendo entidades de um sistema biológico. O conjunto de todas essas relações constitui um relacionamento. Permite que sejam vistos todas as ‘conexões’ em que uma entidade que participa independente em uma linha de tempo. Assim, relacionamentos podem ser vistos como regras descrevendo as influências de nós de entidades em outros relacionamentos. (LE NOVÈRE et. al., 2009).
- **SBGN Fluxo de Atividades (SBGN-FA):** linguagem que permite a descrição do fluxo de atividades num sistema biológico. Revela o fluxo de informações entre entidades bioquímicas em uma rede. Omite informações sobre os estados de transição de entidades e é particularmente conveniente para representar os efeitos de perturbações, de natureza genética ou ambiental. (LE NOVÈRE et. al., 2009).

Em relação a transcrição de esquemas em SBMLs para SBGN será utilizada a linguagem de visualização gráfica SBGN Diagrama de Processos, descrito mais sucintamente na seção abaixo.

5.1 Estrutura SBGN da Linguagem de Descrição de Processos

SBGN é um padrão de linguagem visual e consiste em sentenças visuais construídas através de arranjos de símbolos em duas dimensões espaciais, neste caso, chamados glifos (LE NOVÈRE, 2009). As sentenças são uma representação intencional atribuída de significado.

A linguagem visual tem importância no momento em que possibilita representar, compreender e modificar de forma direta pela manipulação de representações visuais claras e sem ambigüidade, modelagens de sistemas biológicos reais (MOODIE et. al., 2010).

A seguir, para representar as diferentes representações visuais, os diferentes vocábulos para os diferentes grupos de glifos e as imagens para cada grupo são classificados na Tabela 5.1 logo abaixo.

Componente	Abrev.	Papel desempenhado	Exemplos
Grupos de Nós Entidade	GNE	Uma quantidade limitada de entidades que não podem ser distinguidas por outra	Objetos específicos ou espécies químicas e mais
Nós Contêineres	NC	Uma forma encapsulada de uma ou mais construções SBGN	Complexos, compartimentos e sub-mapas
Nós Processos	NP	Um processo que transforma um ou mais GNEs dentro de um ou mais outros GNEs	Transições, associações e dissociações e mais
Operadores Lógicos	-	Combina uma ou várias entradas em saída única	E, OU, NÃO em booleano
Arcos Conexos	-	Links entre GNEs ou NCs para NPs ou NCs para influencias indicadas	Produção, catálise, inibição e mais

Fonte: Documentação da Linguagem de Descrição de Processo SBGN.

Tabela 5.1 – Representação do vocábulo dos conceitos Glifos utilizado no SBGN

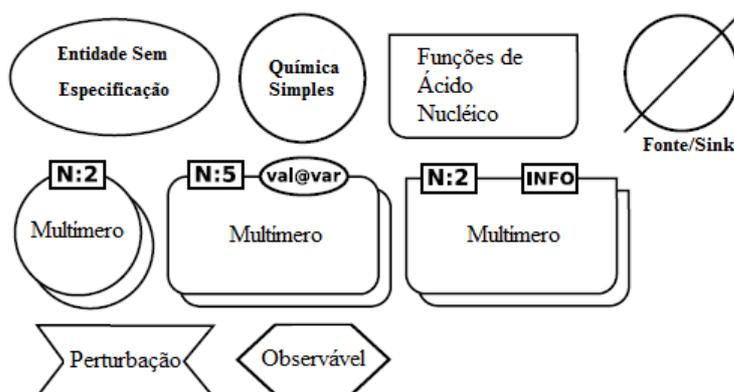
Conforme a Tabela 5.1 existem diferentes classes de glifos. E algumas dessa figuras podem conter determinados tipos de anotações que transmitem algum tipo de informação. Para efeito de interpretação, serão abordados os glifos do Grupo de Nós Entidade (GNE), Nós Contêineres (NC), Nós Processos (NP), Operadores Lógicos e Arcos Conexos.

5.1.1 Grupo de Nós Entidade (GNE)

Um GNE é uma população de entidades que não podem ser distinguidos uma das outras. Por exemplo, todas as entidades moleculares que cumprem inteiramente o mesmo papel em um determinado processo em um ‘grupo’ de entidades. Este grupo pode representar um objeto em diferentes níveis de informação.

O objeto é descrito com todas as instâncias de informações e tudo o que o identifica perante um grupo. São exemplos as diferentes espécies objetos, como o grupo das proteínas, por exemplo. O grupo das proteínas é um jeito de representar uma seleta de indivíduos entidade. Pertencentes de um mesmo compartimento, ou de diferentes grupos (compartimentos). (MOODIE et. al., 2010)

O GNE é composto então pelos glifos: Entidade sem Especificação, Entidade de Química Simples, Macromolécula, Funções de Ácido Nucléico, Multímero, Fonte/Sink, Perturbação, Observável, Tag, Variabilidade de Estado e Marcador Clonado, que são exemplificados, exceto Marcador Clonado, em diferentes glifos na Figura 5.1.



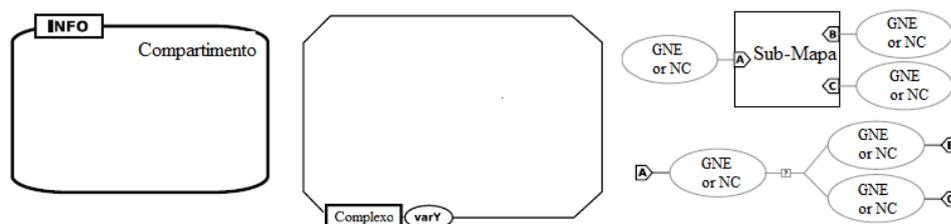
Fonte: Documentação da Linguagem de Descrição de Processo SBGN.

FIGURA 5.1 – Exemplos de glifos do grupo GNE

5.1.2 Nós Contêineres (NC)

Os Nós Contêineres (NC) são construções SBGN que contêm uma ou várias outras construções de processos bioquímicos. Por exemplo, um processo biológico localizado dentro de uma célula contém outros NCs como a representação de uma membrana ou um núcleo, que seria um exemplo de um compartimento ou contêiner.

NCs são compostos então por três nós recipientes: compartimento, complexo e sub-mapa, sendo exemplificados em diferentes imagens na Figura 5.2.



Fonte: Documentação da Linguagem de Descrição de Processo SBGN.

FIGURA 5.2 – Exemplos de glifos do grupo NC

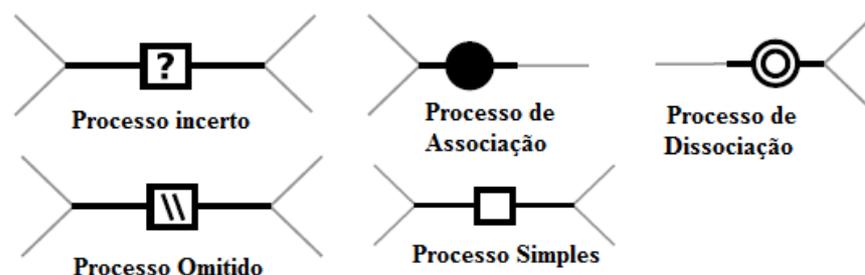
Um complexo pode conter outros NCs e elementos do grupo GNE no seu interior, por exemplo. Um Sub-mapa é um contêiner que é um grupo das entidades de abstração, onde é possível encapsular toda a estrutura de implementação de uma construção SBGN, assim, construir elementos e os ‘obscurecer’ ou esconder a implementação. (MOODIE et. al., 2010)

Os glifos na imagem da Figura 5.2 com inscrições Compartimento e Sub-Mapa são exemplos de glifos NCs Compartimento e Sub-Mapa. A imagem central, com um GNE Tag com inscrição Complexo, é um exemplo de um NC complexo e a o glifo com inscrição varY é um exemplo de GNE de Variabilidade de Estado.

5.1.3 Nós de Processo (NP)

Os Nós Processos (NP) representam processos que transformam (como uma reação) uma ou várias entidades representadas por glifos GNEs em outros diferentes (GNE), através de transições (glifos genéricos) entre elas. (MOODIE et. al., 2010)

NP tem no total quatro transições específicas: processo omitido, processo incerto, associação e dissociação. Exemplificados em diferentes imagens na Figura 5.3. Os glifos NP nesta Figura têm seus nomes que os especificados abaixo, para as diferentes transições.



Fonte: Documentação da Linguagem de Descrição de Processo SBGN.

FIGURA 5.3 – Exemplos de glifos do grupo Nós Processo

5.1.4 Operadores Lógicos

Os Operadores Lógicos são compostos por glifos AND, OR e NOT para diferentes entidades de entrada e uma de saída. Estes glifos são significados com valores booleanos. E tem como formato padrão, um círculo. No centro a inscrição do nome do glifo *AND*, *OR* ou *NOT*. (MOODIE et. al., 2010)

São exemplificados em diferentes imagens na Figura 5.4.



Fonte: Documentação da Linguagem de Descrição de Processo SBGN.

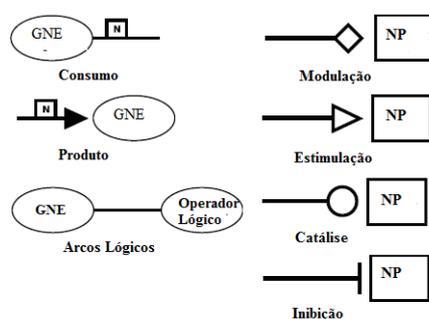
FIGURA 5.4 – Exemplos de glifos do grupo Operadores Lógicos

5.1.5 Arcos Conexos

Os Arcos Conexos são linhas para links de junção entre GNEs e NPs, onde os símbolos ligados às suas extremidades indicam sua semântica.

Esses arcos são compostos pelos glifos: Consumo, Produção, Modulação, Estimulação, Catálise, Inibição, Gatilho, Arco Lógico e Arco de Equivalência. (MOODIE et. al., 2010)

Exemplificados em diferentes imagens na Figura 5.5.



Fonte: Documentação da Linguagem de Descrição de Processo SBGN.

FIGURA 5.5 – Exemplos de glifos do grupo Arcos Conexos

Os glifos com inscrições Consumo, Produto e Arcos Lógicos são exemplos de NPs Consumo, Produto e Arco Lógico. Nas extremidades dos NC estão os glifos que podem formar um diagrama SBGN com esses diferentes NPs, conforme a sintaxe da linguagem. Na imagem do NP Produto, por exemplo, existe um GNE na extremidade.

As demais imagens, com inscrições Modulação, Estimulação, Catálise e Inibição, correspondem respectivamente aos glifos, NP Modulação, Estimulação, Catálise e Inibição e tem nas extremidades relações com diferentes NPs.

6 GML

Anteriormente ao GML, arquivo que representa grafos, existiam diversos programas que trabalhavam com grafos, mas quase todos eles utilizam o seu formato próprio de arquivo, como consequência, a forma de troca de informação entre os diferentes programas é quase impossível pela diferença entre as implementações dos formatos de arquivo. A transferência de informação não é tarefa simples pela quantidade de dados, além da dificuldade de obtenção de resultados satisfatórios da conversão. Por isso, foi desenvolvido um formato de arquivo chamado *Graph Modelling Language*.

GML suporta ao anexar informações arbitrárias aos grafos, diferentes formatações dos nós e arestas, e por isso é capaz de se destacar pela capacidade de adaptabilidade que pode adquirir características de outros formatos de arquivos. GML tem como principais características ser voltada para portabilidade, com uma sintaxe simples, extensível e flexível. Um arquivo GML é composto de uma lista hierárquica de valores-chave. Os grafos podem ser anotados com uma estrutura de dados arbitrária. (KIENLE, 2011)

Algumas características também são evidentes como:

- Os pontos fortes como a fácil *Layout* e uso, leitura/entendimento, requer apenas o texto para ler / escrever, os grafos grandes podem ser facilmente armazenadas em formato de texto, é uma plataforma Independente e as diferentes preocupações são devido à eficiência para processar esses formatos, que não perde em nada para outros formatos de arquivos.
- Os pontos negativos como poder ter diferentes formatações para um GML, assim, o parser para ele deve ser bem estruturado para poder extrair todas as informações, em outras palavras, a implementação do Parser deve ser bem feita.

Como se pode notas, a Figura 6.1 é um exemplo de um GML.

```
01 graph [  
02     comment "This is a sample graph"  
03     directed 1  
04     IsPlanar 1  
05     node [  
06         id 1  
07         label "Node 1"  
08     ]  
09     node [  
10         id 2  
11         label "Node 2"  
12     ]  
13     node [  
14         id 3  
15         label "Node 3"  
16     ]  
17     edge [  
18         source 1  
19         target 2  
20         label "Edge from node 1 to node 2"  
21     ]  
22     edge [  
23         source 2  
24         target 3  
25         label "Edge from node 2 to node 3"  
26     ]  
27     edge [  
28         source 3  
29         target 1  
30         label "Edge from node 3 to node 1"  
31     ]  
32]
```

FIGURA 6.1 – Representa um esquema de um GML.

Neste exemplo, uma chave chamada aresta (edge) contém outras duplas de chaves, por exemplo, a fonte (source) e destino (target), além da identificação (label) que tem valores entre aspas. Os nós, ou vértice, têm identificação (id) e descrição (label). Essa é uma estrutura primária de um documento GML, portanto, um documento sem especificação. No entanto, uma das principais características é justamente a adaptabilidade com formatos

de documentos, respondendo a diferentes formatos de conversão que utilizam essa estrutura sem especificação para tornar novos tipos de arquivos GML, porém especificados conforme um tipo comum de estrutura. Uma especialização, como nos formatos que transferem imagens ou layouts. A especificação em SBML é um dos tipos, que levam em conta a importação de dados biológicos de modelos biológicos. E transformações apenas de layout que usam linguagem visual, como é o caso da especificação SBGN-DP.

```

01graph [
02  directed 1
03    node [
04      id 1
05      zlevel -1
06    cluster [
07      cluster "cell"
08    ]
09    graphics [
10      x 200.0
11      y 100.0
12    ]
13    label "EmptySet"
14    labelgraphics [
15    ]
16    sbml [
17      modelID "Nome_Model_2"
18      sbmlID "EmptySet"
19    ]
20 ]
...

```

FIGURA 6.2 – Apresentação de um nós Espécie

O exemplo do Figura 6.2 é uma forma de controle gráfica (graphics) com a localização do nó (node). É claramente um modelo SBML, pois tem uma chave chamada sbml que abre informações sobre o modelo com modeloID e sbmlID nas linhas 17 e 18 respectivamente. Além de informações de localização do compartimento, que é chamado de cluster (compartimento linha 7). E as posições geográficas da chave graphics (x,y) linha 10 e 11.

Como um grafo é formado de arestas entre nós que tratam de uma abstração gráfica de um sistema biológico documentado. Então devem-se estudar como são tratadas as arestas, a seguir no Figura 6.3.

```

00 edge [
01     id 1
02     source 1
03     target 2
04     graphics [
05
06     ...
07     ]
08     sbml [
09         modelID "EXEMPLO"
10         reversible "false"
11         sbmlID "Reação_1"
12         sbmlRole "product"
13     ]
14 ]

```

FIGURA 6.3 –Figura com informações de uma aresta.

Justamente a posição e informações dos nós (linha 01) e da ligação entre os vértices de destino (linha 03) e fonte (linha 02), que são as identificações que são tratadas para a formação do grafo biológico. As ligações (arestas, linha 00), como mostradas, têm informações relevantes que identificam o formato uma abstração para representação visual que se utiliza do modelo matemático em SBML. Informações do SBML apresentam como modelo, reversão, tipo sbml e papel (respectivamente nas linhas de 9,10,11 e 12).

As identificações respectivamente o nome do sbml e o papel que representa (linhas 11 e 12) de um vértice. No exemplo do Figura 6.3 a aresta (edge) é uma seta que mostra a direção que é a representação do papel de um produto (linha12) pela identificação “product” que é um tipo de aresta que tem como alvo uma espécie (entidade). Além do produto, há o consumo, ou reagente “reactant” (exemplo não adicionado na figura) que é uma base (fonte) da reação. Depende da direção da reação modificará com o termo de um ou outro, ou seja, se a seta indicar a seta para a reação (node) essa será uma reagente, senão, se a seta mostrar-se direcionada a uma espécie, então esta é uma produto.

Porém, tanto quanto o SBML, outro formato pode ter uma especificação, neste caso, especialmente a notação gráfica biológica de descrição de processos bioquímicos SBGN-DP.

6.1 Especificação SBML

A especificação para SBML leva em conta a importação de informações referentes de dados biológicos de modelos biológicos documentos. São importadas informações como nomes de espécies, reações e compartimentos, além de atribuições. Casos de especificação como de produção e consumo em tipos de espécies em reações, que são visíveis pelo documento. Exemplo de forma GML que mostra um nó espécie em SBML é presente no Figura 6.4 abaixo.

```

01 graph [
02   directed 1
03     node [
04         id 1
05         zlevel -1
06     cluster [
07         cluster "cell"
08     ]
09     graphics [
10         x 200.0
11         y 100.0
12     ]
13     label "EmptySet"
14     labelgraphics [
15     ]
16     sbml [
17         modelID "Tyson1991CellModel_2"
18         sbmlID "EmptySet"
19     ]
20 ]
...

```

FIGURA 6.4 – Apresentação de um nós Espécie

6.2 Especificação SBGN-DP

A especificação para SBGN leva em conta a criação de dados biológicos de modelos biológicos, não tão somente documentos, mas a representação gráfica da linguagem SBGN-DP construída, ou seja, não é a linguagem SBGN-DP, e sim, a representação do grafo. As importações de informações como nomes de espécies, reações e compartimentos, por exemplo, são retiradas de documentos SBML para criar uma representação gráfica, podendo se criar diferentes tipos de desenhos que representam modelos biológicos, servindo de base para a criação de diferentes esquemas de modelagens de “projetos” para a construção de pesquisas, ou seja, a criação gráfica como a UML que visa a facilitar ao pesquisador, ou no caso da UML, engenheiros de software a projetar seus softwares. No caso de biólogo, deixa-se claro e sem ambigüidade o desenho de seus sistemas biológico através de notações gráficas padronizadas, no caso, padrões da linguagem de Descrição de Processos SBGN-DP.

O exemplo de documento GML especializado em SBGN é representado pelo nó Figura 6.5 e aresta no Figura 6.6.

```

01 graph [
02     sbgn [
03         role "PROCESSDESCRIPTION"
04     ]
05 directed 1
06 node [
07     id 1
08
09
10     graphics [
11         x 130.0
12         y 260.0
13         ...
14     ]
15     label "glucose"
16     labelgraphics [
17         ...
18     ]
19     sbgn [
20         role "SIMPLECHEMICAL"
21     ]
22 ]

```

FIGURA 6.5 – Exemplo de nó GNE Química Simples

No Figura 6.5 é constituída de um nó com uma identificação (linha 7) que relaciona ao processo químico de um sistema biológico. A linguagem utilizada é denominada na linha 3, o que significa que a linguagem escolhida SBGN-DP para essa representação é a de descrição de processos. A Figura é a representação gráfica de um nó (node) GNE Química Simples, como mostra o papel representativo da GNE (linha 20). Outras informações são extraídas, como a descrição do elemento (linha 15). Além da descrição há a ligação com a localização na tela do elemento gráfico, posições geográficas (linhas 11 e 12).

O papel do nó (linha 20) pode variar, como elementos que significam que são Nós Processo (NP), dentre outros elementos, que representam dentre eles nós GNE, neste caso, até compartimentos.

A aresta é mostrada no Figura 6.6, que representa a ligação entre NPs e GNE.

```

00edge [
01     id 1
02     source 1
03     target 2
04
05
06     graphics [
07         ...
08     ]
09     ...
10
11     sbgn [
12         role "CONSUMPTION"
13     ]
14]

```

FIGURA 6.6 – Aresta GNE Consumo

A aresta (edge) tem uma identificação própria (linha 01) e tem a ligação entre identificação dos vértices fonte (linha 02) e destino (linha 3). A forma como é representado graficamente aresta (linha 06, com o conteúdo da chave graphics). O papel do SBGN (linha 12) , neste exemplo é uma representação de um nó consumo(consumption). Porém podem haver papel de produção de entidades (production), como no caso de para as arestas em

SBML “ product” e “reactant”, respectivamente produção e consumo de uma espécie (GNE).

A união dos elementos gráficos aresta e nós resultamos em um desenho SBGN-DP conforme especificações dessa linguagem.

7 SOFTWARES PARA SISTEMAS BIOLÓGICOS

As ferramentas de softwares para sistemas biológicos existentes oferecem, dentre vários recursos, edição de modelos ou de análise de rede bioquímica, exibindo dados de arquivos resultantes de modelagens para diagramas de visualização gráfica. Exemplos de softwares são:

- Cytoscape (KILLCOYNE et al. 2009);
- CellDesigner (FUNAHASHI et al., 2008);
- EPE (SOROKIN et al., 2006);
- VANTED (JUNKER et al., 2006);
- Arcadia (VILLÉGER et al., 2010);
- Dentre outras.

Normalmente, essas ferramentas interpretam linguagens de descrição em modelos SBML, transformando em diagramas de esquemas padronizados de imagens como o SBGN. (JUNKER et al., 2006)

7.1 CellDesigner

CellDesigner é um editor de diagrama estruturado para a elaboração de redes de regulação genética e bioquímica. As diferentes redes são feitas com base no diagrama de processo, utilizando-se a notação gráfica proposta em SBGN. A representação de modelos por esta ferramenta pode ser vista na imagem da Figura 7.1 (FUNAHASHI et al., 2008).

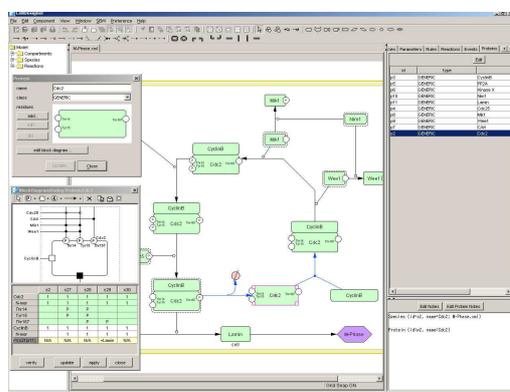


FIGURA 7.1 – Imagem do CellDesigner.

7.2 EPE

EPE (Edinburgh Pathway Editor) é um elaborador gráfico de vias biológicas e aplicativo de edição. Atualmente ele pode desenhar diagramas usando SBGN-DP. Pode exportar SBML e BioPAX a partir da notação de redes bioquímicas. É facilmente configurável para suportar uma ampla variedade de notações gráficas. Além disso, tem uma arquitetura extensível que permite aos exportador vários tipos de representadores de grafos como GML, por exemplo. Imagem do software é apresentado na Figura 7.2. (Sorokin et al., 2006)

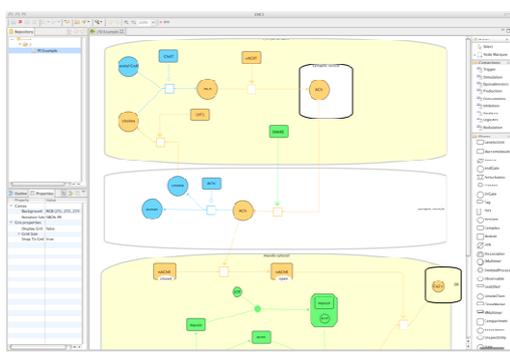


FIGURA 7.2 – Imagem do EDE (Edinburgh Pathway Editor)

7.3 Arcadia

Arcadia traduz através de dados biológicos obtidos de arquivos SBML em diagramas gráficos padronizados, como diagramas SBGN. Os usuários podem ver o mesmo modelo a partir de perspectivas diferentes e alterar facilmente o *layout*. Como mostra na imagem do software na Figura 7.3.

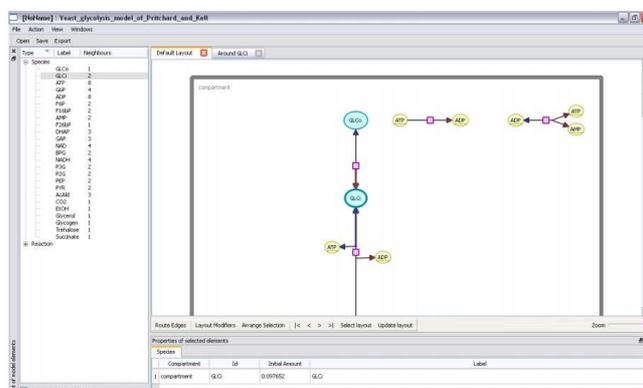


FIGURA 7.3 – Imagem do Arcadia em funcionamento.

7.4 VANTED

VANTED (*Visualization and Analysis of Networks containing Experimental Data*) é uma plataforma independente, disponível gratuitamente, como várias ferramentas de visualizações de dados, porém diferenciada por seus recursos. Esse software ajuda os cientistas a interpretar os seus dados bioquímicos, que são analisados e visualizá-los todos em um contexto pós-modelagem. Este sistema permite carregar e editar gráficos, assim como, representar caminhos biológicos ou hierarquias funcionais de diversas pesquisas biológicas em uma forma inigualável no formato SBGN, utilizando-se da linguagem SBGN-DP. (Junker et. al., 2006)

É possível mapear conjuntos de dados em linguagens de descrição para os elementos gráficos de processos biológicos. Assim, o VANTED pode mostrar na tela diagramas SBGN, além de importar e exportar nos formatos de documentos representativos de grafos como o GML e ler informações relevantes de linguagens de descrição como SBML.

VANTED permite a exibição de dados em um nível até agora inigualável de detalhes em SBGN. Estes recursos em combinação com a simples e intuitiva interface gráfica para o usuário faz dela uma ferramenta muito usada segundo Czaudem et. al. (2010).

A Figura 7.4 mostra em funcionamento essa ferramenta.

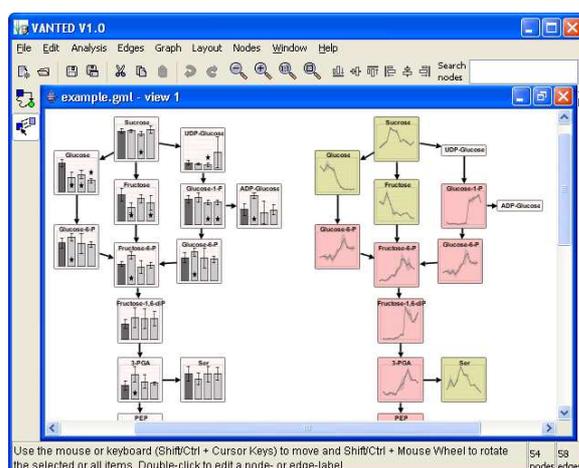


FIGURA 7.4 – Imagem do programa VANTED

8 DESENVOLVIMENTO DA FERRAMENTA

Neste capítulo são abordados os principais pontos no desenvolvimento do presente trabalho, denominado Systems Biology Layout Generator (SBLgen). Apresentando os requisitos principais do Problema a ser trabalhado, os Procedimentos da Ferramenta, assim como, os passos do Algoritmo do SBLgen.

8.1 Requisitos principais do Problema a ser trabalhado

A ferramenta tratou da aplicação de estética e restrição que são fundamentações gráficas no desenho de grafos direcionados, através de minimização de cruzamentos de arestas e agrupamento de vértices. O trabalho tratou de desenho de sistemas biológicos, através da construção do desenho biológico utilizando do layout baseado na linguagem SBGN-DP. Para tal tipo de grafo, arestas possuem um sentido entre vértices. Essas tratadas ou como arestas produtos, reagentes ou modificadoras. Os vértices foram tratados ou como espécies produtos, reagentes, modificadores, sink, fonte ou reações. Em relação aos passos do algoritmo do programa SBLgen, este trata apenas de duas fundamentações gráficas e aplica uma heurística (algoritmo de desenho de grafo) para melhoria iterativa através de minimização de cruzamentos de grafos de modelos biológicos advindos de modelos de dados biológicos em documentos especializados (ver Seção 6.1).

As fundamentações devem satisfazer os seguintes critérios:

- Apresentar arestas com orientação uniforme (por exemplo, voltadas para cima, ou para baixo);
- Mostrar poucos cruzamentos de arestas;
- Agrupar os vértices em agrupamentos (compartimentos).

Em relação ao agrupamento, este é feito com vértices de reações e espécies em compartimentos. E a minimização de cruzamento de arestas é feito entre arestas de conexão entre produtos, reagentes, *sinks* ou fontes representado por linhas retas de tamanhos diferentes. Os vértices, assim como arestas, dependerão das informação obtida exclusivamente do arquivos especializados, neste caso, GMLs exportados pelo programa VANTED (visualizador de dados biológicos).

As principais metas do processo de desenho são de satisfazer um critério estético e de restrição:

- Minimização do número de cruzamentos;
- Agrupamento de reações, espécies, *sinks* e sources.

É importante também destacar que a otimização desse critério não é fácil de ser alcançado, uma vez que envolve algoritmo de complexidade alta, como a exemplo da heurística FR (ver Seção 3.2.2.4). O exemplo de como poderá ficar uma representação de um desenho de um dado biológico é mostrado na Figura 8.1.

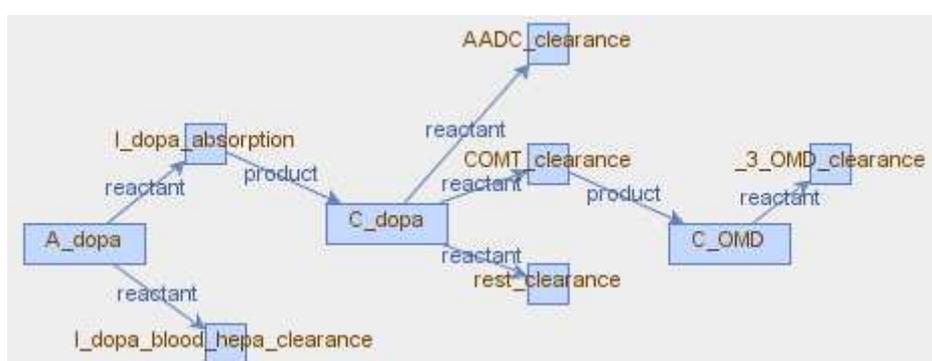


FIGURA 8.1 - Padrão de representação de desenho de grafo de sistemas biológicos

A SBLgen, então, visa minimizar a quantidade de cruzamento de Aretas para o problema de visualização de grafos para sistemas biológicos. No entanto, mesmo minimizando a quantidade de cruzamentos a Figura acima contém alguns problemas, como a não obtenção do saber da dimensão do local onde fica o compartimento de espécies e reações, além de não especificação dos vértices Sink e Fonte (ver Seção 5.1.1), ou seja, se a estrutura está completa e corresponde as definições do SBGN-DP. Desta maneira foi implementada soluções que visam completar o desenho determinando a localização dos vértices das reações e espécies, adicionando-as em compartimentos que delimitam a área destes elementos espécies e reações do grafo biológico na tela.

A ferramenta permite ao usuário visualizar melhor ao executar ações de melhoraria do layout do desenho, através da importação de arquivos GMLs especializados de modelos biológicos em SBML. O programa então, após importação, aplicar uma heurística (minimizadora de cruzamentos) com reposicionamento de espécies e reações no grafo do desenho definido com base dos requisitos da linguagem SBGN proposta. Assim, o SBLgen pré-processa uma adequação do GML para a saída visual no formato SBGN-DP

que assim soma espécies produtos e reagentes no modelo biológico que possam cumprir com a padronização gráfica com a finalidade de completar o modelo do desenho, neste caso, devendo atender aos requisitos de adicionar entidades sinks (produtos não especificados) e sources (reagentes não especificados), componentes das reações que não são, primeiramente, descritos no modelo biológico em SBML. Além da adequação, as reações ou espécies que não tiverem um compartimento explícito.

Ao final do processo pré-processo após os procedimentos exigidos pelo programa, a heurística (FR , 1988) é aplicada.

8.2 Procedimentos Gerais

Os procedimentos gerais mostram os passos do software, desde a importação, aos processos mais específicos de software na Figura 8.2 abaixo.

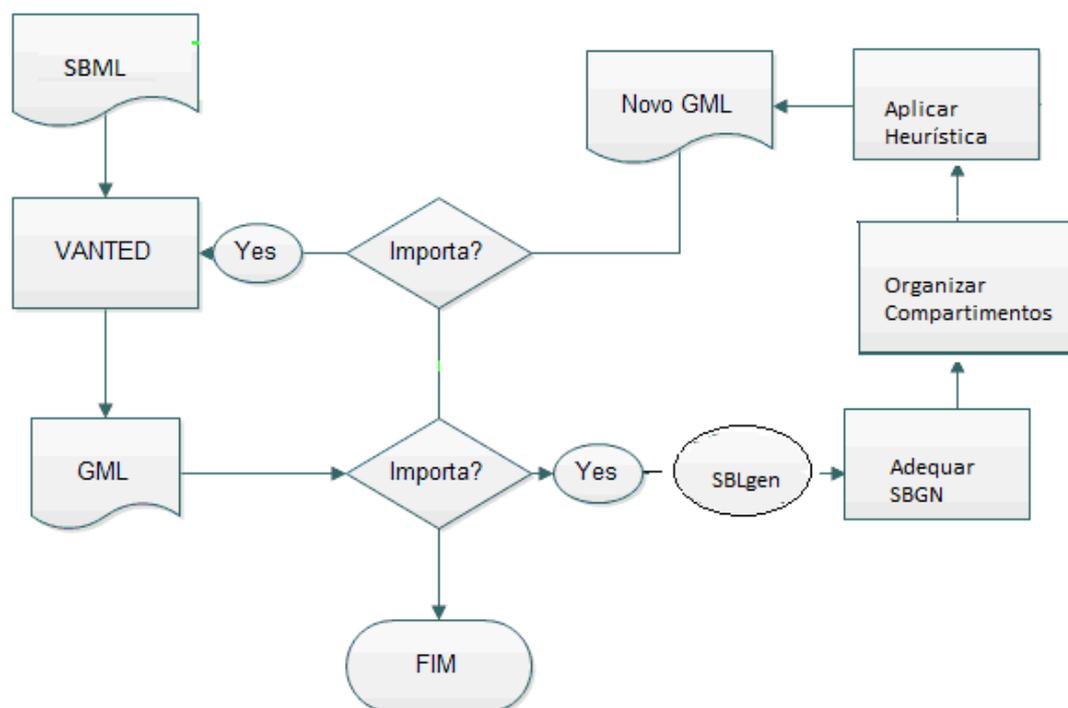


FIGURA 8.2 – Procedimento Geral da ferramenta

No procedimento acima, em um cenário real, o usuário deverá optar por importar GML, que foi especializado em SBML do programa VANTED. Se o fizer, então a ferramenta vai adequar para o formato SBGN, depois organizar os compartimentos e aplicar a heurística, executando o algoritmo de FR (ver Seção 3.2.2.4). O SBLgen então exporta um GML que poderá ser importado para o VANTED novamente, se o usuário se permitir.

8.2.1 Adequar SBGN

O primeiro procedimento da ferramenta SBLgen é o de importação das informações de dados biológicos no formato GML. Após importação é adequado a estrutura SBGN o esquema do desenho, onde a ferramenta SBLgen faz a adição dos elementos à lista de compartimentos, que não são relacionados no modelo SBML, denominados sinks/fontes (ver Seção 5.1.1), que são as espécies produtos/reagentes não especificadas. A Figura 8.3 mostra reações, que são representadas por quadrados e espécies por retângulos. Como pode-se notar a reação um (1), que não tem ao menos uma espécie reagente, e que a falta cria a necessidade de adequação com a adição de entidades chamadas de fonte. Em contrapartida a reação 5 é um caso de falta de produto, portanto é necessária uma sink .

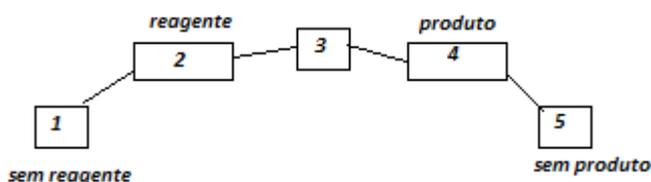


FIGURA 8.3 – Representação de um Sistema Biológico

Além de determinar os sinks e fontes, para reações supracitadas, devem-se determinar também a localidade espacial onde ocorre o processo químico, pois elas não tem determinação do compartimento (local), pois também o arquivo GML não oferece informações necessárias de compartimento para elas. Para determinar um local, uma reação deve ter todos suas espécies produtos e reagentes de um mesmo compartimento. Essa é a primeira cláusula, se verdadeira, então a reação vai pertencer ao compartimento dos elementos que ela se interliga por arestas, senão então vai pertencer a um chamado parent, que a representa o compartimento geral de espécies, reação e compartimentos sem local específico de hierarquia. Os exemplos relacionados a essas cláusulas seriam o do quadrado da reação três (3) e a reação um (1), respectivamente pertencentes a um compartimento 'específico' e 'parent'.

A Figura 8.4 mostra como está a configuração inicial de uma importação de GML. Os círculos em branco são as representações das reações que não tem definidos os compartimentos. Os retângulos em amarelo e azul fazem a alusão a espécies de dois tipos diferentes de compartimentos e o retângulo cinza é um sink/fonte a ser adicionado ao grafo.

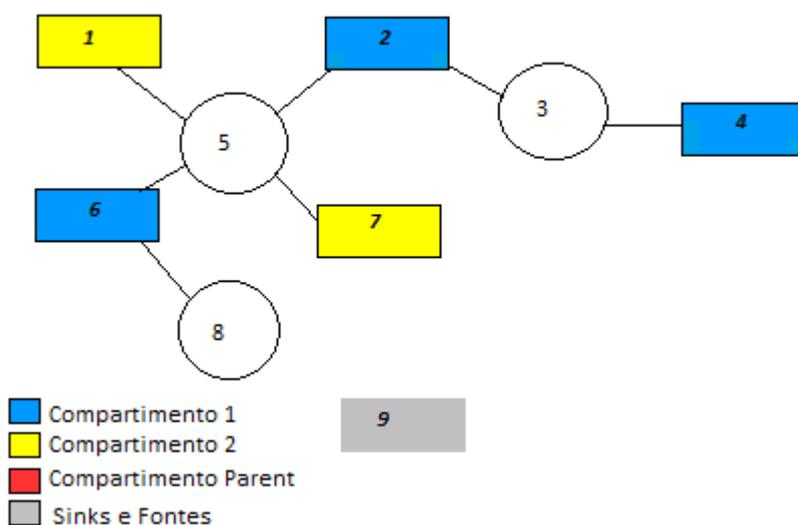


FIGURA 8.4- Representação do GML de um Sistema Biológico antes de Adequar SBGN

Em relação a Adicionar Sources e Sink, nela é mostrada no exemplo da Figura 8.5 diferentes cores que fazem referência a diferentes compartimentos, como dito na Figura anterior. Os círculos correspondem as reações e os retângulo às espécies. O programa faz adições de espécies não especificadas (sink/fonte), em retângulo cinza, com interligações entre uma reação em círculo vermelho, neste caso o número oito.

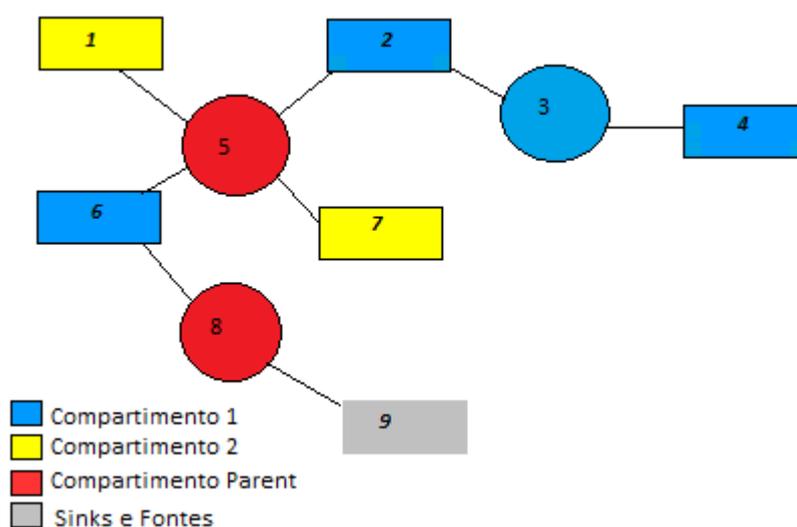


FIGURA 8.5- Representação do GML após Adequar SBGN

Na Figura 8.5 é notada a inserção de cores nas reações em reação a Figura 8.5, significando que foi adicionada uma referência para os compartimentos, ou seja, o procedimento então encontra o compartimento para as reações. A reação três tem a cor azul,

pois todos as espécies que interligam com ela são do compartimento um, portanto ela pertencerá ao compartimento um. No caso das reações cinco e oito, pertencem ao compartimento *parent*, pois as espécies que se interligam a elas são de diferentes compartimentos. Desta maneira, quando uma reação tiver todos as espécies que se ligam a ela de um mesmo compartimento, então essa reação pertencerá a esse compartimento, senão ao *parent*.

8.2.2 Organizar Compartimentos

Na organização do compartimento os nós espécies e reações são inseridos em listas que representam os compartimentos. Após inserção são determinados os tamanhos e posições dos compartimentos, com a finalidade de hierarquizá-los dentro do *parent*. Após hierarquizar são organizadas as posições dos elementos das listas dos compartimentos.

O SBLgen separa em compartimentos os elementos espécies e reações em uma lista nós compartimentos (locais). Após preenchidas essas listas, determinam a dimensão de altura e largura do tamanho de cada compartimento. A quantidade de elementos listados que compõem cada um compartimento é multiplicado por uma distância prefixada dividida por dois, que determina altura e largura de cada compartimento. Por exemplo, se houver dez elementos em um compartimento e a distância de dez pixels entre os elementos, o tamanho de altura e largura na tela serão de cinquenta pixels.

A posição (x,y) de um compartimento atual de uma lista de compartimentos será determinada pela dimensão (w,l) a partir do compartimento anterior. A posição do compartimento anterior vezes a dimensão fixada a cada inserção na tela. Ou seja, um exemplo é dado de uma posição (0,0) de um compartimento e uma dimensão do tamanho da altura e largura (10,10) de outro, a localização do próximo compartimento será determinada para posição geográfica (0,20).

Ao determinar os tamanho para os compartimentos e dimensioná-los, o processo do parágrafo anterior, então posiciona lado a lado os compartimentos na tela em como mostra a Figura 8.6. É definido um espaçamento entre os diferentes compartimentos e sink/fonte, pois posiciona no espaço os elementos em compartimentos, separando-os em locais com dimensões estáticas através de uma referência ao compartimento que cada grupo de nó tem.

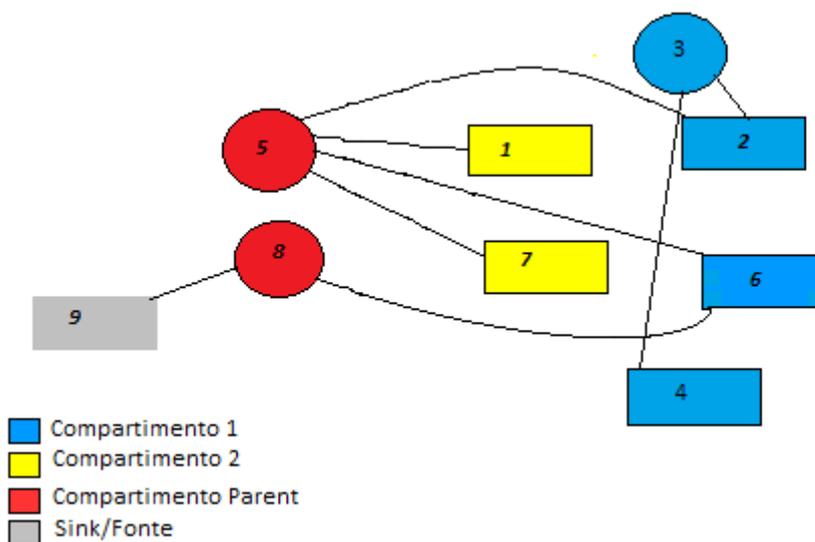


FIGURA 8.6 – Dimensionamento e Posicionamento dos Compartimentos.

Por último, então determina que os compartimentos fiquem dentro o *parent* e esquematiza a inserção de nós posicionando-os (x,y) randomicamente na tela, limitando-os dentro da localização e dimensão dos compartimentos relacionados a cada inserido. A Figura abaixo mostra o esquema hierárquico adotado com base no *parent*.

Após a ferramenta aninhar os compartimentos e inserir internamente as espécies, reações e agora compartimentos no *parent* ficarão como na Figura 8.7.

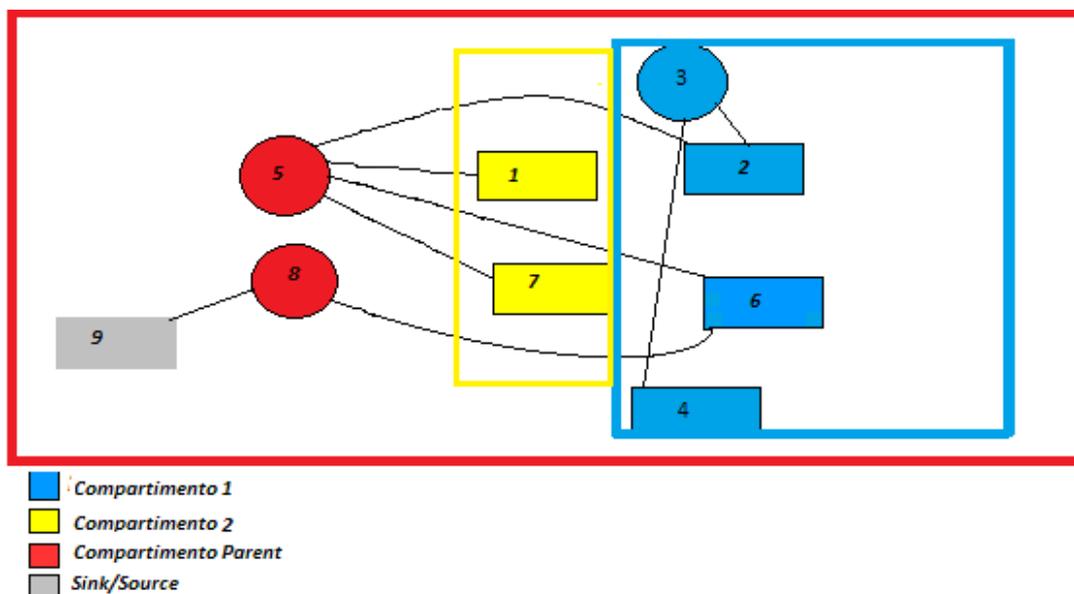


FIGURA 8.7 – Hierarquizar e Inserção Interna no frame *Parent*

8.2.3 Aplicar Heurística

Após organizados os compartimentos, então se aplicam uma heurística ou no caso o algoritmo de FR que deixará algo parecido como na Figura 8.8. A aplicação da heurística leva em conta a expansão e retração de arestas a fim de minimizar a quantidade média de cruzamentos entre elas. As arestas são como molas que podem expandir e retrair com o calor, como o exemplo da Figura abaixo. A expansão é limitada pelo local ou compartimento de cada nó, pois a heurística garante que não ultrapasse os limites das bordas dos compartimentos.

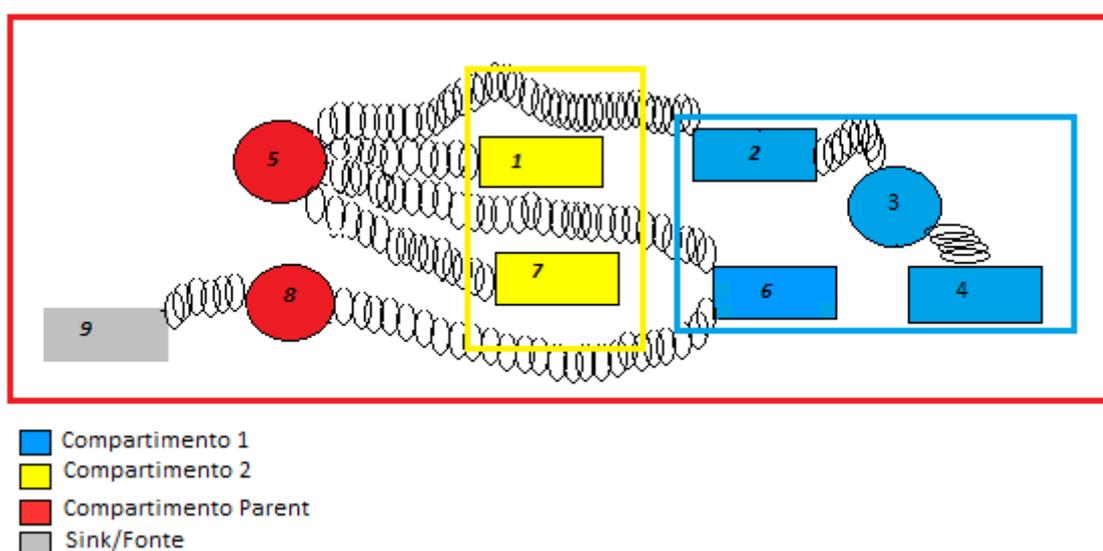


FIGURA 8.8– Aplicação do Algoritmo FR Modificado

8.3 Implementação da Ferramenta

A implementação leva em conta os algoritmos que foram utilizados que determinam os procedimentos e ações do software. Os algoritmos responsáveis por adequar, organizar e heurística aplicada serão explicados nas subseções seguintes da seção Desenvolvimento do SBLgen.

8.3.1 Técnicas e ferramentas utilizadas

A implementação foi realizada ao utilizar a linguagem de programação Java, permitindo a portabilidade da ferramenta para diversos sistemas operacionais, além de

importações de GMLs especializados de SBML e GML especializado em SBGN-DP para exportação do resultado final da execução.

8.3.2 Desenvolvimento do SBLgen

O SBLgen é baseado no conceito de grafos direcionados, mais especificamente de grafos biológicos com vértices (espécies, reações), arestas e subgrafos (sistemas biológicos hierarquizados em compartimentos). Esses conceitos correspondem aos escritos do funcionamento dos algoritmos dessa ferramenta. E esta seção apresenta as principais estruturas funcionais da ferramenta SBLgen.

8.3.2.1 Algoritmo de Adequar SBGN

A Adequação para SBGN, vista na seção 8.2.1, é responsável por adequar o grafo do sistema biológico para uma estrutura próxima a realidade de um desenho SBGN-DP. Esse procedimento adiciona Sinks e Fontes, que são elementos espécies que não são informadas no grafo do GML especializado em SBML. Além de não obter os *sinks e fontes*, a estrutura não informa as reações específicas o compartimento em que se encontram. Assim são adicionadas, por esse procedimento, através do algoritmo específico Adequar SBGN-DP.

```

0 função AdequarSBGN (Lista arestas, Lista nos) início
02         adicionarSINKSOURCES(arestas, nos);
03         relacionarCompartimentoAoNo(arestas, nos);
04         fim

```

FIGURA 8.9 – Chamadas de Procedimentos do AdequarSBGN

O procedimento AdequarSBGN faz uma ligação com a lista de nós totais para os diferentes Compartimentos e listas de nós e arestas.

A chamada ‘adicionarSINKSOURCES’ tem dois parâmetros como entrada que são as lista de arestas e lista de nós. Para cada nó, se houver um que não tem um vértice fonte então criar-se-á um novo. Essa nova espécie será a fonte. Entidade ou espécie que terá uma ligação ou aresta com o nó investigado, que neste caso, reação destino. No caso contrário, na inserção de sinks, se não houver uma espécie destino então está será um sink. Após a criação dos nós, serão atualizadas as listas de nós e Arestas. Para a localização na tela

essas espécies adicionadas, não especificadas, serão inseridas randomicamente dentro do espaçado do *parent*.

A chamada ‘relacionarCompartmentoAoNo’ relaciona um compartimento para cada nó que não tiver explícito um, como no caso das reações. O algoritmo procura um nó que não tem compartimento explícito, se esse nó tiver ligações com vértices que pertencem a um mesmo compartimento, então esse nó ou reação pertencerá ao compartimento em que todos esses vértices estão ligados, porém se houver no mínimo um vértice que pertence a um compartimento diferente dos demais, então a reação que não tem compartimento explícito será do compartimento *parent*.

8.3.2.2 Algoritmo de Organizar os Compartimentos

Organizar os Compartimentos, vista na seção 8.2.2, como intuitivamente faz entender, hierarquiza e organiza internamente os compartimentos. Organizando as informações relevantes ao grafo de um sistema biológico (lista de compartimentos). E o trecho do código abaixo mostra procedimentos desse algoritmo.

```

01 função OrganizarCompartimentos(Lista arestas, Lista nós) início
02             agrupamento(arestas, nós) ;
03             organizacaoInterna();
04 fim

```

FIGURA 8.10 – Procedimento que organiza os compartimentos.

A chamada ‘agrupamento’ dimensiona e hierarquiza os compartimentos. Para cada nó que representa um agrupamento ou compartimento pegam-se o número de reações e espécies desse e multiplica-se pela distância prefixada, e assim, determinam-se o tamanho das dimensões de altura e largura para esse compartimento. A localização desse compartimento é determinada pelas informações das dimensões do compartimento anterior. Utilizando-se do dobro do tamanho do compartimento anterior para determinar a localização próximo compartimento, ou seja, locando conforme determinação da posição anterior.

Em relação à chamada ‘organizacaoInterna’ utilizam-se dos nós agrupados dos compartimentos, retendo as informações de cada espécie e reação (vértices) para lista específica de compartimento, lançando-os da lista randomicamente dentro do local do compartimento, limitando as localidades pelo tamanho e localização do compartimento em que estão alocados.

Após lançamento randômico dos elementos componentes dentro dos compartimentos, então finaliza-se esse processo.

8.3.2.3 Algoritmo de Aplicar Heurística

Aplicar Heurística, vista na seção 8.2.3, é a mostra da implementação do procedimento da heurística FR modificada. O código abaixo é uma adaptação do Algoritmo FR, explicado na (seção 3.2.2.3), porém para sistemas baseados em agrupamentos/compartimentos de dados biológicos estas modificações foram necessárias, uma vez que, os sub-procedimentos desta força direcionada possuem, em via de regra, ter maior preocupação com o esquema de simetria e hierarquia entre de compartimentos, espécies e reações que visa a redução da quantidade de cruzamento de arestas para alcançar um ponto de simetria².

O método original FR, e descrito na Seção 3.2.2.3, é um método que foi modificado na presente pesquisa do trabalho. A estrutura mostrada abaixo na Figura 8.11 foi reutilizada com base mesma notação do Algoritmo FR:

² Não é objetivo desse trabalho avaliar o critério estético “simetria”, todavia, o algoritmo *Spring FR* naturalmente produz desenhos simétricos, assim, tal critério acaba estando indiretamente presente nos desenhos produzidos. Contudo, não é computada para efeito de avaliação de quantidade.

Heurística Spring FR modificada(Compartimentos, E)

```

01 {W e L são largura e altura de um compartimento e Compartimentos é uma lista de compartimentos. }
03 função k(c) = início raiz ((c.W*c.L)/|c.V|) fim;
04 função fa(x,c) = início return x2/k(c) fim;
05 função fr(x,c) = início return k(c)2/x fim;
06 função dif(p1, p2) = início p3.x=p1.x-p2.x; p3.y=p1.y-p2.y; return p3 fim;
07 função cool(t) = início return t*0.1 fim;
08 t = 10000; numero_cruzamento_antigo = calculo_cruzamento(Compartimentos);
09 enquanto t > 0.0001 faça início
10     para cada c em Compartimentos faça
11         para cada v em c.V faça início
12             v.d.x = 0; v.d.y = 0;
13             para cada u em c.V faça início
14                 se (u != v) então início
15                     delta = dif(v.pos, u.pos);
16                     v.d.x += (delta.x/|delta|)*fr(|delta|,c)
17                     v.d.y += (delta.y/|delta|)*fr(|delta|,c)
18                 fim { do se da linha 14 }
19             fim { do para da linha 11 }
20         para cada c em Compartimentos faça
21             para cada (v,u) em E faça início
22                 {cada aresta é um par ordenado de vértices v e u}
23                 delta = dif(v.pos, u.pos);
24                 se (v.c == u.c) então início
25                     v.d.x -= (delta.x/|delta|)*fa(|delta|,v.c); v.d.y=(delta.x/|delta|)*fa(|delta|,v.c);
26                     u.d.y += (delta/|delta|)*fa(|delta|,u.c); u.d.y = (delta.x/|delta|)*fa(|delta|,v.c);
27                 fim { do se linha 24 }
28             fim { do para da linha 21 }
29         para cada c em Compartimentos faça
30             para cada v em c.V faça início
31                 ant.pos = v.pos;
32                 novo.pos.x = v.pos.x + (v.d.y/|v.d|)*min(v.d,t); novo.pos.y = v.pos.y + (v.d.y/|v.d|)*min(v.d,t);
33                 v.pos = melhor_movimento(ant.pos,novo.pos,Compartimentos);
34                 v.pos.x = min(c.W/2, Max(-c.W/2, v.pos.x));
35                 v.pos.y = min(c.L/2, Max(-c.L/2, v.pos.y));
36                 se (numero_cruzamento_antigo > calculo_cruzamento(Compartimentos)) então início
37                     t=cool(t); numero_cruzamento_antigo = calculo_cruzamento(Compartiment );fim{do se linha 36}
38                 fim {do para linha 30}
39             fim {do para linha 29}
40     t = cool(t); fim {do enquanto linha 9}

```

FIGURA 8.11 – Algoritmo FR modificado

Na Figura 8.11 acima, as variáveis W , L e V indicam, respectivamente, largura, altura e lista de vértices de um compartimento. Enquanto que d e pos indicam, respectivamente, os vetores (x,y) deslocamento e posição de um vértice. E $delta$ é o vetor diferença entre a posição de dois vértices distintos. As variáveis d são referências de vetores posição que guardam o deslocamento para um vértice.

A função k , diferente da variável k no algoritmo, recebe por parâmetro um compartimento, obtendo a largura, altura e número de vértices. Referência desse compartimento, para serem usados no cálculo dessa função. Outra diferença ocorre nas funções fr e fa que recebem um novo parâmetro em relação ao original, passando a k o compartimento parametrizado.

Além da função k , são adicionadas novas como os casos de *dif* e *calculo_cruzamento*. A função *dif* é a diferença entre valores de vetores posição passados por parâmetro. E *calculo_cruzamento* é a obtenção da melhor configuração da lista de compartimentos, avaliando se minimizou o número de cruzamentos entre arestas dados dois vetores posições de vértices e a lista de compartimentos passados por parâmetro, retornando, assim, um vetor posição que obteve o menor número de cruzamentos globais.

A primeira parte do funcionamento do algoritmo, leva em conta o '*para*' localizado na linha 10, significando que para todos os compartimentos são calculadas as forças de repulsão entre os vértices de um mesmo compartimento, guardando num vetor posição dentro d para cada vértice.

Em relação ao segundo passo, são calculadas as forças de atração entre vértices, que possuem aresta entre si e que estão no mesmo compartimento. É vista a cláusula que garante que seja do mesmo compartimento na linha 24.

A terceira e última parte, para cada vértice de um compartimento, são avaliados o deslocamento antigo e o novo de um vértice na função *melhor_movimento*, na linha 33 que retorna a melhor configuração global. Além disso, as posições dos vértices ficam limitadas ao tamanho do compartimento (linhas 34 e 35) e o número de iterações do algoritmo é dado por uma função de resfriamento (função *cool*), que tem temperatura inicial dez mil (10000) e, reduz para dez por cento (10%) de seu valor a cada iteração. A redução da temperatura pode acelerar, caso ocorra redução do número de cruzamentos ao reposicionar um vértice (linhas 36 e 37), assim, ocorre um novo resfriamento.

Ao final do processo, haverá um desenho simétrico dentro de frame dos compartimentos.

9 TESTES

O objetivo dos testes visam executar o algoritmo de heurística FR avaliando a variabilidade da quantidade geral de cruzamentos através de testes em diferentes modelos biológicos documentados e escolhidos do site BioModels (www.ebi.ac.uk/biomodels). O primeiro teste leva em conta modelos com arestas e nós de tamanho parecidos. Enquanto que o segundo teste leva em conta uma quantidade bem maior de arestas em relação aos nós, sendo quase o dobro de arestas em relação aos nós. Em ambos os testes, neles haverá uma tabela que informação a quantidade de vértices totais (com a quantidade de sinks e fontes, espécies). Em relação aos resultados, estes levam em conta a temperatura e quantidade média de reduções de cruzamentos entre arestas e porcentagem de cruzamentos reduzidos.

9.1 Teste em modelos com quantidade de arestas superiores aos nós

A avaliação utiliza dois modelos de sistemas biológicos com quantidade de compartimento variados e com quantidade bem maior de arestas em relação aos nós, sendo apresentados os resultados nos Gráficos 1 e 2 com a explicação logo abaixo dos resultados da comparação da relação do número de cruzamentos ao longo do esfriamento da “temperatura” do desenho do grafo biológico e porcentagem de redução nos diferentes modelos.

Os modelos dos sistemas biológicos comparados têm as seguintes configurações:

- Modelo dezoito (18) este contém mais informações na página (<http://www.ebi.ac.uk/biomodels-main/BIOMD0000000018>);
- Modelo cento e quarenta e quatro (144) este contém mais informações na página (<http://www.ebi.ac.uk/biomodels-main/BIOMD0000000144>).

Estes modelos estão mais detalhados na tabela 9.1.

A tabela 9.1 listas as informações de quantidade de dados biológicos obtidos dos modelos dezoito (18) e cento e quarenta e quatro (144). Informando a quantidade de compartimento, de arestas interligadas, quantidade de nós sink/fonte, além das espécies e reações.

Modelo	Compartimentos	Arestas	Nós Sinks/Fonte	Nós Espécies	Nós Reações
18	2	167	20	33	47
144	2	204	32	18	54

Tabela 9.1 – Lista de informações gerais de dois diferentes modelos

A tabela abaixo tem informações específicas de quantidades geral dos nós e arestas. Os nós somados dos sink/fontes mais Nós Espécies, mais Compartimento e Nós Reações resultada na quantidade geral de Nós. Em contrapartida, as Arestas então são fichadas das interligações entre todos os nós.

Modelo	Nós	Arestas
18	100	167
144	104	204

Tabela 9.2 – Lista de informações específica de três diferentes modelos

Abaixo a relação do número de cruzamentos com temperatura entre dez mil e próximo de zero. Cada linha de cruzamento/temperatura é uma média entre cinco testes.

9.1.1 Média de redução de cruzamento pela temperatura

Em relação à média feita foram somados os números de cruzamentos de cinco testes para cada temperatura e foram divididas pelo número de testes e resultando na média de redução do cruzamento para cada temperatura do gráfico, mostrado no gráfico 1 abaixo.

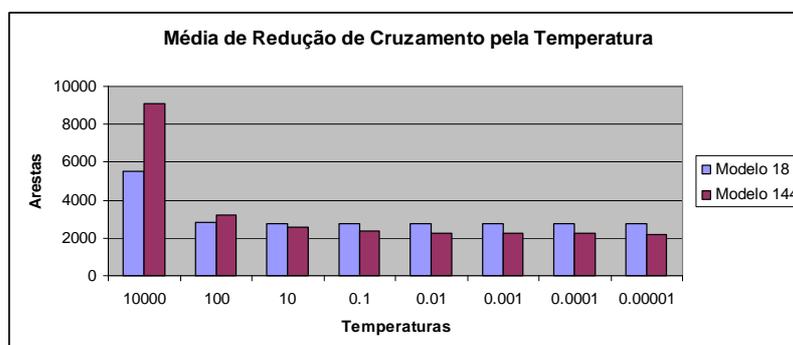


Gráfico 1 –Gráfico de barras de Redução de Cruzamento Médio ao longo do Esfriamento da Temperatura para os Modelos 18 e 144.

O Gráfico 1 mostra a tabela de barras da redução do cruzamento médio ao longo do esfriamento da temperatura que controla a atração e repulsão entre os nós do desenho de grafo, como mostra os modelos dezoito e modelo cento e quanta e quatro para ambos a quantidade médias acima de cinco mil cruzamentos iniciais, anteriores a aplicação da temperatura média de dez mil graus (10000°). Quando a temperatura do modelo dezoito se aproxima de zero, estiliza em pouco mais de dois mil cruzamentos, deixando o modelo dezoito (18) com uma quantidade média de dois mil e duzentos (2200) cruzamentos.

Os modelos têm muita redução pelo número de cruzamentos, pois tem inicialmente pela grande gama de cruzamentos e estabilizam a quantidade quando próximos a temperatura de cem graus (100°). Isso se deve a condição de limitação ao tamanho do compartimento (local frame da tela), pois a quantidade dentro de um compartimento determina o tamanho, se a quantidade no interior de um compartimento for reduzido, então os vértices não se moverão tanto, assim, não havendo tantas reduções.

9.1.2 Redução média de cruzamentos em relação à configuração inicial

O gráfico dois mostra a redução média de cruzamentos entre a temperatura mais alta e a temperatura mais estável nos modelos com apenas um compartimento.

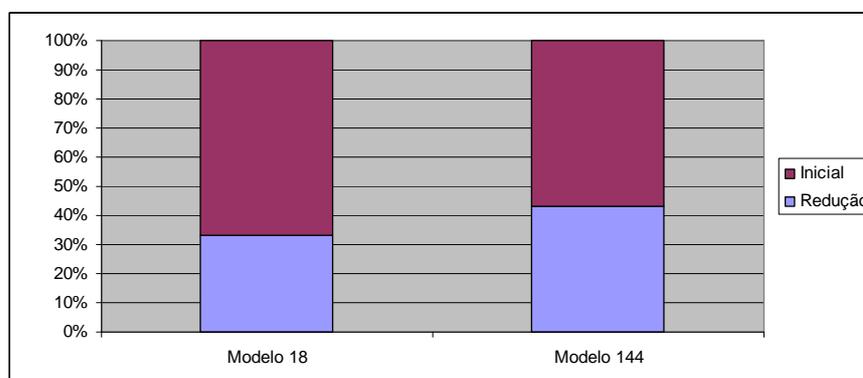


Gráfico 2 – Redução média de cruzamento dada a condição inicial em relação a redução propriamente dita

Como mostra no modelo dezoito, no Gráfico 2, há uma redução média de cruzamentos de cerca de trinta percento (30%) em relação as configurações iniciais, enquanto que o modelo cento e quarenta e quatro deve uma redução maior de quarenta e dois percento (42%).

9.2 Teste em modelos com número de nós e arestas similares

A avaliação utiliza três diferentes configurações de modelos e estes foram escolhido pois a quantidade de arestas totais e nós totais deles são relativamente parecidas, porém com quantidade de compartimento variados, sendo apresentados os resultados nos Gráficos 2 e 3 com a explicação logo abaixo dos resultados da comparação da relação do número de cruzamentos ao longo do esfriamento da “temperatura” do desenho do grafo biológico e também da porcentagem de redução.

Os modelos dos sistemas biológicos comparados têm os seguintes arquivo :

- Modelo duzentos e dois (202) este contém mais informações na página (<http://www.ebi.ac.uk/biomodels-main/BIOMD0000000202>)
- Modelo três (003) este contém mais informações na página (<http://www.ebi.ac.uk/biomodels-main/BIOMD0000000003>)
- Modelo quatro (004) este contém mais informações na página (<http://www.ebi.ac.uk/biomodels-main/BIOMD0000000004>)

A tabela abaixo tem informações específicas de quantidades geral dos nós e arestas. Os nós somados dos sink/fontes mais Nós Espécies, mais Compartimento e Nós Reações resultada na quantidade geral de Nós. Em contrapartida, as Arestas então são fichadas das interligações entre todos os nós.

Modelo	Compartimentos	Arestas	Sinks/Fonte	Nós Espécies	Nós Reações
202	2	36	23	5	7
003	2	14	7	3	7
004	2	14	3	5	7

Tabela 9.3 – Lista de informações gerais de três diferentes modelos

A tabela abaixo tem informações específicas de quantidades geral dos nós e arestas. Os nós somados dos sink/fontes mais Nós Espécies, mais Compartimento e Nós Reações resultada na quantidade geral de Nós. Em contrapartida, as Arestas então são fichadas das interligações entre todos os nós.

Modelo	Nós totais	Arestas totais
202	35	37
003	17	19
004	15	17

Tabela 9.4 – Lista de informações específica de três diferentes modelos

9.2.1 Média de redução de cruzamento pela temperatura

Em relação à média feita, foram somados os números de cruzamentos de cinco testes para cada temperatura e foram divididas pelo número de testes e resultou na média de redução do cruzamento para cada temperatura do gráfico, mostrado no Gráfico 3 abaixo.

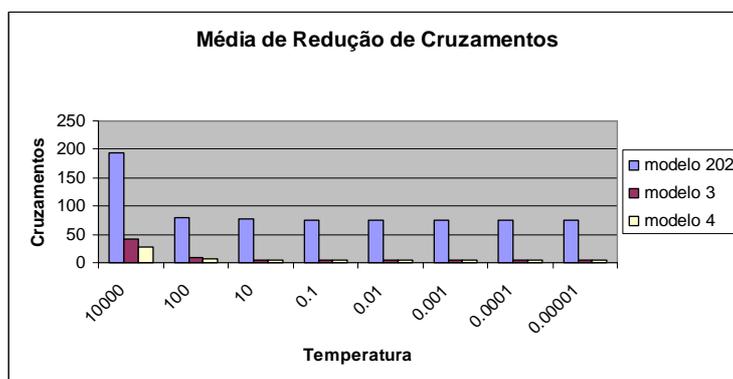


Gráfico 3 – Gráfico de barras de Redução de Cruzamento Médio ao longo do Esfriamento da Temperatura para os Modelos 202, 3 e 4.

O Gráfico três mostra a o esquema de linha (pela temperatura) da redução do cruzamento médio ao longo do esfriamento da temperatura que controla a atração e repulsão entre os nós do desenho de grafo, como mostra o modelo duzentos e dois com quantidade média acima de cento e setenta (170) cruzamentos iniciais, anteriores a aplicação da temperatura média de dez mil graus (10000°), que então apresenta um redução estável para cinquenta e seis cruzamentos. Enquanto a temperatura do modelo três se aproxima de zero estiliza em um pouco mais de dois cruzamentos (2). Enquanto que outro modelo quatro fica em quatro cruzamentos.

9.2.2 Redução média de cruzamentos em relação à configuração inicial

O gráfico dois mostra a porcentagem da redução média de cruzamentos da relação entre a configuração Inicial e quantidade reduzida.

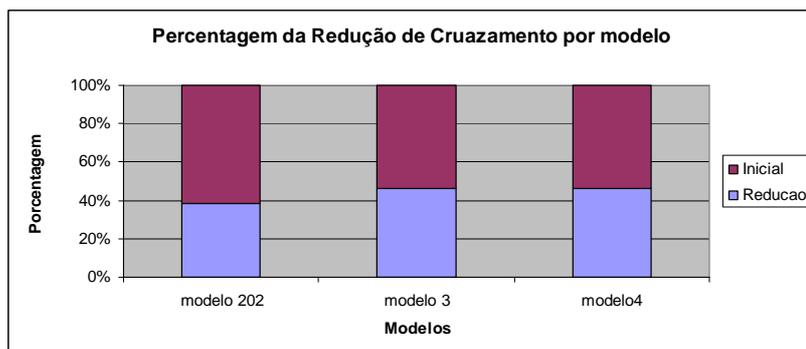


Gráfico 4 – Redução média de cruzamento dada a condição inicial em relação à redução propriamente dita

O gráfico quatro mostra a redução média de cruzamentos entre a temperatura mais alta e a temperatura mais estável nos modelos com mais de um compartimento.

Como mostra no modelo duzentos e dois (202) no gráfico há uma redução média de cruzamentos na temperatura mais alta de quarenta percento (40) dos cruzamentos. Enquanto que nos modelos três e quatro uma diferença com redução média quarenta e cinco percento (45%) cruzamentos reduzidos na temperatura mais estável.

10 DISCUSSÃO

O presente trabalho de monografia descreveu a aplicação do algoritmo modificado do FR, algoritmo de força direcionada tradicional, após aplicação da adequação para comportar o formato SBGN-DP e organização de compartimentos em desenhos biológicos. As modificações do FR permitiram a aplicação para representações gráficas de sistemas biológicos, preservando o posicionamento dos nós em seus devidos compartimentos.

Dentro da análise dos testes, foram possíveis obter informações relevantes que demonstraram a redução acentuada da quantidade média de cruzamentos de arestas para o desenho. O trabalho teve o foco principal voltado a este tipo de redução, pois o algoritmo FR original trata este critério de suma importância, o que não é diferente do algoritmo modificado do programa.

Como o critério estético de redução é o principal fundamento gráfico do algoritmo, sendo o segundo de restrição do agrupamento de vértices que neste caso fazem referência aos compartimentos para as espécies, reações, *sinks* e fontes. No entanto, as reações, que são nós também do grafo ou rede bioquímica, não tem um local definido nos modelos biológicos, pois eles não fornecem informações suficientes sobre seu compartimento, devendo ao programa SBLgen determinar um para elas.

Assim como o modelo biológico não informa a localização das reações, também tem poucas informações que possam dizer da adequação ao SBGN, neste caso com a adição de *sinks* e fontes. A ferramenta completa o grafo biológico com a adição desses elementos importantes para a compreensão do modelo biológico estudado, visando à adequação para SBGN-DP.

De modo geral, os resultados alcançados pelo algoritmo na aplicação do programa SBLgen foram bons, porém não pode-se afirmar que os grafos obtiveram as melhores configurações em relação a menor quantidade possível de cruzamentos e de visualização dos dados biológicos, pois os critérios restringem demais o desempenho da compreensão do desenho visto em algumas das soluções exportadas para GML especializado em SBGN-DP. No entanto, devido a complexidade do tema abordado e vasta gama de algoritmos que visam a otimização da visualização de sistemas biológicos, várias linhas de pesquisa podem vir a ser foco de trabalhos futuros, entre elas:

- Aplicação de novos algoritmos de Desenho de grafo neste problema;
- Adotar critérios de redução da área total do desenho, dentre outros critérios;

- Aplicar os algoritmos de desenho de grafos em outras linhas de pesquisa.

Uma possibilidade seria o tratamento hierárquico dos compartimentos dentro do desenho na execução do algoritmo FR. Ou seja, o algoritmo seria aplicado recursivamente, como se cada compartimento fosse um frame. E, ao se concluir o posicionamento dentro de um compartimento, o mesmo seria tratado como um nó dentro do compartimento que o contém. Esta abordagem permite a movimentação dos compartimentos e nós à medida que o desenho está sendo melhorado.

A aplicação de Desenho de Grafos em Biologia Sistêmica é mais um passo rumo a integração de áreas distintas para que o conhecimento científico se torne mais facilmente compreendido entre os pesquisadores. Este trabalho foi realizado para dar suporte aos pesquisadores de Biologia Sistêmica no que diz respeito à Visualização Científica de seus dados.

11 REFERÊNCIAS BIBLIOGRÁFICAS

ONG, Y. S., GOOI, H. B. e CHAN, C. K. **Algorithms for automatic generation of one-line diagrams**, IEE Proc. Gener. Transm. Distrib., Vol. 147, no 5: 292-298, Sept. 2000.

AIELLO, A.; SILVEIRA, R. I. **Trazado de grafos mediante métodos dirigidos por fuerzas: revisión del estado del arte y presentación de algoritmos para grafos donde los vértices son regiones geográficas**. [S.l.], Dezembro 2004.

KIRKPATRICK, S. GELATT, C. D. VECCHI, M. P. **Optimization by simulated annealing**. Science, 220(4598):671 (680), 1983.

KAMADA, T. and KAWAI, S.. **An algorithm for drawing general undirected graphs**. Inform. Process. Lett., 31:7 (15), 1989.

FRUCHTERMAN, T. M.; REINGOLD, E. M. **Graph drawing by force-directed placement**. Software - Practice and Experience, v. 21, n. 11, p. 1129–1164, 1991.

KILLCOYNE, S., CARTER, G. W., SMITH, J., BOYLE, J. (2009). **Cytoscape: a community-based framework for network modeling**. Methods Mol Biol (Clifton, NJ) 563: 219–23

FUNAHASHI, A. et al. (2008). **CellDesigner 3.5: a versatile modeling tool for biochemical networks**, Proc. IEEE, 96, 1254-1265

SOROKIN, A. et al. (2006). **The pathway editor: a tool for managing complex biological networks**, IBM J. Res. Dev., 50, 561-573

JUNKER, B.H. et al. (2006) **VANTED: a system for advanced data analysis and visualization in the context of biological networks**. BMC Bioinformatics, 7, 109.1–109.13.

VILLÉGER, A. C.; PETTIFER, S. R.; KELL, D. B.(2010). **Arcadia: a visualization tool for metabolic pathways**. Bioinformatics (Oxford, England), 26 (11), pp. 1470-1471.

HUCKA, M., HOOPS, S., KEATING, S., LE NOVÈRE, N., SAHLE, S., e WILKINSON, D. (2008) **Systems Biology Markup Language (SBML) Level 2: Structures and Facilities for Model Definitions**. Acessado em 8 de Agosto de 2010. Disponível em <http://dx.doi.org/10.1038/npre.2008.2715.1>

HUCKA, M., FINNEY, A., BORNSTEIN, B. J., KEATING, S. M., SHAPIRO, B. E., MATTHEWS, J., KOVITZ, B. L., SCHILSTRA, M. J., FUNAHASHI, A., DOYLE, J. C., KITANO, H. (2004) **Evolving a lingua franca and associated software infrastructure for computational systems biology: the Systems Biology Markup**

Language (SBML) project. Systems Biology, 1 (1): 41-53. ISSN 1741-2471

HUCKA, M., FINNEY, A., SAURO, H.M., BOLOURI, H., DOYLE, J. C., KITANO, H. (2003). **The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models**. *Bioinformatics*, 19 (4): 524–31. doi:10.1093/bioinformatics/btg015

LE NOVÈRE, N. (2006). **Model storage, exchange and integration**. *BMC neuroscience*, 7 (1): S11 doi:10.1186/1471-2202-7-S1-S11

BORNTEIN, B. J., KEATING, S. M., JOURAKU, A., HUCKA, M. (2008) **LibSBML: An API Library for SBML**. *Bioinformatics*, 24(6):880–881, doi:10.1093/bioinformatics/btn051

LE NOVÈRE, N., HUCKA, M., MI, H., MOODIE, S., SCHREIBER, F., SOROKIN, A. (2009). **The Systems Biology Graphical Notation: Perspective**. *Nat Biotechnol* 27 (8): 735–41. doi:10.1038/nbt.15

KRAWUSE F., UHLENDORF, J., LUBITZ, T., SCHULZ, M., KLIPP, E., LIEBERMEISTER, W.(2010). **Annotation and merging of SBML models with semanticSBML**. *Bioinformatics* 26(3): 421-422

ALVES, R., ANTUNES, F., SALVADOR, A. (2006). **Tools for kinetic modeling of biochemical networks**. *Nature Computational Biology*. 24(6): 667-672

KLIPP, E., LIEBERMEISTER, W., HELBIG, A., KOWALD A., SCHABER, J. (2007), **Systems biology standards - the community speaks**, *Nature Biotechnology* 25, 390 – 391

MOODIE, S., LE NOVÈRE, N. , DEMIR, E., Huaiyu, M., SCHREIER, F. (2010). **Systems Biology Graphical Notation: Process Description language Level 1**. Acessado em 23 de novembro de 2010. Disponível em em <http://hdl.handle.net/10101/npre.2010.3721.2>.

KIENLE, H. **Informação sobre GML**. Acessado em 23 de Novembro de 2010. Disponível em <http://www.programtransformation.org/Transform/GraphModelingLanguage>

WIKIPEDIA. **Graph Drawing (2010)**. Disponível em http://en.wikipedia.org/wiki/Graph_drawing. Acessado em 20 de Janeiro de 2011.

WIKIPEDIA. **Lei de Hook (2011)**. Disponível em http://pt.wikipedia.org/wiki/Lei_de_Hooke. Acessado em 22 de Janeiro de 2011.