UNIVERSIDADE DE SÃO PAULO
FFCLRP–DEPARTAMENTO DE COMPUTAÇÃO E MATEMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

**Suporte à acessibilidade, reprodutibilidade e
transparência em uma plataforma integrada de
análise de dados de expressão gênica**

Wilson Daniel da Silva

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da USP, como parte das exigências para a obtenção do título de Mestre em Ciências, Área: Computação Aplicada.

Ribeirão Preto–SP

2018

Wilson Daniel da Silva

**Suporte à acessibilidade, reprodutibilidade e
transparência em uma plataforma integrada de
análise de dados de expressão gênica**

Versão Revisada

(Versão original encontra-se na unidade que aloja
o Programa de Pós-graduação)

Orientador: Prof. Dr. Cléver Ricardo Guareis de Farias

Ribeirão Preto–SP

2018

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Wilson Daniel da Silva

Suporte à acessibilidade, reprodutibilidade e transparência em uma plataforma integrada de análise de dados de expressão gênica. Ribeirão Preto–SP, 2018.
154p. : il.; 30 cm.

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da USP, como parte das exigências para a obtenção do título de Mestre em Ciências,
Área: Computação Aplicada.

Orientador: Prof. Dr. Cléver Ricardo Guareis de Farias

1. Plataforma integrada de análise de expressão gênica. 2. Acessibilidade, reprodutibilidade e transparência. 3. SemanticSCo Web. 4. API SIBRI.

Wilson Daniel da Silva

**Support for accessibility, reproducibility and
transparency in an integrated gene expression data
analysis platform**

Revised Version

(Original version is in the unit that hosts
the Graduate Program)

Supervisor: Prof. Dr. Cléver Ricardo Guareis de Farias

Ribeirão Preto–SP

2018

Wilson Daniel da Silva

**Suporte à acessibilidade, reprodutibilidade e
transparência em uma plataforma integrada de
análise de dados de expressão gênica**

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto da USP, como parte das exigências para a obtenção do título de Mestre em Ciências, Área: Computação Aplicada.

Trabalho aprovado. Ribeirão Preto–SP, 25 de Outubro de 2018:

**Prof. Dr. Cléver Ricardo Guareis de
Farias**
Orientador

**Prof. Dr. Ricardo Zorzetto Nicoliello
Vêncio**

Profa. Dra. Ellen Francine Barbosa

**Prof. Dr. José Gonçalves Pereira
Filho**

Ribeirão Preto–SP
2018

Agradecimentos

Ao meu orientador, Prof. Dr. Cléver Ricardo Guareis de Farias, por toda a paciência, ensinamentos, e apoio dedicados durante a elaboração desse trabalho. Aos meus pais, que me incentivaram e apoiaram durante todo o período de estudos, sem os quais, não conseguiria ter chegado até aqui. A minha esposa, por todo o amor, carinho e apoio que me fizeram seguir em busca deste objetivo. A todos os amigos que me acompanharam durante este percurso.

*“Não vos conformeis com este mundo,
mas transformai-vos pela renovação do vosso espírito,
para que possais discernir qual é a vontade de Deus,
o que é bom, o que lhe agrada e o que é perfeito”
(Romanos 12, 2)*

Resumo

Modernas técnicas de biologia molecular são utilizadas para quantificar os níveis de expressão gênica de amostras celulares, frequentemente submetidas a diferentes condições experimentais. A genômica funcional é a área da bioinformática responsável pela caracterização do papel funcional dos genes e processos biológicos associados. A análise de expressão gênica requer o uso (integrado) de diversas ferramentas de análise para a obtenção de respostas às questões biológicas de interesse. Uma plataforma de análise pode ser utilizada para facilitar a integração sintática e semântica destas ferramentas. Tal plataforma deve oferecer acessibilidade, reprodutibilidade e transparência, de modo a promover a colaboração e a disseminação da pesquisa científica. A plataforma SemanticSCo foi concebida para prover suporte à composição semântica de serviços de análise de expressão gênica. Contudo, esta plataforma, em sua versão *standalone*, apresenta baixa acessibilidade, além de não prover suporte para a reprodução e o compartilhamento dos *workflows* desenvolvidos por seus usuários. O principal objetivo deste projeto foi ampliar a acessibilidade da plataforma SemanticSCo, bem como introduzir suporte para a reprodução e compartilhamento de *workflows*. Como resultado principal, desenvolvemos uma nova versão da plataforma SemanticSCo, chamada SemanticSCo Web. A plataforma SemanticSCo Web possui uma interface simples e amigável, contribuindo para o desenvolvimento colaborativo de pesquisa científica reproduzível no domínio da genômica funcional. Adicionalmente, desenvolvemos uma API chamada SIBRI para permitir a execução automática, via *framework* Activiti, de serviços RESTful modelados como processos BPMN. O uso desta API facilita a integração de aplicações desenvolvidas na linguagem Java e que fazem uso de serviços RESTful, dado que esta API provê suporte à interação com o usuário em um ambiente web para que o mesmo possa prover valores de parâmetros que serão utilizados na execução de um dado serviço. Desta forma, a API SIBRI simplifica a incorporação de novos serviços na plataforma SemanticSCo Web.

Palavras-chave: Plataforma integrada de análise de expressão gênica; Acessibilidade, reprodutibilidade e transparência; SemanticSCo Web; API SIBRI.

Abstract

Modern molecular biology techniques are used to quantify the levels of gene expression of cellular samples, often submitted to different experimental conditions. Functional genomics is the area of bioinformatics responsible for the characterization of the functional role of genes and associated biological processes. The analysis of gene expression requires the (integrated) use of several analysis tools to obtain answers to a given biological question. An analysis platform can be used to facilitate the syntactic and semantic integration of these tools. Such platform should offer accessibility, reproducibility and transparency in order to promote collaboration and dissemination of scientific research. The SemanticSCo platform was designed to provide support for the semantic composition of gene expression analysis services. However, this platform, in its standalone version, presents low accessibility, and does not provide support for the reproduction and sharing of developed workflows. The main objective of this project was to increase the accessibility of the SemanticSCo platform, as well as to introduce support for the reproduction and sharing of workflows (transparency). As a main result, we have developed a new version of the SemanticSCo platform, called SemanticSCo Web. The SemanticSCo Web platform has a simple and user-friendly interface, contributing to the collaborative development of reproducible scientific research in the functional genomics domain. Additionally, we have developed an API called SIBRI to allow the automatic execution, via Activity framework, of RESTful services modeled as BPMN processes. The use of this API facilitates the integration of applications developed in Java that uses RESTful services, since this API provides support for the interaction with the end user in a web environment so s/he can provide parameters to be used in the execution of a given service. In this way, the API SIBRI simplifies the incorporation of new services in the SemanticSCo Web platform.

Keywords: Integrated gene expression analysis platform; Accessibility, reproducibility and transparency; SemanticSCo Web; SIBRI API.

Lista de figuras

Figura 1 – Interdependência entre as classes do pacote <code>core</code>	63
Figura 2 – Interdependência entre as classes do pacote <code>parameter</code>	65
Figura 3 – Dinâmica de execução da API SIBRI	67
Figura 4 – Representação visual WSDL da especificação do serviço <code>GenericAnalysis</code>	69
Figura 5 – Especificação BPMN para o serviço <code>GenericAnalysis</code>	69
Figura 6 – Representação XML da especificação BPMN das operações definidas no WSDL do serviço <code>GenericAnalysis</code>	70
Figura 7 – Especificação BPMN estendida para uso da API SIBRI na execução das operações do serviço <code>GenericAnalysis</code>	71
Figura 8 – Arquitetura web da plataforma SemanticSCo.	75
Figura 9 – Arquitetura do módulo <code>Composite Service Enactment</code>	78
Figura 10 – Interface para criação de <i>workflows</i> de análise na plataforma SemanticSCo Web.	80
Figura 11 – Interface para pesquisa por experimentos no ArrayExpress	82
Figura 12 – Interface para seleção dos arquivos contidos no experimento que devem ser incluídos no <code>Data Source</code>	83
Figura 13 – Interação do usuário com o ArrayExpress para a obtenção de dados sobre experimentos de expressão gênica.	83
Figura 14 – Persistência de dados utilizando o módulo <code>Composition Manager</code>	86
Figura 15 – Estruturas das tabelas do banco de dados da plataforma SemanticSCo.	87
Figura 16 – Arquitetura do módulo <code>Workflow Execution Manager</code>	89
Figura 17 – Interdependência entre as classes incorporadas à plataforma SemanticSCo Web.	90
Figura 18 – Lista de <i>workflows</i> criados pelo usuário na plataforma SemanticSCo. Em destaque, os botões utilizados para compartilhar um <i>workflow</i>	92
Figura 19 – Seleção de usuários e modo de acesso para um <i>workflow</i> compartilhado para um grupo de usuários.	93
Figura 20 – Listagem dos <i>workflows</i> compartilhados com o usuário <i>logado</i>	94
Figura 21 – Arquitetura do módulo <code>Social Network Integration</code>	95

Figura 22 – Interação do usuário para acesso a plataforma SemanticSCo Web.	96
Figura 23 – Interação do usuário para compartilhamento de <i>workflows</i> na plataforma SemanticSCo Web.	97
Figura 24 – Refatoração do serviço AS2.	99
Figura 25 – Refatoração do serviço AS3.	101
Figura 26 – Refatoração do serviço AS4.	101
Figura 27 – Refatoração do serviço AS5.	102
Figura 28 – Refatoração do serviço AS6.	102
Figura 29 – Etapas para a criação do <i>workflow</i> para a análise diferencial dos dados de Microarray Agilent One-color.	107
Figura 30 – Detalhamento da integração com o repositório ArrayExpress para obtenção dos dados do experimento E-MTAB-2976.	108
Figura 31 – <i>Workflow</i> para a análise diferencial dos dados de Microarray Agilent One-color.	109
Figura 32 – Etapas para a criação do <i>workflow</i> para a geração de um dendrograma a partir da clusterização de dados de Microarray Agilent One-color.	111
Figura 33 – <i>Workflow</i> para a clusterização de dados de Microarray Agilent One-color.	112
Figura 34 – Etapas para a criação do <i>workflow</i> para a análise funcional dos dados de Microarray Agilent One-color.	113
Figura 35 – <i>Workflow</i> para a análise funcional dos dados de Microarray Agilent One-color.	114
Figura 36 – Dendrograma gerado a partir do resultado da clusterização dos 40 genes com maior variação diferencial.	116
Figura 37 – <i>Workflow</i> 4 compartilhado no grupo Study MSC-T1D vs MSC-C do Facebook.	117
Figura 38 – Interface de autenticação da plataforma SemanticSCo.	131
Figura 39 – Interface inicial da plataforma SemanticSCo.	132
Figura 40 – Interface com a listagem de <i>workflows</i> criados pelo usuário da plataforma SemanticSCo.	133
Figura 41 – Interface para criação de <i>workflows</i> na plataforma SemanticSCo.	135
Figura 42 – Interface para acesso aos <i>workflows</i> compartilhados na plataforma SemanticSCo.	136

Figura 43 – Interface para edição dos dados do perfil do usuário na plataforma SemanticSCo.	137
Figura 44 – Etapas para a criação do <i>workflow</i> para a análise de dados de Microarray Agilent One-color.	140
Figura 45 – <i>Workflow</i> para análise de dados de Microarray Agilent One-color. . . .	141
Figura 46 – Etapas para a criação do <i>workflow</i> 1 para a análise de dados de Microarray Affymetrix One-color.	144
Figura 47 – <i>Workflow</i> 1 para análise de dados de Microarray Affymetrix One-color. .	146
Figura 48 – Etapas para a criação do <i>workflow</i> 2 para a análise de dados de Microarray Affymetrix One-color.	148
Figura 49 – <i>Workflow</i> 2 para análise de dados de Microarray Affymetrix One-color. .	149
Figura 50 – Etapas para criação do <i>workflow</i> para a análise de dados de RNA-Seq. .	152
Figura 51 – <i>Workflow</i> para análise de dados de RNA-Seq.	153

Lista de tabelas

Tabela 1 – Comparativo dos requisitos de Acessibilidade, Reprodutibilidade e Transparência.	42
Tabela 2 – Identificação de plataformas em arquivos de dados brutos para experimentos de <i>microarray</i>	45
Tabela 3 – Formatos padrão para arquivos de dados brutos de experimentos de sequenciamento de RNA.	46
Tabela 4 – Exemplos de URLs para a obtenção de dados de experimentos no ArrayExpress.	48
Tabela 5 – Novos serviços criados com base nos adaptadores AS1 a AS6.	104
Tabela 6 – Parâmetros utilizados para a execução do <i>Workflow</i> 1.	110
Tabela 7 – Parâmetros utilizados para a execução do <i>Workflow</i> 3.	112
Tabela 8 – Parâmetros utilizados para a execução do <i>Workflow</i> 4.	114
Tabela 9 – Parâmetros utilizados para a execução do <i>workflow</i> para a análise de dados de Microarray Agilent.	142
Tabela 10 – Parâmetros utilizados para a execução do <i>workflow</i> 1 para análise de dados de Microarray Affymetrix.	147
Tabela 11 – Parâmetros utilizados para a execução do <i>workflow</i> 2 para análise de dados de Microarray Affymetrix.	149
Tabela 12 – Parâmetros utilizados para a execução do <i>workflow</i> para análise de dados de RNA-Seq.	154

Sumário

1	INTRODUÇÃO	27
1.1	Motivação	27
1.2	Objetivo	31
1.3	Metodologia	32
1.4	Estrutura do documento	33
2	FUNDAMENTAÇÃO TEÓRICA	35
2.1	Plataformas integradas de análise	35
2.1.1	Plataformas não orientadas a serviços	36
2.1.2	Plataformas orientadas a serviços	39
2.1.3	Plataforma híbrida	40
2.1.4	Análise comparativa das plataformas	41
2.2	Repositórios públicos de dados genômicos	42
2.2.1	Formato de um experimento	44
2.2.2	Dados de <i>microarray</i>	45
2.2.3	Dados de sequenciamento de RNA	46
2.2.4	Acesso programático ao ArrayExpress	47
2.2.5	Dados do Gene Expression Omnibus (GEO)	49
2.3	Redes sociais	50
2.3.1	Redes sociais de propósito geral	51
2.3.2	Rede social profissional	53
2.3.3	Redes sociais de colaboração científica	53
2.3.4	Integração com redes sociais	55
3	SIMPLE BPMN REST INTERACTION	59
3.1	Suporte à execução de processos BPMN	59
3.2	API SIBRI	61
3.2.1	Visão geral	61
3.2.2	Pacote core	62
3.2.3	Pacotes builder e parameter	65

3.2.4	Dinâmica de execução	66
3.2.5	Extensão da API	68
3.3	Cenário de uso da API	68
3.4	Considerações finais	72
4	PLATAFORMA SEMANTICSCO WEB	73
4.1	Visão geral da plataforma SemanticSCo Web	74
4.1.1	Construção de <i>workflows</i> de análise	74
4.1.2	Arquitetura da plataforma	74
4.2	Suporte à Acessibilidade	77
4.2.1	Composite Service Enactment	77
4.2.2	Gene Expression Repository Data	81
4.3	Suporte à Reprodutibilidade	84
4.3.1	Composition Manager	84
4.3.2	Workflow Execution Manager	88
4.4	Suporte à Transparência	91
4.4.1	Compartilhamento de <i>workflows</i>	91
4.4.2	Social Network Integration	94
4.5	Melhorias Adicionais à Plataforma SemanticSCo	97
4.5.1	Descoberta de novas atividades de análise	97
4.5.2	Refatoração dos serviços de adaptação do repositório GEAS	98
4.6	Considerações finais	103
5	ESTUDO DE CASO	105
5.1	Visão geral	105
5.2	Construção dos workflows	107
5.2.1	Análise diferencial	107
5.2.2	Clusterização	109
5.2.3	Enriquecimento funcional	112
5.3	Resultados	113
5.4	Considerações finais	117
6	CONCLUSÃO	119
6.1	Principais contribuições	119

6.1.1	API SIBRI	119
6.1.2	Plataforma SemanticSCo Web	120
6.2	Discussão	121
6.3	Trabalhos futuros	123
	Referências	125
APÊNDICE A	PLATAFORMA SEMANTICSCO WEB	131
APÊNDICE B	REPRODUÇÃO DO ESTUDO PARA ANÁLISE DE DADOS DE MICROARRAY AGILENT . .	139
APÊNDICE C	REPRODUÇÃO DO ESTUDO PARA ANÁLISE DE DADOS DE MICROARRAY AFFYMETRIX	143
APÊNDICE D	REPRODUÇÃO DO ESTUDO PARA ANÁLISE DE DADOS DE RNA-SEQ ILLUMINA	151

Introdução

A integração de diferentes ferramentas computacionais para a análise de dados na área de genômica funcional representa um grande desafio para os biólogos. Os dados resultantes do processamento de uma dada ferramenta nem sempre são compatíveis com os dados de entrada de outra ferramenta. Plataformas integradas de análise surgiram com a finalidade de simplificar a integração destas diferentes ferramentas em um ambiente controlado pela plataforma. A plataforma SemanticSCo foi concebida para prover suporte à composição semântica de serviços web na área da genômica funcional. Contudo, esta plataforma, em sua versão *standalone*, apresenta baixa usabilidade, além de não fornecer recursos para a reprodução e o compartilhamento dos experimentos criados. Este projeto tem por objetivo aprimorar a usabilidade da plataforma SemanticSCo, provendo recursos de acessibilidade, reprodutibilidade e transparência dos experimentos realizados na plataforma.

O restante deste capítulo está estruturado da seguinte forma: a seção 1.1 apresenta a motivação para a evolução da plataforma SemanticSCo; a seção 1.2 apresenta os objetivos do trabalho; a seção 1.3 apresenta a metodologia utilizada para desenvolvimento deste trabalho; e por fim, a seção 1.4 apresenta a estrutura dos demais capítulos desta dissertação.

1.1 Motivação

Abordagens computacionais têm sido cada vez mais utilizadas como instrumento de pesquisa científica. As pesquisas conduzidas segundo estas abordagens têm produzido um grande volume de dados que necessitam de diferentes técnicas para processamento. e-Science representa um novo paradigma de pesquisa científica no qual o computador torna-se

um componente fundamental do método de investigação científica (BELL; HEY; SZALAY, 2009; HEY; TANSLEY; TOLLE, 2009). Por meio do uso de diferentes abordagens, promove-se a colaboração e a inovação durante todo o ciclo de pesquisa científica. Diversas áreas da ciência têm se beneficiado deste novo paradigma, incluindo física, química, astronomia, engenharia aeroespacial, bioinformática, entre outras.

A bioinformática caracteriza-se pelo uso de algoritmos e métodos estatísticos na compreensão e resolução de problemas biológicos (ATTWOOD et al., 2011). A necessidade de manipular um grande volume de dados heterogêneos, gerados por diferentes técnicas de biologia molecular, representa um dos principais desafios da bioinformática atualmente. Dentro da bioinformática, a área de pesquisa de genômica funcional estuda a função dos genes por meio da medição dos seus níveis de expressão (BUTTE, 2002). A expressão de um gene refere-se ao processo biológico no qual informações codificadas são sintetizadas em produtos gênicos funcionais (proteínas ou RNAs). O nível de expressão de um gene refere-se à quantificação de moléculas funcionais a que o gene dá origem, podendo apresentar diferentes níveis de expressão dependendo da célula, tecido ou organismo.

Diferentes técnicas, tais como Microarrays de DNA (BUTTE, 2002; SLONIM; YANAI, 2009) e RNA-Seq (WIT et al., 2012; WOLF, 2013), podem ser utilizadas para a obtenção de dados de expressão gênica. Uma vez obtidos estes dados, ferramentas computacionais são utilizadas para a realização de diferentes atividades de análise, tais como normalização, identificação de genes diferencialmente expressos, construção de redes regulatórias, clusterização de perfis de expressão e análise funcional.

A realização de atividades de análises em dados de expressão gênica requer o uso de diferentes ferramentas de análise e recursos computacionais. Estas ferramentas e recursos precisam ser combinados, de forma manual ou por meio de uma plataforma integrada de suporte, para a resolução do problema de interesse. A integração manual não é uma tarefa trivial para usuários sem conhecimentos técnicos em computação (biologistas). Frequentemente, o usuário precisa adaptar a saída de dados de uma ferramenta com a entrada de dados esperada por outra ferramenta. Esta atividade requer conhecimentos em linguagens de programação e *scripts*, bem como a compreensão de aspectos sintáticos, semânticos e de execução de cada ferramenta. A utilização de serviços web têm sido uma opção cada vez mais utilizada neste contexto, pois facilita a integração de ferramentas construídas a partir de diferentes tecnologias na forma de serviços (BLEDA et al., 2012;

GUARDIA, 2016; GUARDIA et al., 2015). Tais serviços estão disponíveis de forma pública, podendo ser combinados manualmente ou de forma semiautomática por plataformas de suporte para criar *workflows* de análise.

Plataformas integradas de análise vêm sendo desenvolvidas para prover aos biólogos um ambiente amigável para a realização de suas análises (BOLCHINI et al., 2009). A integração de ferramentas neste tipo de plataforma é facilitada, pois, normalmente, esta tarefa não requer que o usuário possua conhecimentos avançados em linguagens de programação e *scripts*. Além disso, a combinação de ferramentas geralmente não exige do usuário adequações de dados, facilitando o processo de integração. Por outro lado, o uso de uma plataforma integrada frequentemente restringe as atividades de análises às ferramentas e recursos disponíveis em uma dada plataforma.

Fundamentalmente, uma plataforma integrada de análise deve prover suporte para a criação e a execução de um *workflow* de análise. Contudo, a simples disponibilidade desta funcionalidade não necessariamente contribui para uma alta usabilidade desta plataforma. Funcionalidades adicionais, visando promover a colaboração e a disseminação da pesquisa científica, tais como a acessibilidade, reprodutibilidade e transparência, são igualmente desejáveis.

O termo acessibilidade está normalmente associado a usuários de sistemas computacionais com algum tipo de limitação (ISO; IEC; IEEE, 2010). Contudo, a acessibilidade também descreve o quão simples é a utilização de um produto, sistema ou serviço por usuários com diferentes capacitações tecnológicas. Uma plataforma integrada de análise pode prover diferentes níveis de acessibilidade. Uma plataforma com alta acessibilidade permite a utilização de suas ferramentas por usuários sem conhecimentos computacionais avançados. Este tipo de plataforma facilita o uso e o aprendizado das funcionalidades contidas na plataforma. Por outro lado, uma plataforma com baixa acessibilidade exige de seus usuário conhecimentos técnicos avançados, o que pode restringir seu uso para certos tipos de usuários.

Reprodutibilidade é um aspecto importante da pesquisa científica. O conceito de pesquisa computacional reproduzível representa a capacidade de terceiros repetirem as atividades de análise realizadas em um determinado estudo de modo a obter os resultados computacionais iniciais (MESIROV, 2010). Um sistema que suporte este conceito é

chamado de Sistema de Pesquisa Reproduzível (SPR). Ao permitir que terceiros possam reproduzir um experimento, uma plataforma provê recursos para que o experimento original possa ser novamente executado e os novos resultados obtidos possam ser comparados com os resultados originais.

Finalmente, a existência de recursos que facilitam o compartilhamento e a divulgação dos resultados experimentais obtidos promovem a colaboração científica. Adicionalmente, tais facilidades aumentam a confiança nos resultados. Transparência é a capacidade de comunicar e compartilhar os resultados obtidos por um experimento de uma maneira compreensível (GOECKS et al., 2010). Ao prover a transparência nos experimentos criados, uma plataforma tem por objetivo promover a colaboração e a divulgação dos resultados dos experimentos entre os pesquisadores. Além disso, tal característica permite que novos estudos possam ser realizados com base em um experimento compartilhado.

Semantic Services Composition (SemanticSCo) (GUARDIA et al., 2017; GUARDIA, 2016) é uma plataforma de análise de expressão gênica integrada que suporta a composição semiautomática de serviços web semânticos. Serviços web semânticos são serviços web cujas descrições de serviço são anotadas semanticamente com termos provenientes de uma ontologia (W3C, 2007). Tais anotações têm por objetivo atribuir um significado não-ambíguo a um determinado serviço.

A plataforma SemanticSCo permite que seus usuários criem e executem *workflows* de análise a partir da composição semântica de um conjunto de serviços RESTful. SemanticSCo faz uso dos serviços disponibilizados pelo repositório Gene Expression Analysis Services (GEAS) (GUARDIA et al., 2015) para a execução das diferentes atividades de análise. Detalhes técnicos da composição e execução dos serviços de análise são escondidos de seus usuários, uma vez que a composição dos serviços ocorre não apenas levando-se em consideração aspectos sintáticos, mas principalmente aspectos semânticos.

Os usuários da plataforma SemanticSCo podem ser classificados em dois tipos: biólogos e bioinformatas. Usuários biólogos possuem um conhecimento biológico profundo sobre os processos de análise de dados de expressão gênica, enquanto que bioinformatas possuem um conhecimento computacional avançado, que lhes permitem desenvolver novos serviços de análise de expressão gênica.

A etapa de criação de *workflows* de análise é um processo iterativo, normalmente

realizado por usuários biólogos. Este processo iterativo de criação do *workflow* atende às necessidades dos estudos de expressão gênica, que possuem natureza exploratória. SemanticSCo sugere aos biólogos os serviços de análise disponíveis com base no conjunto de dados utilizados como entrada e/ou atividades de análise desejadas. A qualquer momento é permitido que o usuário modifique os dados, parâmetros ou até mesmo serviços utilizados em seu *workflow*.

Novos serviços podem ser incorporados à plataforma. Ferramentas de análise são desenvolvidas e encapsuladas em serviços web semânticos por usuários bioinformatas para que possam ser publicadas na plataforma. Ao criar novos serviços para a plataforma, os bioinformatas também são responsáveis por especificar o comportamento local dos serviços, definindo a ordem em que as operações podem ser executadas. Quando um serviço é publicado na plataforma, este fica disponível para os usuários biólogos durante o processo de criação de um *workflow* de análise.

Apesar de todos os recursos fornecidos pela plataforma SemanticSCo, algumas limitações relacionadas ao suporte à acessibilidade, reprodutibilidade e transparência podem ser observadas. A plataforma possui acessibilidade limitada, a qual restringe sua utilização em diferentes tipos de dispositivos computacionais. Em sua versão inicial, a criação e a execução de *workflows* está disponível por meio de uma interface gráfica *desktop*, que necessita ser executada localmente no computador do usuário. A plataforma também não suporta a reprodução de experimentos, pois não permite que um *workflow* construído possa ser posteriormente recuperado e reproduzido. Finalmente, a plataforma também não possui recursos que promovam a transparência dos experimentos realizados, dado que não possui recursos para o compartilhamento dos *workflows* criados entre os usuários da plataforma.

1.2 Objetivo

O objetivo geral deste trabalho é propor um conjunto de extensões à plataforma SemanticSCo para ampliar sua acessibilidade e prover suporte à pesquisa científica colaborativa. Neste sentido, este trabalho possui os seguintes objetivos específicos: i) ampliar o suporte à acessibilidade por meio do desenvolvimento de uma nova interface web, simples e intuitiva; ii) prover o suporte à reprodutibilidade por meio do desenvolvimento de mecanismos que

permitam a persistência e posterior reprodução dos experimentos criados na plataforma; e, por fim, iii) prover o suporte à transparência por meio da integração da plataforma com redes sociais *online* de ampla utilização, com o intuito de compartilhar nestas redes os *workflows* construídos na plataforma.

A utilização de tecnologias web recentes na implementação de uma nova versão da plataforma SemanticSCo permitirá seu uso em diferentes dispositivos, ampliando sua aceitação pelos usuários. A inclusão do suporte à reprodutibilidade deverá contemplar, além da reprodução de um experimento construído na plataforma, a integração com repositórios de dados de genômica funcional amplamente utilizados pela comunidade científica. Esta integração visa recuperar os dados de experimentos armazenados nestes repositórios, e disponibilizá-los na plataforma para a realização de análises. O suporte à transparência por meio do compartilhamento dos *workflows* desenvolvidos em redes sociais visa favorecer a colaboração científica entre biólogos e bioinformatas.

1.3 Metodologia

Inicialmente foi realizada uma revisão bibliográfica sobre ferramentas e plataformas integradas de análise, técnicas de expressão gênica e repositórios de dados genômicos. Adicionalmente, foi realizado um levantamento dos principais recursos oferecidos por diferentes plataformas integradas de análise, além dos recursos para a integração de dados genômicos. Estas atividades tiveram por objetivo construir uma base sólida de conhecimento para melhor compreender os desafios da análise de expressão gênica, bem como para projetar a evolução da plataforma SemanticSCo.

A seguir, realizamos um estudo sobre a integração de aplicações com redes sociais *online*. Este estudo objetivou uma visão geral sobre recursos disponíveis por diferentes redes, assim como as APIs disponíveis para utilização. Foram avaliados a utilização de APIs JavaScript, além da utilização do *framework* de autenticação chamado OAuth 2.0 para a integração com as redes sociais Facebook, LinkedIn e Google+.

Após completar os estudos de fundamentação teórica do trabalho, foi realizado um estudo detalhado da plataforma SemanticSCo. Este estudo objetivou obter uma visão geral do funcionamento desta plataforma, bem como do funcionamento dos mecanismos

de inclusão de novos serviços à plataforma, descoberta e composição de serviços para a criação e a execução de um *workflow* (execução de serviços).

Em seguida, iniciou-se o projeto e a implementação dos recursos para prover suporte aos requisitos de acessibilidade, reprodutibilidade e transparência. A linguagem de programação Java foi utilizada como base para o desenvolvimento da nova interface de acesso, persistência de dados, e integração com repositórios de dados genômicos e redes sociais. Adicionalmente, uma Application Programming Interface (API) chamada Simple BPMN REST Interaction (SIBRI) foi criada para simplificar o processo de inclusão de novos serviços à plataforma. Por fim, os serviços adaptadores utilizados pela plataforma foram refatorados com a finalidade de simplificar sua utilização.

Finalmente, de modo a validar o funcionamento da plataforma, foi reproduzido parte de um estudo de análise de expressão gênica (LIMA et al., 2016) com o intuito de demonstrar as principais funcionalidades implementadas na plataforma para permitir a colaboração científica. Estudos adicionais também foram realizados para testar a plataforma SemanticSCo Web com base nos experimentos realizados na versão original da plataforma (GUARDIA, 2016).

1.4 Estrutura do documento

Os demais capítulos deste documento estão estruturados da forma apresentada a seguir: o capítulo 2 apresenta a fundamentação teórica e técnica para compreensão dos principais conceitos utilizados neste trabalho; o capítulo 3 descreve a Simple BPMN REST Interaction (SIBRI), uma API genérica e customizável desenvolvida para prover suporte à execução de processos Business Process Model and Notation (BPMN); o capítulo 4 descreve com detalhes técnicos a evolução e os novos recursos incluídos à plataforma SemanticSCo; o capítulo 5 apresenta um estudo de caso realizado na plataforma SemanticSco Web de modo a validar o seu funcionamento; por fim, o capítulo 6 apresenta a conclusão do trabalho, assim como uma discussão sobre trabalhos futuros envolvendo a plataforma SemanticSCo.

Fundamentação Teórica

Este capítulo apresenta uma visão geral acerca de três conjuntos de conceitos fundamentais para a compreensão do contexto e das soluções tecnológicas de integração utilizadas no trabalho. Inicialmente, diferentes plataformas integradas de análise são introduzidas, destacando-se os aspectos de acessibilidade, reprodutibilidade e transparência de cada plataforma. Na sequência, repositórios públicos de dados genômicos são apresentados, destacando-se aspectos de armazenamento e recuperação de dados de um experimento. Finalmente, uma visão geral sobre redes sociais *online* é apresentada, destacando-se como tais redes podem ser utilizadas como mecanismo de colaboração científica.

O capítulo está estruturado da seguinte forma: a seção 2.1 apresenta uma visão geral sobre plataformas integradas de análise; a seção 2.2 apresenta conceitos básicos sobre repositórios públicos de dados genômicos, assim como uma visão geral da integração via serviços web para a recuperação de dados contidos nestes repositórios; e, por fim, a seção 2.3 introduz as redes sociais *online* e os principais mecanismos para a integração com estas redes.

2.1 Plataformas integradas de análise

Plataformas integradas de análise permitem a construção e a execução de *workflows* de análise a partir de um conjunto de ferramentas e recursos disponíveis na plataforma. Com o advento e a popularização dos serviços web, plataformas integradas de análise passaram a incorporar serviços web como recursos adicionais. A inclusão destes recursos amplia as funcionalidades disponíveis em uma plataforma, aumentando sua capacidade de análise.

Neste sentido, sob a ótica da orientação a serviços, plataformas integradas de análise podem ser classificadas em: plataformas não orientadas a serviços, plataformas orientadas a serviços e plataformas híbridas.

Plataformas não orientadas a serviços proveem suporte à integração de diferentes tipos de ferramentas e recursos na criação de um *workflow* de análise. Contudo, estas ferramentas e recursos são normalmente acessados e executados localmente na máquina do próprio usuário ou no servidor da plataforma. Já plataformas orientadas a serviços utilizam exclusivamente serviços web na criação de um *workflow* de análise. Tais serviços podem ser integrados e executados com diferentes níveis de transparência para seus usuários. Cada serviço é executado (remotamente) em seu respectivo provedor. Por fim, as plataformas híbridas são aquelas que combinam os recursos utilizados por plataformas não orientadas a serviços com os recursos de plataformas orientadas a serviços. Nestas plataformas, recursos internos e externos são combinados na criação de um *workflow* de análise.

Diferentes plataformas de análise genômica estão atualmente disponíveis, tais como Galaxy (AFGAN et al., 2016), RNA-Seq Analysis Pipeline (RAP) (D'ANTONIO et al., 2015), TRanscriptome User-Friendly Analysis (TRUFA) (KORNOBIS et al., 2015), Bio-jETI (LAMPRECHT; MARGARIA; STEFFEN, 2009), Babelomics (ALONSO et al., 2015) e Taverna (WOLSTENCROFT et al., 2013).

2.1.1 Plataformas não orientadas a serviços

Galaxy

Galaxy (GOECKS et al., 2010; AFGAN et al., 2016) é uma plataforma web não orientada a serviços que permite a realização de análise de experimentos genômicos. Esta plataforma, implementada na linguagem Python, provê recursos para a criação, a reprodução e a divulgação dos experimentos criados na plataforma. Cada atividade de análise em um experimento é representada por meio de uma sequência de passos em um histórico de execução. Este histórico também pode ser representado graficamente por um *workflow* de análise. Este *workflow* apresenta as interações entre cada uma das atividades de análise de um experimento. A plataforma Galaxy possui vários recursos que facilitam o compartilhamento dos experimentos criados na plataforma. Durante a criação de um experimento,

informações adicionais podem ser incluídas no experimento com a finalidade de indicar quais foram as intenções do usuário. Estas informações podem ser incluídas nas atividades de análise e ficam disponíveis posteriormente, quando o experimento for recuperado. A plataforma ainda possui uma área específica onde os experimentos criados podem ser compartilhados com outros usuários, facilitando a divulgação destes experimentos. Uma vez que um experimento é visualizado por outros usuários, este experimento pode ser acessado e reproduzido da maneira em que foi originalmente concebido.

Os usuários da plataforma Galaxy podem enviar arquivos de dados para seus experimentos diretamente pelo navegador web. A plataforma ainda provê mecanismos de integração que permitem a obtenção de dados diretamente de repositórios públicos de dados genômicos como o NCBI e o University of California Santa Cruz (UCSC) (UCSC, 2002). Esta integração simplifica o processo de criação dos experimentos de análise dentro da plataforma. Galaxy ainda provê o Galaxy Pages (GOECKS et al., 2010), uma ferramenta para confecção de documentos web interativos com a finalidade de descrever os experimentos realizados na plataforma. O Galaxy permite que um experimento desenvolvido na plataforma seja incorporado e executado a partir de um documento criado com o Galaxy Pages.

RAP

RNA-Seq Analysis Pipeline (RAP) (D'ANTONIO et al., 2015) é uma plataforma web não orientada a serviços para a criação de *workflows* de análise de experimentos RNA-Seq. A plataforma permite o compartilhamento do resultado dos experimentos apenas entre os seus usuários. As ferramentas disponíveis na plataforma são resultantes da utilização de ferramentas *open-source* e *scripts* próprios para a análise de experimentos RNA-Seq. A plataforma RAP proporciona um ambiente de computação em nuvem para a execução dos *workflows* de análise de forma transparente para seus usuários. A plataforma é implementada utilizando as linguagens Hypertext Pre-processor (PHP) e jQuery, além dos padrões HTML 5 e Cascading Style Sheets (CSS) 3.

O ambiente provido pelo RAP permite a análise de grandes conjuntos de dados, comuns a experimentos de RNA-Seq. Para isso, o RAP provê três mecanismos para realizar o *upload* dos dados: Web Upload, Web Link e o *plugin* Dropbox Chooser (DROPBOX,

2018). O Web Upload consiste na seleção de um arquivo local via navegador web. Neste caso, o tamanho máximo para um arquivo é de 12Gb para sistemas operacionais de 64 bits e 2Gb para sistemas operacionais de 32 bits. Estas limitações podem ser contornadas com o uso dos demais métodos. O Web Link permite o download de arquivos informados via Uniform Resource Locator (URL) para a plataforma RAP. Já o *plugin* Dropbox Chooser permite que arquivos mantidos dentro do Dropbox possam ser recuperados por meio de um serviço de integração.

O suporte a grandes conjuntos de dados providos pelo RAP permite a utilização de arquivos contendo dados brutos de sequenciamento. Após o envio dos arquivos para a plataforma, estes dados são identificados e ficam disponíveis para a utilização em *workflows* de análise de experimentos RNA-Seq criados na plataforma. Ao criar um *workflow* de análise um ou mais arquivos de dados podem ser utilizados. A plataforma permite a definição de diferentes atividades de análise sobre os dados associados aos *workflows*. Ao concluir a construção do *workflow* o usuário pode solicitar a sua execução.

TRUFA

TRanscriptome User-Friendly Analysis (TRUFA) (KORNOBIS et al., 2015) é uma plataforma web não orientada a serviços para a análise de experimentos de RNA-Seq. Essencialmente, a plataforma encapsula diversas ferramentas para suporte às atividades de análise. Estas ferramentas são executadas em uma infraestrutura de computação de alto desempenho, permitindo que os experimentos de análise sejam executados em um tempo relativamente curto. A plataforma é implementada utilizando as linguagens Javascript, Python e Bash.

A plataforma TRUFA provê uma interface simples para acesso às ferramentas de análise contidas na plataforma. O resultado do processamento dos experimentos realizados podem ser posteriormente recuperados. A plataforma não provê recursos para compartilhar com outros usuários estes dados ou mesmos os experimentos construídos na plataforma.

O envio de dados de experimentos de RNA-Seq para a plataforma TRUFA é realizado unicamente utilizando os recursos próprios do navegador web. TRUFA permite o envio de grandes volumes de dados de experimentos de RNA-Seq, não limitando o tamanho máximo para estes arquivos. Na plataforma TRUFA a realização de cada experimento de

análise consiste na seleção prévia de todos os parâmetros e dados de entrada necessários para a execução de um experimento. Uma vez que todos os parâmetros e dados foram informados, o usuário solicita a execução do experimento de análise na plataforma.

2.1.2 Plataformas orientadas a serviços

Bio-jETI

Bio-jETI (MARGARIA; KUBCZAK; STEFFEN, 2008; LAMPRECHT; MARGARIA; STEFFEN, 2009) é uma plataforma orientada a serviços que provê suporte à composição de experimentos em bioinformática. A plataforma permite a utilização de serviços RESTful e SOAP ao compor um experimento. A composição destes serviços ocorre por meio da construção de *workflows* de análise. A criação e a execução de *workflows* está disponível por meio de uma interface gráfica *desktop*.

A plataforma Bio-jETI utiliza o ambiente Java Application Building Center (jABC), para a modelagem dos *workflows*, e o serviço de integração Java Electronic Tool Integration (jETI), para a execução remota dos serviços. Esta arquitetura permite uma grande flexibilidade na composição de serviços de análise dentro da plataforma, porém exige a instalação local destas ferramentas para uso da plataforma.

Bio-jETI permite que os *workflows* desenvolvidos na plataforma possam ser gravados e posteriormente recuperados a fim de que possam ser reproduzidos por seus usuários. Apesar desta possibilidade, a plataforma não provê mecanismos para que estes *workflows* possam ser compartilhados com outros usuários da plataforma.

Babelomics

Babelomics (ALONSO et al., 2015) é uma plataforma web orientada a serviços que provê um conjunto de ferramentas para a análise funcional de experimentos de expressão gênica e outros dados genômicos. De modo a prover tal suporte, Babelomics encapsula as funcionalidades providas pelo repositório de serviços CellBase (BLEDA et al., 2012). A plataforma é implementada utilizando a linguagem Java, com alguns algoritmos escritos em C/C++, além do padrão Hypertext Markup Language (HTML) 5.

A plataforma Babelomics permite o acesso a diversas ferramentas de análise por meio de uma interface simples e intuitiva. Os experimentos realizados podem ser persistidos para posterior recuperação. Porém, a plataforma não provê recursos para compartilhar estes experimentos com outros usuários, com a finalidade de reproduzir os seus resultados.

O ambiente web provido pela plataforma Babelomics permite o acesso às ferramentas de análise para usuários registrados ou não na plataforma. Um usuário registrado pode manter suas análises ao término de sua sessão, enquanto que um usuário não registrado tem seus dados excluídos ao término da sessão. Para usuários registrados, o limite de armazenamento é de 10 Gb. Os dados resultantes da execução de um experimento podem ser utilizados como entrada para um novo experimento criado na plataforma. Babelomics também permite a instalação de uma versão local da plataforma. Neste caso, o limite máximo utilizado para o armazenamento de experimentos pode ser alterado.

2.1.3 Plataforma híbrida

Taverna (WOLSTENCROFT et al., 2013) é uma plataforma desenvolvida para projetar e executar *workflows* científicos no domínio da bioinformática. A plataforma proporciona uma grande flexibilidade durante a criação de seus *workflows*, permitindo a utilização de serviços RESTful e SOAP. Adicionalmente, Taverna ainda permite a interação de *scripts* locais, BioMart *data warehouses*, serviços de grade e nuvem, *scripts* R e de linha de comando distribuídos. Neste sentido, Taverna é considerada uma plataforma híbrida, pois combina recursos de plataformas orientadas a serviços com recursos de plataformas não orientadas a serviços.

A flexibilidade proporcionada pela plataforma aumenta o poder de sua utilização, uma vez que permite a construção e a execução de *workflows* para os mais variados tipos de experimentos e cenários de análise. Porém, tal flexibilidade tem como desvantagem uma maior complexidade durante a criação dos *workflows* de análise na plataforma.

A plataforma Taverna possui diversas versões, tais como Taverna Workbench, Taverna Server, Taverna Player e Taverna Command-Line. Taverna Workbench é a versão principal da plataforma e provê suporte à criação, execução, e disponibilização de *workflows* de análise para outros usuários. Esta versão é implementada na linguagem Java e requer uma instalação local para sua execução. Taverna Workbench utiliza a rede

social myExperiment para o compartilhamento dos *workflows* construídos na plataforma. A integração da plataforma Taverna com o myExperiment permite a publicação direta de um *workflow* Taverna nesta rede social de colaboração científica.

Taverna Server, Taverna Player e Taverna Command-Line não suportam a construção de *workflows*, permitindo apenas a sua execução. O Taverna Server permite que os *workflows* possam ser executados em uma infraestrutura computacional remota, ou como componentes de outras plataformas, como, por exemplo, a plataforma Galaxy. O Taverna Player provê uma interface para o Taverna Server e permite que os *workflows* sejam executados a partir de navegadores web. Por fim, o Taverna Command-Line permite que os *workflows* possam ser executados a partir de uma interface linha de comando.

2.1.4 Análise comparativa das plataformas

As plataformas Galaxy, RAP, TRUFA e Babelomics são exemplos de plataformas integradas de análise com alta acessibilidade. Estas plataformas abstraem detalhes técnicos da composição e execução das ferramentas e serviços, favorecendo sua utilização por usuários sem conhecimentos computacionais avançados. Estas plataformas também proveem ferramentas que permitem a realização de experimentos por meio da web, o que contribui para a acessibilidade da plataforma. Por outro lado, as plataformas Bio-jETI e Taverna são exemplos de plataformas com baixa acessibilidade. Estas plataformas exigem por parte dos seus usuários o conhecimento de diferentes aspectos sintáticos e semânticos das ferramentas (serviços) sendo utilizados na criação de seus *workflows*. Estas plataformas também não possuem uma interface web para a composição de seus experimentos, exigindo a instalação de uma aplicação local (*desktop*) para acesso ao conjunto de ferramentas para a criação de seus *workflows*.

As plataformas Galaxy, Bio-jETI e Taverna suportam a reprodutibilidade de seus experimentos. A plataforma Galaxy provê reprodutibilidade por meio da coleta de metadados gerados automaticamente durante a criação de um experimento, bem como pela coleta de comentários sobre os experimentos inseridos por seus usuários. Os comentários ajudam no entendimento posterior do experimento, de modo que o mesmo possa ser reproduzido da maneira em que foi inicialmente concebido. As plataformas Bio-jETI e Taverna compõe seus experimentos por meio da criação de *workflows* que, posteriormente,

podem ser reproduzidos por terceiros. As plataformas RAP, TRUFA e Babelomics não proveem suporte a reprodutibilidade.

As plataformas Galaxy, RAP e Taverna oferecem suporte à transparência dos experimentos criados. O Galaxy provê transparência por meio do compartilhamento de itens e de uma estrutura web compartilhada. Adicionalmente, esta plataforma utiliza o Galaxy Pages para a divulgação de informações sobre a análise computacional realizada. RAP suporta apenas o compartilhamento de informações sobre experimentos desenvolvidos entre os usuários da própria plataforma. A plataforma Taverna utiliza-se do myExperiment para publicar os experimentos criados. TRUFA, Bio-jETI e Babelomics não suportam a transparência por não proverem recursos para a publicação dos seus experimentos.

A Tabela 1 sumariza o suporte à acessibilidade, reprodutibilidade e transparência em cada uma das plataformas integradas de análise apresentadas anteriormente.

Tabela 1 – Comparativo dos requisitos de Acessibilidade, Reprodutibilidade e Transparência.

Plataformas	Acessibilidade	Reprodutibilidade	Transparência
Galaxy	Alta	Suporta	Suporta
RAP	Alta	Não suporta	Suporta parcialmente
TRUFA	Alta	Não suporta	Não suporta
Bio-jETI	Baixa	Suporta	Não suporta
Babelomics	Alta	Não suporta	Não suporta
Taverna	Baixa	Suporta	Suporta

2.2 Repositórios públicos de dados genômicos

ArrayExpress (EMBL-EBI, 2015; KOLESNIKOV et al., 2015) e Gene Expression Omnibus (GEO) (NCBI, 2016; BARRETT et al., 2007) são repositórios de dados de genômica funcional amplamente utilizados pela comunidade científica. Estes repositórios foram desenvolvidos para prover suporte à pesquisa reproduzível, permitindo acesso aos dados dos experimentos neles publicados. ArrayExpress e GEO permitem a submissão de experimentos seguindo as orientações MIAME (BRAZMA et al., 2001; BRAZMA, 2009) e MINSEQE (FGED, 2008). Os padrões MIAME e MINSEQE estabelecem modelos conceituais para representação de forma não ambígua de dados de experimentos de *microarray* e sequenciamento, respectivamente.

ArrayExpress é hospedado pelo European Bioinformatics Institute (EMBL-EBI). Dados brutos (*raw*) de experimentos de expressão gênica são arquivados no European Nucleotide Archive (ENA), enquanto a descrição do experimento e eventuais dados processados (*processed*) são armazenados diretamente no ArrayExpress. Os experimentos no repositório ArrayExpress são submetidos no formato Microarray Gene Expression Tabular (MAGE-TAB) (FGED, 2011). O MAGE-TAB especifica um formato tabular para representação de um experimento utilizando arquivos texto.

O ArrayExpress permite a submissão de dados de genômica funcional para experimentos de *microarray one-color* e *two-color*, além de experimentos de sequenciamento de RNA. A submissão de experimentos é realizada por meio da ferramenta Annotare 2.0 (EMBL-EBI, 2012).

Durante a submissão de um experimento no repositório, o responsável pelo experimento descreve todas as informações relacionadas ao experimento (metadados), além de incluir arquivos de dados brutos e/ou processados. Os metadados descrevem o experimento fornecendo diferentes informações, tais como objetivo, materiais utilizados, características destes materiais, entre outras. Os arquivos de dados brutos contêm dados obtidos diretamente a partir de um *scanner* de *microarray* ou a partir de uma máquina de sequenciamento de RNA. Os arquivos de dados processados contêm dados gerados a partir de arquivos de dados brutos, os quais passaram por algum tipo de tratamento, como, por exemplo, normalização, antes de serem disponibilizados. No repositório são permitidos arquivos de dados brutos e processados tanto para experimentos de *microarray* quanto para experimentos de sequenciamento de RNA.

O ArrayExpress atribui um identificador único de acesso a todo experimento submetido ao repositório. Este identificador é utilizado para recuperar dados sobre um determinado experimento. O identificador de acesso pode assumir os formatos E-XXXX-n ou A-XXXX-n, onde X representa uma letra e n representa um número. O formato E-XXXX-n é utilizado para representar experimentos, enquanto o formato A-XXXX-n representa *array designs*¹. O conjunto de quatro letras XXXX contido em ambos os formatos é utilizado para representar a origem do experimento ou *array design*. Por exemplo, AFFY para Affymetrix, AGIL para Agilent, MTAB para Annotare, GEOD para NCBI Gene Expression Omnibus (GEO), entre outros. Uma lista completa dos tipos de origens de experimentos pode ser

¹ *Array designs* fornecem informações sobre sequências localizadas em *spots* de *microarrays*.

obtida em (EMBL-EBI, 2015).

2.2.1 Formato de um experimento

O ArrayExpress utiliza o formato Microarray Gene Expression Tabular (MAGE-TAB) (FGED, 2011) para representar os dados de experimentos de *microarray one-color*, *two-color* e experimentos de sequenciamento de RNA. O formato MAGE-TAB segue as orientações MIAME para a representação de dados de experimentos de *microarray*. Este formato representa uma simplificação do formato Microarray Gene Expression Markup Language (MAGE-ML). MAGE-TAB define a representação dos dados de experimentos por meio do seguinte conjunto de arquivos:

- **Investigation Description Format (IDF)**, o qual fornece informações gerais sobre o experimento, como, por exemplo, nome, descrição, informações de contato e referências bibliográficas. Estas informações são organizadas em um arquivo de texto simples delimitado por tabulações;
- **Sample and Data Relationship Format (SDRF)**, o qual contém a descrição do relacionamento entre amostras, *arrays*, dados e outros objetos utilizados ou produzidos no experimento. Um experimento pode possuir um ou mais arquivos SDRF. Estes arquivos possuem o formato de texto simples delimitado por tabulações;
- **Array Design Format (ADF)**, o qual contém informações sobre as sequências localizadas em cada *spot* de um *microarray*. O arquivo ADF possui o formato de texto simples delimitado por tabulações;
- **Arquivos de dados**, os quais são compostos por um ou mais arquivos contendo os dados brutos e/ou processados de um experimento.

Os arquivos de dados IDF, SDRF e ADF são construídos pela plataforma Annotare a partir das informações fornecidas durante a submissão de um experimento.

No ArrayExpress os arquivos de dados brutos e processados são disponibilizados em arquivos compactados, cuja extensão é do tipo `.raw.1.zip`, para arquivos contendo dados brutos, ou, `.processed.1.zip`, para arquivos contendo dados processados. Um experimento pode possuir mais de um conjunto de arquivos compactados contendo os dados

do experimento. Neste caso, o número 1 contido na extensão do arquivo é incrementado para representar diferentes pacotes de arquivos de dados, por exemplo, `.raw.2.zip`, `.raw.3.zip`, `.processed.2.zip`, etc. Uma descrição contendo os nomes de arquivos de dados contidos em cada pacote (arquivo compactado) pode ser obtida realizando a leitura do arquivo SDRF do experimento.

2.2.2 Dados de *microarray*

Em um experimento de *microarray*, dados brutos são gerados por um *software* de *scanner* de *microarray*. Estes dados são armazenados em arquivos nativos da plataforma sem nenhuma alteração após sua leitura. Existem diferentes formatos de dados de *microarray* dependendo da plataforma utilizada para obtenção. O ArrayExpress reconhece dados brutos gerados por diferentes plataformas, tais como Affymetrix, Agilent, Illumina, GenePix, e NimbleScan. O ArrayExpress também suporta o formato *Generic*, que é utilizado para o envio de dados sem a definição de uma plataforma específica.

O ArrayExpress reconhece o tipo de arquivo enviado para a plataforma a partir dos títulos das colunas nos cabeçalhos dos arquivos de dados (EMBL-EBI, 2012). A Tabela 2 sumariza as características básicas utilizadas pelo ArrayExpress para determinar arquivos de dados brutos submetidos para o repositório.

Tabela 2 – Identificação de plataformas em arquivos de dados brutos para experimentos de *microarray*.

Plataforma	Informação verificada no arquivo
Affymetrix	O sistema de submissão reconhece esta plataforma por meio das extensões <code>.CEL</code> e <code>.EXP</code> . Os formatos GDAC e GCOS/XDA são suportados.
Agilent	Cabeçalhos <code>Row</code> , <code>Col</code> , <code>PositionX</code> e <code>PositionY</code> .
Illumina	Para arquivos de texto no formato GenomeStudio os cabeçalhos <code>PROBE_ID</code> , <code>Assay_Name_1.QT1</code> , <code>Assay_Name_1.QT2</code> , <code>Assay_Name_2.QT1</code> e <code>Assay_Name_2.QT2</code> são verificados. Arquivos binários no formato IDAT também são suportados.
GenePix	Cabeçalhos <code>Block</code> , <code>Column</code> , <code>Row</code> , <code>X</code> e <code>Y</code> .
NimbleScan	Cabeçalhos <code>PROBE_ID</code> , <code>X</code> e <code>Y</code> .
Generic	Cabeçalhos <code>MetaColumn</code> , <code>MetaRow</code> , <code>Column</code> e <code>Row</code> .

Os dados processados de experimentos de *microarray* são gerados a partir de arquivos de dados brutos, que passaram por um processo de tratamento, como, por exemplo, normalização. São permitidos arquivos de dados processados, ou uma planilha bidimensional contendo arquivo de texto (.txt) delimitados por tabulações. Para a planilha bidimensional, os nomes de sondas e genes são representados nas linhas, enquanto que os dados de uma ou mais hibridizações são representados por meio de colunas.

Os tipos de arquivos aceitos contendo dados processados de *microarray* incluem: arquivo de texto por hibridização, que possui dados de uma única hibridização; planilha de valores normalizados, que contém dados normalizados de diferentes hibridizações; e planilha contendo dados de valores normalizados de diferentes estágios do processamento de dados.

2.2.3 Dados de sequenciamento de RNA

Dados brutos e processados são aceitos para um experimento de sequenciamento de RNA. Existem diferentes formatos de dados de sequenciamento dependendo da plataforma utilizada para obtenção. O ArrayExpress reconhece o tipo de arquivo enviado para a plataforma a partir da extensão dos arquivos de dados. Não são aceitos experimentos de sequenciamento de RNA sem o envio de dados brutos. Porém, o envio de dados processados é opcional.

No ArrayExpress os dados brutos de sequenciamento de RNA devem seguir as especificações definidas em (EMBL-EBI, 2018). Estes dados devem ser enviados via File Transfer Protocol (FTP). Os formatos padrão de submissão para o ArrayExpress são os formatos CRAM, BAM e Fastq. A Tabela 3 apresenta as extensões de arquivos utilizadas para reconhecer cada um destes formatos.

Tabela 3 – Formatos padrão para arquivos de dados brutos de experimentos de sequenciamento de RNA.

Formato	Extensões para arquivos
CRAM	.cram
BAM	.bam
Fastq	.fastq.gz, .fastq.bz2, .fq.gz, .fq.bz2, .txt.gz, .txt.bz2

Além dos arquivos de dados brutos, também é possível o envio de arquivos de dados processados para experimentos de sequenciamento de RNA. Os arquivos de dados

processados possuem dados gerados a partir de arquivos de dados brutos, os quais passaram por algum tipo de tratamento, como, por exemplo, normalização ou análise diferencial. Os arquivos contendo dados processados são gravados no formato texto (.txt) delimitado por tabulações.

O ArrayExpress não obriga o envio de arquivos de dados processados ao submeter um experimento para o repositório. A vantagem da utilização deste tipo arquivo está relacionada ao seu tamanho. Os arquivos de dados processados são bem menores que os arquivos de dados brutos, que com frequência apresentam tamanhos na ordem de *gigabytes*. O armazenamento e a transferência de dados na ordem de *gigabytes* apresentam um desafio para manipulação e armazenamento destes dados. O ArrayExpress não faz restrições sobre o formato de dados processados enviados para o repositório.

2.2.4 Acesso programático ao ArrayExpress

Os dados de experimentos disponibilizados no repositório ArrayExpress podem ser acessados de quatro formas distintas: interface web, pacote ArrayExpress Bioconductor, acesso programático (serviço web) e FTP. Contudo, no contexto deste trabalho, apenas o acesso programático via serviços web é relevante.

O repositório ArrayExpress disponibiliza o conteúdo dos experimentos por meio de serviços REST. Este conteúdo pode ser retornado nos formatos XML ou JavaScript Object Notation (JSON), dependendo da URL utilizada para invocação do serviço.

Os serviços providos pelo ArrayExpress permitem a obtenção de dados e arquivos de experimentos utilizando diversos critérios de busca, como, por exemplo, palavras-chave ou campos específicos do experimento. São fornecidos dois serviços para obtenção dos dados: **Experiments** e **Files**. O serviço **Experiments** permite obter informações gerais sobre o experimento, como, por exemplo, descrição, protocolos, plataformas utilizadas e características das amostras. O serviço **Files** permite obter informações sobre os arquivos publicados para um determinado experimento no repositório. Este serviço retorna, dentre outras informações, a URL para a obtenção do arquivo físico.

O ArrayExpress permite a utilização de campos específicos de um experimento para consulta de dados disponíveis no repositório. Por exemplo, o campo **accession** permite a localização de um experimento por meio de seu identificador de acesso. Uma

lista completa dos campos disponíveis para utilização nas consultas pode ser obtida em (EMBL-EBI, 2015).

O acesso aos serviços providos pelo ArrayExpress consiste na combinação da URL utilizada para obtenção dos dados em formato XML ou JSON, com o nome do serviço (**experiments** ou **files**). Adicionalmente, outros campos podem ser incluídos na URL para serem utilizados na busca, como, por exemplo, o campo **accession** descrito anteriormente. Caso não se utilize nenhum campo específico na invocação do serviço, são retornados todos os dados contidos no repositório. A Tabela 4 ilustra alguns exemplos de requisições que podem ser feitas no repositório, indicando qual o tipo de informação retornada².

Tabela 4 – Exemplos de URLs para a obtenção de dados de experimentos no ArrayExpress.

URL	Informação retornada
/xml/v3/experiments	Recupera todos os dados sobre experimentos contidos no repositório utilizando o formato XML.
/xml/v3/experiments?accession=E-MEXP-3905	Recupera em formato XML informações sobre o experimento cujo identificador de acesso é E-MEXP-3905.
/xml/v3/files?accession=E-MEXP-3905	Recupera em formato XML a lista de arquivos do experimento cujo identificador de acesso é E-MEXP-3905.
/json/v3/experiments?accession=E-MEXP-3905	Recupera em formato JSON informações sobre o experimento cujo identificador de acesso é E-MEXP-3905.
/json/v3/files?accession=E-MEXP-3905	Recupera em formato JSON a lista de arquivos do experimento cujo identificador de acesso é E-MEXP-3905.

O serviço **Experiments** retorna dados nos formatos XML ou JSON. O elemento raiz dos dados de retorno para este serviço é o elemento **experiments**. Dentro do elemento **experiments** existe uma lista de zero ou mais elementos **experiment** que representam os experimentos que atendem aos critérios de busca utilizados na invocação do serviço. Cada elemento **experiment** possui elementos internos para representar informações sobre um

² Para a invocação correta dos serviços, o prefixo <http://www.ebi.ac.uk/arrayexpress> deve ser acrescentado a cada uma das URLs contidas na tabela.

experimento. Estes elementos internos incluem: **protocol**, o qual fornece informações sobre os protocolos utilizados no experimento; **arraydesign**, o qual fornece informações sobre as plataformas; **provider**, o qual fornece informações sobre o responsável pela submissão do experimento no repositório; **bibliography**, o qual fornece informações sobre a publicação que faz referência ao experimento; **samplecharacteristic**, o qual fornece informações sobre as características da amostra; **bioassaydatagroup**, o qual fornece informações sobre o grupo de dados utilizados na análise; **experimentalvariable**, o qual fornece informações sobre as variáveis experimentais; e por fim, **description**, o qual fornece informações gerais sobre o experimento.

O serviço **Files** retorna dados no formato XML ou JSON. O elemento raiz dos dados de retorno para este serviço é o elemento **files**. Dentro deste elemento existe uma lista de zero ou mais elementos **file** que representam os arquivos que atendem aos critérios de busca utilizados na invocação do serviço. Cada elemento **file** possui informações como o tipo de dados, o tamanho e a localização física (URL) do arquivo no repositório.

2.2.5 Dados do Gene Expression Omnibus (GEO)

GEO é hospedado pelo National Center for Biotechnology Information (NCBI). Os experimentos podem ser submetidos para o GEO em diversos formatos, tais como Gene Expression Omnibus Archive (GEOarchive), Simple Omnibus Format in Text (SOFT) e MIAME Notation in Markup Language (MINiML). O formato GEOarchive é constituído por um pacote de arquivos compactados. O formato SOFT é constituído de um arquivo em formato texto simples. Finalmente, o formato MINiML é uma representação em XML do formato SOFT. O formato GEOarchive é o formato recomendado para a maioria das submissões (NCBI, 2016).

O repositório ArrayExpress realiza semanalmente a importação dos dados de experimentos publicados no repositório GEO (EMBL-EBI, 2015). Esta importação permite o acesso a estes experimentos e aos arquivos de dados publicados neste repositório a partir do ArrayExpress. Os experimentos disponibilizados desta maneira também são acessíveis por meio de seus identificadores de acesso. O ArrayExpress gera um identificador de acesso próprio para cada experimento importado. Tanto o identificador de acesso original do GEO, quanto o identificador de acesso gerado pelo ArrayExpress podem ser utilizados

para recuperar o experimento por meio da interface web do ArrayExpress. Porém, o acesso programático restringe seu uso apenas para o identificador de acesso gerado pelo ArrayExpress.

Ao gerar um identificador de acesso para os experimentos importados do GEO, o ArrayExpress atribui um identificador nos formatos **E-GEOID-n** ou **A-GEOID-n**, que representam experimentos ou plataformas, respectivamente. Em ambos os formatos, a letra **n** representa a parte numérica componente do identificador de acesso original do GEO. Por exemplo, os identificadores de acesso **E-GEOID-35642** e **A-GEOID-567** são as versões geradas pelo ArrayExpress dos identificadores de acesso **GSE35642** e **GPL567**, originalmente utilizados pelo GEO.

Os experimentos importados para o ArrayExpress também mantêm uma referência para o experimento original armazenado no repositório GEO. Durante o processo de importação dos dados do GEO para o ArrayExpress, os arquivos Simple Omnibus Format in Text (SOFT) originários do GEO, juntamente com arquivos de dados dos experimentos, são recuperados e processados. Ferramentas de mineração de dados e métodos automatizados são utilizados para recuperar informações gerais sobre experimentos e disponibilizá-las seguindo a padronização de experimentos utilizada no ArrayExpress.

2.3 Redes sociais

Redes sociais *online* são ambientes que permitem aos seus usuários compartilhar, organizar, pesquisar e interagir com outros usuários. A conexão entre os usuários participantes ocorre por meio de um relacionamento interpessoal ou algum tipo de interesse em comum (afinidade). Diferentemente de outros conteúdos na web, como, por exemplo, *sites* pessoais e fóruns de discussão, as redes sociais são organizadas em torno dos usuários participantes e proveem a base para manter uma relação social e/ou profissional entre eles (MISLOVE et al., 2007).

As redes sociais tornaram-se bastante populares devido ao ambiente de interação que proporcionam (NOORDEN, 2014; ZHANG; CHOUDHURY; GRUDIN, 2014). Este ambiente permite que seus usuários interajam e compartilhem conteúdo com a finalidade de alcançar um objetivo comum. Redes sociais facilitam a colaboração entre seus usuários,

podendo ser utilizadas para a divulgação de pesquisa científica. Por meio das redes sociais é possível interagir com pesquisadores com diferentes especialidades e encontrar eventuais colaboradores para projetos de interesse.

Redes sociais podem ser classificadas como redes de propósito geral, redes profissionais e redes de colaboração científica. Redes sociais de propósito geral não restringem o conteúdo compartilhado por seus usuários a um determinado assunto ou público. Devido a esta característica, redes sociais de propósito geral são as mais utilizadas atualmente. Tais redes fornecem mecanismos para suporte à interação entre seus usuários, além de proverem recursos para criação de grupos onde os seus usuários podem discutir assuntos específicos de interesse. As redes sociais profissionais têm por objetivo principal incentivar a discussão sobre assuntos de cunho profissional. Estas redes também proveem recursos para a busca de oportunidades profissionais de seus participantes. Discussões sobre assuntos gerais, que fogem do cunho profissional, não são bem vistas por usuários da rede. Por fim, as redes sociais para colaboração científica tem como característica incentivar a divulgação dos trabalhos de pesquisa que seus usuários estão realizando. Adicionalmente, estas redes também permitem que seus usuários encontrem outros pesquisadores que possam contribuir com seus próprios projetos, criando uma rede de colaboração para o desenvolvimento científico.

Diferentes redes sociais estão atualmente disponíveis, tais como Facebook (FACEBOOK, 2018), Google+ (GOOGLE, 2018), LinkedIn (LINKEDIN, 2018), ResearchGate (RESEARCHGATE, 2018) e myExperiment (GOBLE et al., 2010).

2.3.1 Redes sociais de propósito geral

Facebook

Facebook é possivelmente a rede de relacionamentos de propósito geral mais popular da atualidade (SHEARER; GOTTFRIED, 2017). Esta rede permite compartilhar conteúdo, bem como encontrar e interagir com outros usuários que participam da rede. Esta interação ocorre por meio do círculo de relacionamento (amigos), linha do tempo, páginas e grupos. A linha do tempo concentra todas as publicações de um usuário. As páginas são utilizadas para divulgar conteúdo de qualquer tipo, podendo ser relacionadas a

negócios, organizações, marcas, comunidades, entre outros. Grupos são utilizados para reunir usuários com interesses comuns, permitindo a interação e a colaboração entre os participantes. Os grupos podem ser públicos, fechados ou secretos (FACEBOOK, 2018). Nos grupos públicos qualquer pessoa pode encontrar o grupo, visualizar seus membros e publicações mesmo sem participar do grupo. Nos grupos fechados, qualquer pessoa pode encontrar o grupo e visualizar seus membros, porém as publicações realizadas dentro do grupo estão restritas aos participantes. Nos grupos secretos, somente membros podem encontrar o grupo, visualizar seus participantes, publicações e interagir com o grupo.

O Facebook provê ferramentas que permitem aos desenvolvedores integrarem suas aplicações com a rede social. Estas ferramentas fornecem soluções para processos de autenticação (*login*), compartilhamento de conteúdo, comunicação, estatísticas de utilização e monetização. Estas soluções de integração são suportadas por APIs e pacotes de *software* providos pelo próprio Facebook ou por terceiros.

Google+

Google Plus (Google+) é uma rede social de propósito geral que integra o conjunto de serviços fornecidos pelo Google, incluindo, Gmail, YouTube e Hangouts (GOOGLE, 2018). A rede social permite que seus usuários interajam com outros usuários por meios de círculos de amizade construídos entre os mesmos.

O ambiente provido pelo Google+ ainda provê recursos para criação de áreas utilizadas para trocar experiências sobre um determinado assunto. Estas comunidades podem reunir vários membros com interesses comuns para que possam trocar informações. A integração com outros serviços do Google é um dos grandes diferenciais da rede social. Esta integração facilita a incorporação de vídeos (YouTube) e imagens (Google Fotos), entre outros serviços.

Google+ fornece diversas ferramentas para integração de aplicações a rede social. Soluções para processos de autenticação (*login*), compartilhamento de conteúdo, e busca de informações disponíveis na rede social estão disponíveis por meio da ferramenta Google+ API.

2.3.2 Rede social profissional

LinkedIn é uma rede social que tem como foco a integração de profissionais das mais diversas áreas. A rede tem como objetivo prover um ambiente que permita o *networking* entre seus participantes, possibilitando a troca de ideias, experiências profissionais, além de informações sobre oportunidades de carreira no mercado de trabalho (LINKEDIN, 2018).

Quando um usuário cadastra-se nesta rede social, este publica um *curriculum vitae*. Então, com base nas informações inseridas, conexões com outros usuários passam a ser sugeridas pela rede. A comunicação entre estes diferentes usuários só é permitida caso exista algum vínculo entre eles. Para que um usuário adicione novos usuários em sua rede, estes novos usuários devem ter tido algum tipo de contato prévio com o usuário que deseja adicioná-los. A rede social define este contato prévio ao cruzar informações sobre experiências profissionais em empresas ou instituições de ensino que os usuários têm no presente ou tiveram passado. A rede também permite que um usuário adicione novos usuários que não tiveram um contato direto. Para tanto, basta que estes usuários façam parte da rede de relacionamentos de uma de suas conexões (usuários em sua rede) ou o usuário saiba previamente alguma informação pessoal do usuário a ser adicionado, como, por exemplo, o endereço de e-mail de um usuário.

Assim como o Facebook e o Google+, o LinkedIn provê ferramentas para que desenvolvedores integrem suas aplicações com a rede profissional. Soluções para processos de autenticação (*login*), compartilhamento de conteúdo, e gerenciamento de páginas de empresas estão disponíveis por meio de uma API REST.

2.3.3 Redes sociais de colaboração científica

ResearchGate

ResearchGate é uma rede social para colaboração científica que permite aos pesquisadores das mais diversas áreas de conhecimento encontrar outros pesquisadores com interesses comuns e interagir com os mesmos com o objetivo de compartilhar suas experiências (RESEARCHGATE, 2018). A rede social provê um ambiente que permite a seus usuários seguir outros pesquisadores para os quais exista algum interesse no trabalho, criando

uma rede de relacionamentos. Um usuário da rede social recebe notificações sobre as publicações realizadas por outros usuários de sua rede de relacionamentos (usuários que ele segue).

O ambiente provido pelo ResearchGate incentiva a publicação de resultados de uma pesquisa. A rede social permite a divulgação de diferentes tipos de trabalhos, tenham sido publicados ou não, tais como artigos, apresentações, teses e dados obtidos a partir de experimentos. Adicionalmente, a rede também fornece ferramentas para acompanhar as estatísticas sobre a leitura e citações das publicações na rede social.

O ResearchGate ainda permite o acesso a ferramentas para busca de oportunidades de trabalho em pesquisa. A rede social não fornece ferramentas para integração de aplicações à rede social. Toda publicação de conteúdo é realizado manualmente por seus usuários na rede.

myExperiment

myExperiment é uma rede social para colaboração científica que permite a publicação e o compartilhamento de *workflows* de análise com grupos de usuários. myExperiment permite a formação de uma comunidade de pesquisadores com um interesse comum (GOBLE et al., 2010), fornecendo ferramentas para publicar os resultados de suas pesquisas de maneira a facilitar a reprodução por outros pesquisadores.

Workflows criados por diferentes plataformas integradas de análise podem ser compartilhados entre usuários na rede social. Estes *workflows* podem ser recuperadores e executados em suas plataformas de origem para a reprodução dos experimentos publicados na rede. Para publicar um *workflow* de análise, basta submeter o arquivo obtido a partir de uma plataforma integrada de análise. Qualquer formato de arquivo é permitido, sendo necessário realizar a devida identificação da plataforma de origem durante o processo de publicação do *workflow*.

O myExperiment não fornece ferramentas para a integração de aplicações com a rede social. Porém, possui uma integração nativa com a plataforma Taverna, permitindo a esta plataforma publicar e acessar os *workflows* diretamente nesta rede social.

2.3.4 Integração com redes sociais

Atualmente usuários que fazem uso de diferentes sistemas computacionais precisam utilizar um *login* e uma senha para a autenticação em cada um destes sistemas. O gerenciamento das informações de acesso para cada sistema utilizado é um problema para o usuário. Para simplificar o gerenciamento destas informações, os usuários frequentemente utilizam um mesmo conjunto de informações de acesso em diferentes sistemas, o que representa uma ameaça à segurança das informações. Algumas redes sociais proveem recursos que simplificam o processo de autenticação de uma aplicação integrada à rede social com a finalidade de resolver este problema. Este recurso consiste na utilização da conta de usuário da rede social para a autenticação na aplicação integrada.

A utilização de uma única conta entre diversas aplicações facilita a interação do usuário com diversos sistemas, uma vez que reduz a quantidade de informações que os usuários precisam guardar sobre *logins* e senhas de acesso em cada aplicação. Este tipo de integração também simplifica o cadastro do usuário em uma nova aplicação, pois parte das informações solicitadas durante o cadastro são importadas a partir do perfil do usuário na rede social. Para isto, diversas informações estão disponíveis, como, por exemplo, endereço eletrônico, nome, gênero, localização, foto do perfil, relação de amigos e assuntos de interesse, de acordo com critérios específicos de cada rede social.

A integração de sistemas com a rede social é possível por meio de um registro da aplicação a ser integrada junto à rede social. A aplicação registrada recebe um identificador único que a representa, chamado de ID de Aplicativo. O ID de Aplicativo permite que a aplicação sendo integrada possa analisar o estado de autenticação do usuário corrente, podendo determinar se o usuário está ou não conectado à rede social e à própria aplicação. Tal característica permite personalizar a aplicação integrada fornecendo diferentes conteúdos mediante a autenticação ou não do usuário.

Quando um usuário efetua a autenticação utilizando a rede social, a aplicação que utiliza este recurso recebe uma chave de acesso que representa o usuário autenticado. Esta chave de acesso pode ser utilizada pela aplicação para identificar o usuário em todas as solicitações realizadas posteriormente à rede social. A chave de acesso ainda possui informações sobre quais dados pessoais o usuário poderá ou não compartilhar com uma dada aplicação. Como exemplos de requisições que necessitam de uma chave de acesso,

pode-se citar a publicação de conteúdo e a consulta de informações do perfil do usuário por meio da plataforma integrada. As redes sociais Facebook, Google+ e LinkedIn utilizam um *framework* de autenticação chamado OAuth 2.0 para autenticar seus usuários.

***Frameworks* de autenticação e compartilhamento de conteúdo**

O *framework* de autenticação OAuth 2.0 (IETF, 2012) permite que aplicações obtenham acesso limitado a serviços disponibilizados na web. Este *framework* provê um fluxo de interação entre uma aplicação origem e uma aplicação destino para a obtenção da chave de acesso provido pela aplicação destino.

Scribe Java (GROMOV; GARANINA, 2018) é uma biblioteca escrita na linguagem Java que contém ferramentas para acesso a aplicações que utilizem o *framework* OAuth. Esta biblioteca pode ser facilmente incorporada a aplicações Java. Scribe Java simplifica o acesso a diversas redes sociais, como, por exemplo, Facebook, Google+ e LinkedIn. Durante a sua utilização, são informados à biblioteca os parâmetros obtidos durante o registro da aplicação origem (aplicação externa) na aplicação destino (rede social). Por meio destes parâmetros a biblioteca Scribe Java solicita uma chave de acesso à rede social. Esta chave é utilizada em chamadas subsequentes para interação com esta rede.

Redes sociais também proveem recursos para compartilhamento de conteúdo externo dentro da rede social. Qualquer conteúdo disponível por meio de uma URL na web pode ser compartilhado. Ao compartilhar um conteúdo na rede social o usuário tem como objetivo divulgar este conteúdo, permitindo que outros usuários da rede social tenham acesso ao mesmo.

Os conteúdos passíveis de publicação na rede social normalmente possuem um botão específico para compartilhamento. Ao clicar neste botão o usuário poderá selecionar as opções de compartilhamento disponíveis em cada rede social. Estes recursos podem variar de uma rede social para outra. Durante o compartilhamento de conteúdo, o usuário deverá ser primeiramente autenticado na rede social, para que o conteúdo seja publicado em seu nome. Neste ponto, se o usuário ainda não tiver sido autenticado, informações de acesso serão solicitadas pela rede social.

Redes sociais disponibilizam uma API pública que permite acesso a recursos para a publicação de conteúdo na rede. Esta API normalmente utiliza componentes JavaScript

que devem ser incluídos no conteúdo a ser compartilhado. Parâmetros referentes ao conteúdo a ser compartilhado devem ser passados a estes componentes, como, por exemplo, a URL do recurso a ser publicado na rede social. As redes sociais Facebook, Google+ e LinkedIn possuem ferramentas que auxiliam na geração do código JavaScript necessário à inclusão de um conteúdo a ser compartilhado.

Simple BPMN REST Interaction

Este capítulo apresenta a Simple BPMN REST Interaction (SIBRI), uma API genérica e customizável desenvolvida para prover suporte à execução pelo *framework* Activiti de um processo Business Process Model and Notation (BPMN) representando o comportamento local de um serviço web RESTful. Esta API simplifica a integração de aplicações desenvolvidas na linguagem Java com o *framework* Activiti, de modo a prover um fluxo genérico para a interação com o usuário final e a execução de serviços RESTful.

O capítulo está estruturado da seguinte forma: a seção 3.1 apresenta uma visão geral do uso do *framework* Activiti no suporte a execução de processos BPMN; a seção 3.2 apresenta uma visão geral da API SIBRI, sua estrutura e dinâmica de interação do usuário, além de discutir como a API pode ser estendida para necessidades específicas de execução; e por fim, a seção 3.3 exemplifica o uso da API.

3.1 Suporte à execução de processos BPMN

No contexto de trabalho, serviços RESTful são descritos utilizando a especificação WSDL 2.0. Uma especificação WSDL descreve um serviço web, definindo suas interfaces de acesso, as operações e o formato das mensagens trocadas para a comunicação com este serviço. Um serviço RESTful consiste de um conjunto de operações, as quais devem ser executadas em uma determinada ordem. O comportamento local de um serviço representa a ordem em que este conjunto de operações deve ser invocado para a obtenção de um resultado de interesse (GUARDIA, 2016).

Business Process Model and Notation (BPMN) é uma linguagem utilizada para

descrever processos de negócio, por meio da especificação de um conjunto de tarefas e da ordem de execução destas tarefas para a obtenção de um resultado esperado. Uma especificação BPMN também pode ser utilizada para descrever o comportamento local de um serviço. Neste caso, cada operação do serviço pode ser mapeada para uma tarefa BPMN correspondente. Adicionalmente, relacionamentos devem ser estabelecidos entre estas tarefas para representar o fluxo de execução das operações do serviço.

Activiti é um *framework* que permite projetar, implementar e executar processos descritos por meio de uma especificação BPMN (ACTIVITI, 2018). O *framework* Activiti interpreta uma especificação BPMN e executa as operações descritas na ordem pré-estabelecida. Uma especificação BPMN pode ser estendida para incorporar atributos e *tags* específicos do *framework* Activiti. Por meio desta extensão, recursos adicionais, não previstos na especificação BPMN, podem ser providos.

O *framework* Activiti não provê suporte diretamente à execução de uma especificação BPMN que represente as operações locais de um serviço RESTful. Tal limitação pode ser indiretamente contornada pelo *framework* por meio da criação de classes Java que podem realizar qualquer tipo de operação, incluindo a invocação de operações em serviços RESTful. Para a utilização deste recurso, uma classe que implementa a interface `JavaDelegate` (pacote `org.activiti.engine.delegate`) deve ser provida. Esta classe deverá ser referenciada em uma tarefa contida na especificação BPMN de um serviço. O método `execute` será invocado durante a execução da tarefa, resultando na invocação da operação específica do serviço RESTful. Esta estrutura ainda permite que parâmetros sejam passados para a classe de implementação por meio da inclusão de atributos do tipo `org.activiti.engine.delegate.Expression` na classe de implementação. Estes atributos podem ter seus valores definidos na própria especificação BPMN.

Assim, para que o *framework* Activiti possa ser utilizado para a execução de um serviço RESTful, a especificação BPMN base do comportamento local deste serviço deve ser estendida por meio da definição de dois atributos. O primeiro atributo, `activiti:class`, indica qual classe Java será responsável pela execução desta tarefa. O segundo atributo, `activiti:field`, indica quais informações (valores de parâmetros) devem ser repassadas para a classe Java responsável pela execução da tarefa.

A solução fornecida pelo *framework* Activiti para execução de serviços RESTful

requer a criação de uma classe específica para a execução de cada tarefa contida na especificação BPMN do comportamento local de um serviço. Tal característica dificulta a utilização de novos serviços mapeados utilizando a especificação BPMN, pois exige conhecimentos de programação em Java para a criação de classes específicas para suporte. Adicionalmente, o uso de classes específicas para cada operação não favorece o reúso destas classes. Finalmente, o *framework* Activiti não provê suporte para a interação com usuário para a coleta dos valores de parâmetros de entrada. Assim, estes valores podem ser definidos de forma fixa diretamente na especificação BPMN estendida. Neste contexto, a API Simple BPMN REST Interaction (SIBRI) foi criada para sanar tais limitações.

3.2 API SIBRI

3.2.1 Visão geral

Simple BPMN REST Interaction (SIBRI) é uma API genérica e customizável desenvolvida para prover suporte à execução de um processo BPMN representando o comportamento local de um serviço web RESTful. SIBRI elimina a necessidade de codificação de classes específicas, simplificando a sua utilização em diferentes contextos. Esta API define um conjunto de classes genéricas responsáveis pela interação com o *framework* Activiti e pela execução de cada operação definida em um serviço de análise. Adicionalmente, esta API suporta a definição de parâmetros de entrada em uma especificação BPMN, assim como a definição de valores pré-definidos para estes parâmetros. Finalmente, SIBRI suporta a definição de classes customizáveis para tratar a interação com o usuário durante a invocação de um serviço.

Em linhas gerais, a API SIBRI provê suporte à:

- Especificação de classes genéricas responsáveis pela execução das requisições às operações de um serviço. A identificação da classe responsável pela execução de uma determinada operação (tarefa BPMN) deve ser definida na especificação BPMN associada ao comportamento local do serviço de interesse;
- Inserção de parâmetros definidos na especificação BPMN. Estes parâmetros indicam quais informações devem ser solicitadas ao usuário durante a execução de cada

operação de um serviço. A API define mecanismos para permitir que o usuário da aplicação informe o valor de cada parâmetro definido;

- Construção de classes customizáveis para acomodar requisitos específicos da execução de cada serviço. Tais classes podem ser definidas para permitir o desenvolvimento de mecanismos de interação com o usuário e/ou invocação de serviços diferentes do suporte provido pelas classes genéricas;
- Construção de objetos que devem ser enviados em formato específico para um serviço. Tais objetos podem conter, por exemplo, dados no formato XML, necessários à execução de um determinado serviço.

As classes da API SIBRI foram criadas e agrupadas em três pacotes. O pacote **core** contém as principais classes da API. Este pacote define uma solução genérica para a execução de serviços RESTful. O pacote **builder** define uma interface genérica para a conversão de objetos durante a invocação de um serviço. Por fim, o pacote **parameter** provê suporte à interação com o usuário de modo a permitir a definição de parâmetros necessários à execução de uma determinada operação.

3.2.2 Pacote core

As classes do pacote **core** (`br.usp.sibri.core`) fornecem mecanismos para suporte à execução das operações de um serviço, definição de parâmetros a serem solicitados ao usuário, e definição de valores padrão para estes parâmetros. Estas classes utilizam os demais pacotes da API para prover suas funcionalidades. A Figura 1 apresenta a estrutura de classes do pacote **core**.

A classe **AbstractServiceTask** contém as operações comuns às classes responsáveis pela execução de uma operação de um serviço (tarefa BPMN). Esta classe possui atributos e métodos que provêm suporte a execução de uma operação. De maneira geral, estes atributos são inicializados com as informações definidas na especificação BPMN por meio do atributo de extensão `activiti:field`. Os atributos configuráveis definidos por **AbstractServiceTask** são:

Figura 1 – Interdependência entre as classes do pacote **core**.

- **requiredParameters**, o qual define os parâmetros solicitados de forma obrigatória durante a execução de uma operação. Cada parâmetro solicitado ao usuário é definido de forma composta por dois valores separados pelo caractere dois pontos (“:”). O primeiro valor representa o nome do parâmetro de entrada da operação, enquanto que o segundo valor representa a mensagem de texto que será apresentada ao usuário;
- **defaultValues**, o qual define de forma opcional valores padrão para os parâmetros definidos pelo atributo **requiredParameters**. O valor padrão de um parâmetro deve ser informado neste campo utilizando-se como separador o caractere dois pontos (“:”);
- **builders**, o qual especifica uma classe Java responsável pela conversão do valor de um parâmetro em um objeto em um formato específico antes da invocação da operação;
- **httpMethod**, o qual define o método HTTP que deve ser invocado para a execução da operação;
- **urlPathParam**, o qual define a estrutura da URL utilizada para a passagem de parâmetros na invocação da operação;
- **bodyData**, o qual indica a existência de um parâmetro que deve ser enviado no corpo da requisição HTTP. Esta operação é suportada pelos métodos POST, PUT e DELETE.

A classe `AbstractServiceTask` provê ainda diferentes métodos auxiliares. Os métodos `getWebResource` e `getExecution` são responsáveis pela obtenção do contexto de execução de um serviço RESTful e do contexto de execução de um processo BPMN, respectivamente. Os métodos `getId`, `getServiceOperationId`, `getSessionId`, `getVariableAsString` e `getVariable` são utilizados para obtenção de informações sobre o ambiente de execução. Os métodos `setVariable`, `stopProcessing`, `continueProcessing`, `setNextStep` podem ser utilizados para atribuir valores de execução, assim como alterar o fluxo de execução dos serviços. Os métodos `buildInputRequiredParameter`, `buildSelectRequiredParameter`, `buildMultipleSelectRequiredParameter`, `buildBooleanRequiredParameter`, `builderParam` e `getRequiredParameter` são utilizados para a construção e a leitura de parâmetros que serão solicitados ao usuário. A classe ainda possui um método abstrato `invoke` que deve ser implementado pelas classes filhas com a finalidade de invocar um serviço RESTful associado.

A classe `DefaultServiceTask` provê uma implementação genérica para a classe abstrata `AbstractServiceTask`. `DefaultServiceTask` provê suporte à invocação de serviços RESTful utilizando os métodos HTTP GET, POST, PUT e DELETE. Esta classe também provê uma implementação base para manipular os atributos definidos pela classe abstrata, permitindo assim a invocação dos métodos HTTP suportados.

A classe `GetServiceTask` especializa a classe `DefaultServiceTask` para prover suporte à invocação de requisições HTTP GET. De forma análoga, as classes `PostServiceTask`, `PutServiceTask` e `DeleteServiceTask` foram definidas para prover suporte à invocação de requisições HTTP POST, PUT e DELETE, respectivamente.

A classe `GatewayServiceTask` especializa a classe abstrata `AbstractJavaService` de modo a prover suporte à interação com o usuário (requisição de parâmetros). Os parâmetros solicitados são utilizados na definição de valores das variáveis que controlam o fluxo de execução de tarefas BPMN.

Finalmente, a classe `ServiceTaskNotImplemented` representa uma operação ainda não implementada de um serviço. Assim, embora esta classe não invoque serviço algum, esta pode ser utilizada no desenvolvimento/teste incremental de novos serviços.

3.2.3 Pacotes builder e parameter

O pacote `builder` (`br.usp.sibri.builder`) define a interface `ObjectBuilder`, a qual deve ser implementada por uma classe responsável pela construção de objetos que representam valores de parâmetros em formatos específicos. Tais objetos podem conter, por exemplo, valores de parâmetros encapsulados no formato XML. Assim, a partir das informações coletadas por meio da interação com o usuário, um conteúdo XML pode ser construído e enviado ao serviço alvo da requisição. A interface `ObjectBuilder` define apenas o método `build`, o qual recebe, processa e retorna um objeto do tipo `java.lang.Object`. Esta interface é utilizada em conjunto com a classe `AbstractServiceTask`.

O pacote `parameter` (`br.usp.sibri.parameter`) contém as classes responsáveis por coletar os parâmetros necessários à execução de uma determinada operação. As classes do pacote `core` interagem com as classes contidas neste pacote para permitir a coleta destes parâmetros. A Figura 2 apresenta a estrutura do pacote `parameter`.

Figura 2 – Interdependência entre as classes do pacote `parameter`.

A classe `RequiredParameter` indica que o valor de determinado parâmetro necessário à invocação de uma operação deve ser fornecido pelo usuário. O valor deste parâmetro será então utilizado na invocação da operação.

A classe `AbstractRequiredParameter` representa de forma abstrata um parâmetro específico que deve ser solicitado ao usuário. O tipo enumerado `ComponentType` representa os diferentes tipos de parâmetros existentes (`SELECT`, `MULTIPLE`, `BOOLEAN` e `INPUT`). Os tipos `SELECT` e `MULTIPLE` representam opções a serem apresentadas ao usuário. A diferença entre estes tipos reside no número de alternativas que podem ser escolhidas pelo usuário. O tipo `SELECT` permite a escolha de apenas uma alternativa, enquanto o tipo `MULTIPLE` permite a escolha de mais de uma alternativa. O tipo `BOOLEAN` representa uma alternativa

booleana. Finalmente, o tipo `INPUT` representa um valor diretamente informado pelo usuário. `AbstractRequiredParameter` possui ainda um conjunto de atributos, métodos auxiliares e métodos abstratos (não representados na Figura 2) para a verificação das respostas do usuário. Por exemplo, o método `getAnswered` é utilizado para verificar se um parâmetro foi informado pelo usuário, enquanto que o método `getValueSelected` é utilizado para obter o valor deste parâmetro.

As classes `SelectRequiredParameter`, `MultipleSelectRequiredParameter`, `BooleanRequiredParameter` e `InputRequiredParameter` especializam a classe `AbstractRequiredParameter` para representar concretamente os mecanismos de interação com o usuário, os quais são respectivamente utilizados para a obtenção dos tipos de parâmetro `SELECT`, `MULTIPLE`, `BOOLEAN` e `INPUT`. Estas classes implementam concretamente os métodos abstratos definidos na classe `AbstractRequiredParameter`.

Por fim, a exceção `RequiredParameterException` sinaliza que a execução de uma operação não pode ser iniciada, haja visto a ausência de um ou mais parâmetros exigidos para a invocação desta operação. Os valores dos parâmetros faltantes são então solicitados ao usuário.

3.2.4 Dinâmica de execução

A Figura 3 ilustra de maneira geral a dinâmica de execução da API SIBRI por meio de um diagrama de sequência UML. Esta execução representa um típico cenário de uso no qual uma operação com um ou mais parâmetros obrigatórios deve ser executada.

O usuário solicita ao *framework* Activiti a execução de uma especificação BPMN representando o comportamento local de um serviço. De modo a executar o serviço especificado, o *framework* Activiti dispara a execução das tarefas especificadas pelo comportamento local. Para cada tarefa especificada, o *framework* invoca o método `execute` da classe `AbstractServiceTask`. Esta classe, por sua vez, irá inicializar todos os atributos de controle e irá invocar o método `invoke` na classe `DefaultServiceTask`. A classe `DefaultServiceTask` verifica se existem parâmetros a serem solicitados ao usuário. Caso estes parâmetros não existam, o serviço é imediatamente invocado por meio do método `invokeService`. Caso contrário, uma lista de parâmetros deve ser construída.

De modo a construir a lista de parâmetros, a classe `DefaultServiceTask` invoca o

Figura 3 – Dinâmica de execução da API SIBRI

método `getRequiredParameter`, o qual irá utilizar o método `buildInputRequiredParameter` da classe `RequiredParameter` para construir um objeto `InputRequiredParameter` para cada parâmetro necessário à invocação da operação. Este processamento encerra-se por meio do método `stopProcessing` quando a lista tiver sido criada. Após a criação da lista de parâmetros, os valores de cada parâmetro devem ser informados pelo usuário (não representado na figura 3). Enquanto houver algum parâmetro pendente de definição pelo usuário, a operação não pode ser invocada. Quando todos os parâmetros tiverem sido informados, a classe `DefaultServiceTask` constrói os objetos no formato esperado pelo serviço utilizando o método `build` da interface `ObjectBuilder`, e invoca a execução da operação por meio do método `invokeService`.

3.2.5 Extensão da API

Não obstante o suporte padrão fornecido pelas classes da API SIBRI, podem existir situações que necessitem de um tratamento diferenciado. Por exemplo, os valores de um parâmetro do tipo **SELECT** devem ser obtidos de forma dinâmica por meio da consulta a um banco de dados ou pela execução de um processamento qualquer. Tal comportamento não é suportado diretamente pela API. Contudo, a API pode ser estendida para suportar este requisito de interação.

SIBRI pode ser estendida pela definição tanto de uma classe Java que especialize a classe **DefaultServiceTask**, quanto de uma classe Java que especialize a classe **AbstractJavaService**. A especialização da classe **DefaultServiceTask** representa a opção de extensão mais simples porque esta classe implementa o fluxo de interação com o usuário para a solicitação de parâmetros. Embora a classe **AbstractJavaService** também possa ser especializada para prover a funcionalidade desejada, trata-se de uma opção mais complexa pois esta classe não possui tal comportamento.

Adicionalmente, caso um requisito não possa ser atendido diretamente por meio da extensão das classes **DefaultServiceTask** e **AbstractJavaService** em virtude das especificidades do problema envolvido, uma nova classe que implementa diretamente a interface **JavaDelegate** do *framework* Activiti pode também ser desenvolvida. A API SIBRI não restringe o uso de nenhum dos recursos disponíveis no *framework*.

3.3 Cenário de uso da API

Para exemplificar o uso da API SIBRI foi definido um cenário base contendo um serviço RESTful hipotético de análise de dados chamado **GenericAnalysis**. Este serviço é composto por três operações **submitData**, **executeAnalysis** e **retrieveResult**. A operação **submitData** é responsável por enviar um conjunto de arquivos para análise. A operação **executeAnalysis** realiza o processamento da análise sobre os arquivos enviados. Por fim, a operação **retrieveResult** é utilizada para obter um arquivo contendo o resultado da análise.

A Figura 4 apresenta uma representação visual da especificação WSDL do serviço **GenericAnalysis**, segundo a notação proposta em (GUARDIA, 2016). De acordo com esta representação, as operações **submitData**, **executeAnalysis** e **retrieveResult** fazem parte da interface do serviço **GenericAnalysis**. Parâmetros de entrada e/ou saída presentes em cada operação não estão representados na figura.

Figura 4 – Representação visual WSDL da especificação do serviço **GenericAnalysis**.

A partir da especificação WSDL deste serviço, as seguintes atividades deverão ser realizadas para permitir a execução (automática) do mesmo por meio da API SIBRI: 1) criação de uma especificação BPMN base para representar a ordem de execução das operações do serviço; e 2) extensão da especificação BPMN base para uso da API SIBRI.

Criação de uma especificação BPMN base

A Figura 5 apresenta graficamente a especificação BPMN para este serviço. A ordem de invocação das operações deste serviço (comportamento local) inicia-se pelo elemento **Start** passando por cada uma das operações até alcançar o elemento **End**.

Figura 5 – Especificação BPMN para o serviço **GenericAnalysis**.

A Figura 6 apresenta trechos da representação XML da especificação BPMN para a execução do serviço **GenericAnalysis**. Esta especificação está dividida em cinco trechos. O trecho 1 apresenta a URL do arquivo WSDL associado, bem como a URL do arquivo BPMN propriamente dito. O trecho 2 inicia o mapeamento de um processo (serviço) BPMN. O atributo **id** é utilizado para criar um identificador único para o serviço na especificação

Figura 6 – Representação XML da especificação BPMN das operações definidas no WSDL do serviço **GenericAnalysis**.

BPMN, enquanto que o atributo `name` é utilizado para prover um nome ao mesmo. O trecho 3 declara os eventos que podem ocorrer durante a execução do serviço. Por exemplo, nesta especificação pode-se identificar um evento de início (**Start**) cujo identificador (`id`) é `StartEvent_1` e um evento de término (**End**), cujo identificador é `EndEvent_1`. O trecho 4 declara as operações (tarefas) de um serviço. Pode-se verificar que cada tarefa está vinculada (mapeada) para uma operação do WSDL correspondente (mesmo nome usado na especificação WSDL). Por fim, o trecho 5 apresenta a definição da sequência do fluxo da execução das operações locais do serviço. Partindo do elemento `StartEvent_1` o fluxo irá invocar sequencialmente as operações `submitData`, `executeAnalysis`, `retrieveResult` e, por fim, o elemento `EndEvent_1`.

Extensão da especificação BPMN base

Para que esta especificação BPMN possa ser executada utilizando o *framework* Activiti por meio do suporte provido pela API SIBRI, faz-se necessário estendê-la com a definição de quais classes Java serão utilizadas para executar cada uma das tarefas BPMN. Durante

esta extensão, valores de parâmetros ainda podem ser definidos para serem utilizados durante a execução das operações do serviço.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions
  ...
  xmlns:activiti="http://activiti.org/bpmn"
  xmlns:wsdl="http://server1/analysisservices/genericanalysis.wsdl"
  targetNamespace="http://server1/analysisservices/genericanalysis.bpmn">

  <!-- Mapeamento do processo -->
  <process id="GenericAnalysis" name="GenericAnalysis" isExecutable="true">

    <!-- Tarefas de serviços -->
    <serviceTask id="submitData" name="wsdl:ServiceInterface.submitData"
      activiti:class="br.usp.sibri.core.PostServiceTask" />

    <serviceTask id="executeAnalysis" name="wsdl:ServiceInterface.executeAnalysis"
      activiti:class="br.usp.sibri.core.PostServiceTask">
      <extensionElements>
        <activiti:field name="requiredParameters" stringValue="cutoff:Cut-off" />
        <activiti:field name="defaultValues" stringValue="cutoff:0.05" />
        <activiti:field name="bodyData" stringValue="cutoff" />
      </extensionElements>
    </serviceTask>

    <serviceTask id="retrieveResult" name="wsdl:ServiceInterface.retrieveResult"
      activiti:class="br.usp.sibri.core.GetServiceTask" />

    <!-- Objetos de dados -->
    <dataObject id="DataObject_1" name="serviceURL" itemSubjectRef="xsd:string">
      <extensionElements>
        <activiti:value>http://server1/analysisservices/genericanalysis</activiti:value>
      </extensionElements>
    </dataObject>

    ...
  </process>
</definitions>
```

Figura 7 – Especificação BPMN estendida para uso da API SIBRI na execução das operações do serviço GenericAnalysis.

A Figura 7 apresenta a extensão da especificação BPMN para permitir a execução do serviço GenericAnalysis pela API SIBRI. A especificação estendida divide-se em três trechos. O trecho 1 inclui a declaração do *namespace* para uso do *framework* Activiti. O trecho 2 mapeia cada tarefa BPMN para uma classe da API SIBRI responsável por sua execução. O mapeamento é realizado utilizando o atributo `activiti:class`. O valor deste atributo é o nome da classe da API SIBRI responsável por realizar a invocação da operação do serviço GenericAnalysis. As operações `submitData` e `executeAnalysis` foram mapeadas para a classe `br.usp.sibri.core.PostServiceTask`, enquanto que a operação `retrieveResult` foi mapeada para a classe `br.usp.sibri.core.GetServiceTask`.

Ainda no trecho 2, pode-se observar o uso do elemento `activiti:field`, responsável por informar os parâmetros que devem ser repassados à API SIBRI. Neste exemplo, este elemento é utilizado para indicar que o parâmetro `cutoff` deve ser solicitado ao usuário durante a execução desta operação do serviço. Como consequência, o parâmetro `requiredParameters` da classe `AbstractServiceTask` deve ser instanciado. Este parâmetro terá seu valor padrão definido como 0.05 (atributo `defaultValues`) e deverá ser passado no corpo da requisição POST (atributo `bodyData`), definida pelo uso da classe `br.usp.sibri.core.PostServiceTask` para esta operação. Por fim, o trecho 3 apresenta a definição de um parâmetro com visibilidade global a todas as tarefas contidas em um serviço por meio do elemento `activiti:value`. Neste exemplo, este elemento está sendo utilizado para definir a URL para a invocação do serviço (parâmetro `serviceURL`).

3.4 Considerações finais

A API SIBRI foi concebida para simplificar a incorporação de novos serviços de análise na plataforma SemanticSCo Web. Contudo, o uso da API SIBRI não está restrito a esta plataforma ou a este domínio de aplicação. A API SIBRI pode ser utilizada em diferentes domínios de aplicação para a integração de aplicações desenvolvidas na linguagem Java com o *framework* Activiti, de modo a prover um fluxo genérico para a interação com o usuário final e a execução de serviços RESTful.

A padronização provida pela API SIBRI para a integração com o *framework* Activiti ainda permite que ferramentas possam ser futuramente desenvolvidas para a geração semi-automática do arquivo de especificação BPMN. Tais ferramentas poderiam utilizar uma interface visual para que o usuário possa determinar a ordem em que as operações de um serviço seriam invocadas e quais parâmetros deveriam ser fornecidos. Estas ferramentas teriam como diferencial simplificar a construção da especificação BPMN ao sugerir parâmetros e classes de suporte disponíveis na API SIBRI para a execução de cada operação de um serviço. Como resultado final, o arquivo contendo a especificação BPMN poderia ser gerado automaticamente pela ferramenta desenvolvida.

Plataforma SemanticSCo Web

SemanticSCo é uma plataforma para a análise de dados de expressão gênica que suporta a composição semiautomática de serviços web. A plataforma permite a criação e a execução de *workflows* de análise baseados na composição semântica de um conjunto de serviços. SemanticSCo simplifica o processo de criação dos *workflows* ao sugerir aos seus usuários as atividades de análise (serviços) que podem ser incluídas em um *workflow*, baseando-se em dados e em atividades de análise semanticamente anotados. A plataforma SemanticSCo Web representa uma evolução da plataforma SemanticSCo para o ambiente web. Adicionalmente um conjunto de melhorias e novos recursos foram incorporados à plataforma.

Este capítulo apresenta uma visão geral da plataforma SemanticSCo Web. O capítulo está estruturado da seguinte forma: a seção 4.1 apresenta a arquitetura da plataforma; a seção 4.2 apresenta as melhorias introduzidas para ampliar o suporte à acessibilidade; a seção 4.3 apresenta os recursos introduzidos para prover suporte à reprodutibilidade; a seção 4.4 apresenta os recursos introduzidos para prover suporte à transparência; e por fim, a seção 4.5 apresenta outras modificações incorporadas à plataforma.

4.1 Visão geral da plataforma SemanticSCo Web

4.1.1 Construção de *workflows* de análise

A plataforma SemanticSCo (GUARDIA, 2016; GUARDIA et al., 2017) provê suporte para que seus usuários (biólogos/bioinformatas) criem *workflows* de análise de dados de expressão gênica. Estes *workflows* são criados a partir da composição semiautomática de serviços web semânticos. SemanticSCo utiliza um conjunto de serviços de análise de dados de expressão gênica incorporados à plataforma a partir do repositório Gene Expression Analysis Services (GEAS) (GUARDIA et al., 2015).

SemanticSCo utiliza uma estratégia de composição semiautomática e interativa. De acordo com esta estratégia, *workflows* são criados em duas etapas. Na primeira etapa, o usuário especifica atividades de análise abstratas e a ordem em que estas atividades devem ser executadas. Na segunda etapa, serviços concretos são atribuídos a estas atividades, permitindo ao usuário solicitar a execução do *workflow*. A plataforma SemanticSCo sugere novas atividades de análise que podem ser incluídas com base no conjunto de dados e atividades de análise atualmente disponíveis no *workflow*. Para isso, a plataforma utiliza informações de anotações semânticas associadas aos serviços que estão disponíveis na plataforma. A associação entre os serviços (ligação de arestas entre estes serviços) determinam os dados de entrada que devem ser utilizados por cada serviço.

A plataforma SemanticSCo permite a execução de cada atividade de análise isoladamente ou em conjunto durante a execução completa de um *workflow*. Parâmetros de execução podem ser solicitados ao usuário. Este processo atende à natureza exploratória dos estudos de expressão gênica. O biólogo pode especificar um conjunto de atividades de análise e adicionar novas atividades conforme seus estudos vão evoluindo. A ordem de execução e os parâmetros utilizados podem ser modificados a qualquer momento.

4.1.2 Arquitetura da plataforma

A plataforma SemanticSCo Web foi desenvolvida utilizando a linguagem Java (ORACLE, 2015), fazendo uso da linguagem HTML 5 (W3C, 2017) e dos recursos disponíveis na plataforma Java Enterprise Edition 7 (JEE) (ORACLE, 2013b). A Figura 8 apresenta uma

visão geral da arquitetura da plataforma. Esta arquitetura é formada por um conjunto de módulos, cada qual formado por um conjunto de componentes (não representados na figura). Os módulos incluídos nesta nova versão estão destacados em azul, enquanto os módulos reutilizados da versão original da plataforma estão destacados em marrom. Finalmente, serviços externos à plataforma estão destacados em verde. Uma seta representa uma relação de dependência entre dois módulos.

Figura 8 – Arquitetura web da plataforma SemanticSCo.

O usuário acessa a plataforma utilizando um navegador web. O módulo **Composite Service Enactment** controla toda a interação do usuário com a plataforma a fim de criar e executar um *workflow* de análise. Este módulo é responsável por apresentar qualquer informação sobre *workflows* aos usuários da plataforma. Durante a criação de um *workflow*, o módulo **Composite Service Enactment** pode interagir com o módulo **Gene Expression Repository Data** para buscar dados de experimentos de expressão gênica que podem ser analisados na plataforma. O módulo **Coordinator** é responsável por intermediar a comunicação entre o módulo **Composite Service Enactment** e demais módulos responsáveis pela composição e execução de serviços (módulos **Service Composition** e **Composition and Execution Context**).

A interação entre os módulos **Composite Service Enactment** e **Coordinator** ocorre por meio de um conjunto de comandos primitivos. O módulo **Coordinator** interage com o módulo **Service Composition** para descobrir serviços que possam ser utilizados

em uma composição. O módulo **Service Composition** interage com o módulo **Service Registry** para recuperar a lista de todos os serviços disponíveis na plataforma. O módulo **Coordinator** armazena as informações sobre os serviços descobertos e a composição do *workflow* no módulo **Composition and Execution Context**. Após a construção de um *workflow*, o módulo **Composite Service Enactment** interage com o módulo **Workflow Execution Manager** para controlar sua execução. Cada serviço selecionado para execução é então invocado diretamente no repositório de Serviços de Análise (GEAS). Um *workflow* pode ser compartilhado entre usuários da plataforma ou usuários de uma rede social utilizando o módulo **Social Network Integration**. O módulo **Authentication** controla o acesso de um usuário à plataforma. Este módulo pode utilizar recursos das redes sociais (via módulo **Social Network Integration**) para simplificar este acesso.

Os comandos primitivos utilizados na comunicação entre os módulos **Composite Service Enactment** e **Coordinator** são responsáveis pela execução de uma determinada atividade associada ao processo de criação de um *workflow* de análise. Estes comandos ainda especificam os formatos de mensagens de requisição e resposta trocadas entre estes dois módulos. O módulo **Coordinator** provê três operações para suporte à execução dos comandos primitivos: **startSession**, **clearSession** e **coordinate**. A operação **startSession** inicia um novo *workflow* de análise exclusivo para um determinado usuário. Para identificar cada *workflow* criado, a operação **startSession** retorna um identificador único que deve ser utilizado em chamadas subsequentes para a identificação de um dado *workflow* em desenvolvimento. A operação **clearSession** finaliza um *workflow* existente excluindo os seus dados. Por fim, a operação **coordinate** é utilizada durante a invocação de cada um dos comandos primitivos utilizados no fluxo de comandos. Maiores informações sobre os comandos primitivos podem ser encontradas em (GUARDIA, 2016).

Durante a construção de um *workflow* de análise, o módulo **Composition and Execution Context** armazena informações referentes ao processo de composição e execução dos serviços do *workflow* de análise. Cada processo de composição é gerenciado por uma instância diferente do módulo **Composition and Execution Context**, que por sua vez é associado a uma sessão de usuário.

O módulo **Service Creation and Publication** fornece suporte à criação e publicação de serviços web semânticos na plataforma SemanticSCo. Estes serviços são registrados junto ao módulo **Service Registry**. Para que um serviço possa ser publicado

na plataforma, sua interface deve ser descrita por meio de uma especificação Web Services Description Language (WSDL) 2.0 e seu comportamento local especificado por meio de uma diagrama Business Process Model and Notation (BPMN) 2.0. Adicionalmente, a especificação WSDL deve ser anotada semanticamente com o padrão Semantic Annotations for WSDL and XML Schema (SAWSDL) (W3C, 2007). Esta anotação associa informações semânticas a cada serviço de análise utilizado pela plataforma, permitindo que a mesma possa sugerir tais serviços aos usuários durante o processo de criação de um *workflow* de análise. O módulo **Service Creation and Publication** provê componentes distintos para auxiliar o usuário responsável pela criação e publicação de novos serviços à plataforma (usualmente um bioinformata). Os componentes **Wsd12 Creator** e **Wsd12Bpmn Mapper** auxiliam no processo de criação dos serviços, enquanto que os componentes **BPMN Interpreter** e **BPMN Publisher** auxiliam no processo de publicação.

O módulo **Service Registry** mantém o registro de serviços de análise disponíveis na plataforma. Este componente utiliza estruturas de categorização definidas como extensões do Universal Description, Discovery and Integration (UDDI) 3.0.2 (OASIS, 2004). Estas estruturas de categorização tem como objetivo permitir o armazenamento de informações extraídas a partir dos serviços mapeados.

4.2 Suporte à Acessibilidade

A plataforma SemanticSCo Web possui uma interface gráfica simples, a qual abstrai diferentes aspectos relacionados à criação e execução de *workflows* na plataforma.

4.2.1 Composite Service Enactment

O módulo **Composite Service Enactment** provê uma interface gráfica simples e intuitiva para a criação e execução de *workflows* de análise. Na versão original da plataforma SemanticSCo, o componente **Composite Service Enactment** provia uma interface *desktop*. Esta interface limitava o uso da plataforma pois exigia a execução de uma aplicação local para acesso à plataforma. Na nova versão da plataforma SemanticSCo, este componente foi totalmente reestruturado a partir da implementação de uma interface web, acessível a partir de diferentes dispositivos computacionais.

O módulo **Composite Service Enactment** consiste de dois componentes: **Workflow Construction** e **Composition Manager**. Estes componentes foram desenvolvidos utilizando as tecnologias JavaServer Faces (JSF) (ORACLE, 2013a) e Enterprise JavaBeans (EJB) (ORACLE, 2013d), respectivamente. JSF é um *framework* para o desenvolvimento de aplicações Java para web que permite a reutilização de componentes a partir de bibliotecas próprias ou de terceiros. Este *framework* permite a criação de conteúdo para seus clientes (navegadores web), facilitando a construção e reutilização de componentes. O EJB é uma tecnologia que permite escrever componentes que incorporam regras de negócio da aplicação. Para isso, o EJB fornece recursos como controle de transação, controle de concorrência e segurança, removendo dos desenvolvedores a necessidade de implementação de rotinas específicas para os mesmos. A Figura 9 apresenta a arquitetura do módulo **Composite Service Enactment**.

O componente **Workflow Construction** é responsável por exibir (*renderizar*) um *workflow* de análise em um formato visual para o usuário. Este componente interage com o módulo **Gene Expression Repository Data** para recuperar dados de experimentos de expressão gênica a partir do repositório ArrayExpress. O componente **Composition Manager** é responsável por persistir todas informações relacionadas ao *workflow* de análise em um banco de dados relacional MySQL (MYSQL, 2018), utilizando para isso a especificação Java Persistence API (JPA) (ORACLE, 2013c). O JPA realiza o mapeamento objeto-relacional dos dados a fim de simplificar a consulta e persistência de informações contidos no banco de dados MySQL. Por fim, o componente **Composition Manager** interage com o módulo **Workflow Execution Manager** para solicitar a execução dos *workflows* construídos na plataforma.

Figura 9 – Arquitetura do módulo **Composite Service Enactment**.

Interface gráfica

As interfaces apresentadas ao usuário na nova versão foram criadas utilizando o JSF 2.2, em conjunto da biblioteca de componentes PrimeFaces (PRIMEFACES, 2018). A versão 2.2 do JSF tem suporte à geração de código HTML 5. As tecnologias CSS 3 (W3C, 2015) e jQuery (JQUERY, 2018) também foram utilizadas em conjunto com o JSF e PrimeFaces para proporcionar uma interface web agradável aos usuários. O CSS 3 é uma linguagem para a definição de estilos na apresentação de documentos HTML, enquanto que o jQuery é uma biblioteca para simplificar a utilização de JavaScript em diferentes navegadores web.

A criação dos diagramas utilizados para apresentar os *workflows* de análise construídos na plataforma SemanticSCo Web foi implementada utilizando o componente **Diagram**, disponível no PrimeFaces. Este componente foi estendido utilizando CSS e jQuery para criar uma solução personalizada para a plataforma, de modo a simplificar a criação e a execução de *workflows*. Por exemplo, os elementos **Start**, **Datasource** e **Adapter** foram construídos especificamente para a plataforma, não existindo na implementação original do componente **Diagram**. Adicionalmente, a conexão entre dois elementos contidos no *workflow* foi facilitada de modo a permitir que todo um elemento alvo de conexão possa ser utilizado como (ponto) destino de uma conexão. Na implementação original do componente **Diagram**, era necessário especificar um ponto específico (em um elemento) que seria utilizado como destino para a criação de uma conexão entre dois elementos.

A Figura 10 apresenta a nova interface desenvolvida para criação de *workflows* de análise na plataforma SemanticSCo Web. Esta interface está dividida em duas partes. Na parte superior (1) são apresentados o título da plataforma, o nome e a foto do usuário. Ao clicar sobre o nome ou sobre a foto, o usuário pode editar suas informações pessoais ou pode sair da plataforma. Ainda nesta parte, existe uma barra de menus que permite o acesso a diferentes funcionalidades da plataforma. A opção **Dashboard** redireciona o usuário para a interface principal da plataforma. A opção **Workflow** redireciona para a listagem de *workflows* construídos pelo usuário. A opção **Shared** dá acesso aos *workflows* compartilhados com o usuário. Finalmente, a opção **Documentation** exibe a documentação da plataforma.

Figura 10 – Interface para criação de *workflows* de análise na plataforma SemanticSCo Web.

A parte inferior (2) apresenta o painel de exibição de um *workflow*. O elemento **Start** indica o ponto de início de execução do *workflow*. Um retângulo com bordas arredondadas representa uma atividade de análise (serviço). Um retângulo com um ícone na parte superior direita representa um serviço de adaptação. Serviços de adaptação são frequentemente utilizados para a manipulação de dados dentro da plataforma. Um cilindro com múltiplos tons de cinza representa uma fonte de dados utilizada como entrada para as atividades de análise e serviços de adaptação. As arestas entre os elementos indicam a sequência de execução do *workflow*. O ícone azul ao lado de cada elemento indica o ponto de origem utilizado para criar uma aresta entre dois elementos. Para realizar a ligação, o usuário deve clicar sobre o ícone azul e arrastar a aresta até qualquer parte do elemento que deseja utilizar como destino da conexão.

Na parte superior do painel de exibição de um *workflow* existe uma barra de ferramentas contendo seis botões usados na edição de um *workflow*. O botão **Add Data Source** inclui uma nova fonte de dados. O botão **Add Activity** inclui uma nova atividade de análise. Se houver um elemento selecionado quando o botão **Add Activity** for acionado, a descoberta de novas atividades de análise ocorrerá com base nas fontes de dados incluídas no *workflow* e no elemento selecionado. Caso contrário, a descoberta ocorrerá com base em todos os elementos contidos no *workflow* (fontes de dados e atividades de análise). O botão **Service Discovery** descobre os serviços de análise que podem produzir o tipo de dado necessário como entrada para um elemento selecionado. O botão **Remove** apaga um elemento selecionado no *workflow*. Qualquer elemento pode ser apagado, com a exceção

do elemento **Start**. As associações entre os elementos (arestas) podem ser removidas clicando sobre elas e selecionando a opção **Remove** na janela que será exibida. O botão **Execute** é responsável pela execução de um serviço de análise selecionado. Se o elemento selecionado for **Start**, o botão **Execute** irá executar o *workflow* completo, seguindo a ordem de execução definida pelas arestas e a relação de dependência entre os elementos, caso contrário, apenas o serviço de análise selecionado será executado individualmente. Finalmente, o botão **Close** encerra a edição e visualização do *workflow*.

Do lado direito da interface existe um painel de detalhes que permite visualizar e alterar informações dependendo do tipo de elemento selecionado. Por exemplo, o usuário pode alterar o nome de cada um dos elementos contidos no *workflow*, assim como os parâmetros utilizados para a execução dos serviços de análise.

4.2.2 Gene Expression Repository Data

Durante a criação de um *workflow* por meio do módulo **Composite Service Enactment**, o usuário pode optar por incluir dados a partir de um arquivo local ou remoto. Ao optar por um arquivo local, os dados são providos a partir da máquina do usuário. Por outro lado, ao optar por um arquivo remoto, os dados são recuperados a partir do repositório ArrayExpress.

O módulo **Gene Expression Repository Data** é responsável por interagir com o ArrayExpress para a obtenção de dados de experimentos de expressão gênica. Este módulo permite que dados de experimentos publicados no ArrayExpress ou GEO sejam recuperados e estejam disponíveis durante o processo de criação do *workflow* no módulo **Composite Service Enactment**. O ArrayExpress é utilizado como única fonte de dados para recuperar experimentos em ambos os repositórios.

A Figura 11 apresenta a interface para pesquisa por experimentos no ArrayExpress. Ao digitar o identificador de acesso do experimento e clicar sobre o botão **Find experiment**, o usuário terá acesso às informações básicas cadastradas sobre o experimento. Ao clicar sobre o botão **Next**, o usuário poderá selecionar os arquivos contidos no experimento para inclusão no *workflow* (veja Figura 12). Após realizar a seleção de um ou mais arquivos de interesse, o usuário deve especificar o tipo de dados que está sendo carregado para a plataforma. Esta definição é comum ao carregamento de arquivos tanto via seleção

Figura 11 – Interface para pesquisa por experimentos no ArrayExpress

local (**Upload from file**) quanto via seleção remota (**Get file from repository**). A informação sobre o tipo de dados carregado para a plataforma será utilizada no processo de composição semântica.

O módulo **Gene Expression Repository Data** utiliza componentes EJB para realizar a invocação de serviços RESTful disponíveis no repositório ArrayExpress (ver seção 2.2.4). Por meio destes serviços, são obtidas informações sobre os experimentos e arquivos de dados disponíveis no repositório. Tais informações são utilizadas pelo usuário para validar o conjunto de dados sendo carregado.

A plataforma SemanticSCo suporta o uso dos formatos de identificadores de acesso aos experimentos utilizados pelo ArrayExpress e pelo GEO. Uma vez que a pesquisa (via serviços web) por experimentos no ArrayExpress permite apenas o uso de identificadores no formato deste repositório, o módulo **Gene Expression Repository Data** realiza a conversão, de forma transparente ao usuário, de um identificador no formato utilizado pelo GEO para o formato usado no ArrayExpress.

A Figura 13 ilustra por meio de um diagrama de sequência UML as interações associadas à obtenção de dados sobre experimentos de expressão gênica a partir do ArrayExpress. O usuário, por meio do módulo **Composite Service Enactment**, informa à plataforma

Figura 12 – Interface para seleção dos arquivos contidos no experimento que devem ser incluídos no Data Source

Figura 13 – Interação do usuário com o ArrayExpress para a obtenção de dados sobre experimentos de expressão gênica.

que deseja requisitar os dados de um experimento a partir do ArrayExpress. O módulo **Composite Service Enactment** requisita o experimento específico para o módulo **Gene Expression Repository Data**. Este módulo invoca o serviço **experiments** do ArrayExpress. O serviço **experiments** retorna informações básicas sobre o experimento desejado pelo usuário. Na sequência, o usuário então requisita a lista de arquivos do experimento. O módulo **Composite Service Enactment** interage com o **Gene Expression Repository Data** para requisitar a lista de arquivos. O módulo **Gene Expression Repository Data** realiza a invocação do serviço **files** para a obtenção dos nomes dos arquivos. Este módulo lê as informações disponíveis a partir do arquivo SDRF do experimento e as exibe no formato de lista para o usuário. Neste ponto, o usuário da plataforma seleciona os arquivos desejados. O módulo **Composite Service Enactment** então interage com o **Gene Expression Repository Data** para carregar os arquivos desejados pelo usuário. O módulo **Gene Expression Repository Data** carrega os arquivos selecionados e retorna o controle para o módulo **Composite Service Enactment**. Por fim, este módulo persiste os arquivos carregados como fonte de dados no *workflow* do usuário.

4.3 Suporte à Reprodutibilidade

A plataforma SemanticSCo Web provê suporte para a reprodução dos experimentos criados na plataforma. O componente **Composition Manager** do módulo **Composite Service Enactment** e o módulo **Workflow Execution Manager** são os responsáveis pela reprodução de um experimento de análise de expressão gênica.

4.3.1 Composition Manager

O componente **Composition Manager** persiste todas as informações relacionadas ao *workflow* de análise em um banco de dados relacional. As informações persistidas incluem as atividades de análise, o serviço associado a cada atividade, dados de entrada e saída de cada atividade, e os parâmetros de execução de cada serviço. Uma vez que estas informações são persistidas, a plataforma permite que qualquer *workflow* desenvolvido possa ser reproduzido da maneira em que foi projetado e executado pela última vez. A Figura 14 apresenta os componentes envolvidos no processo de persistência de um *workflow*

criado na plataforma SemanticSCo Web.

Ao receber a solicitação de criação de um elemento dentro do *workflow* por meio do componente **Workflow Construction**, o componente **Composition Manager** invoca diferentes componentes EJB para a criação do elemento correspondente (grupo **Diagram Element Beans**). Há um componente EJB para cada tipo de componente passível de manipulação dentro do *workflow*. Por exemplo, os EJBs **DataSourceBean** e **ServiceBean** são responsáveis por controlar as operações envolvendo o elemento fonte de dados (**DataSource**) e serviço de análise (**Service**), respectivamente. Um componente EJB recebe a solicitação para a criação de um elemento e solicita à camada de acesso aos dados (grupo **Data Access Objects**) a persistência do objeto em banco de dados relacional. Esta camada possui componentes implementados conforme o padrão de projeto Data Access Object (DAO) (ALUR; CRUPI; MALKS, 2003), além de entidades que realizam o mapeamento de estruturas de dados (tabelas) para objetos (classes) utilizando recursos do Java Persistence API (JPA) (ORACLE, 2013c). Existe um componente DAO e uma classe de entidade para cada componente EJB. Por exemplo, a classe DAO **ServiceDAO** e a entidade **Service** são responsáveis por atender as requisições do EJB **ServiceBean**. Os componentes DAO utilizam a interface **Entity Manager** do JPA para interagir com o banco de dados MySQL (MYSQL, 2018) a fim de persistir os dados armazenados nas entidades. A recuperação, alteração e remoção de elementos dentro do *workflow* utilizam este mesmo fluxo de execução.

Durante o desenvolvimento e a execução de um *workflow* de análise, o usuário pode incluir anotações pessoais para quaisquer componentes do *workflow*, de modo a descrever o raciocínio utilizado na construção/execução de determinada atividade de análise, bem como descrever eventuais percepções relacionadas. Estas anotações também são persistidas, ficando disponíveis durante a reprodução de um *workflow* previamente construído.

A Figura 15 apresenta a estrutura das tabelas de dados utilizadas na plataforma SemanticSCo. O grupo **Composition and Execution** apresenta as tabelas envolvidas na persistência dos dados associados à composição e à execução de um *workflow* de análise. A tabela **SCOWORKFLOW** armazena as informações básicas do *workflow*. A tabela **SCOELEMENT** armazena as informações comuns a todos os elementos do *workflow* (serviços de análises, fonte de dados e elemento de início). A tabela **SCOCONNECTION** armazena a conexão entre dois elementos do diagrama. A tabela **SCOSTART** representa o elemento de início

Figura 14 – Persistência de dados utilizando o módulo **Composition Manager**.

do *workflow*. A tabela **SCOSERVICE** armazena as informações de um serviço contido no *workflow*. A tabela **SCOINPUT** armazena os parâmetros utilizados pelo usuário durante a execução do *workflow*. A tabela **SCOCONCEPT** armazena informações sobre a anotação semântica utilizada por um serviço. A tabela **SCODATASOURCE** armazena uma fonte de dados utilizada no *workflow*. A tabela **SCOFIELD** armazena um arquivo incluído no *workflow*, quer seja uma fonte de dados utilizada como entrada para um serviço de análise ou dados resultantes do processamento de um serviço. Finalmente, as tabelas **SCODATASOURCE_FILE** e **SCOSERVICE_FILE** armazenam a associação de fonte de dados e serviços de análise com os arquivos de dados.

As tabelas do grupo **User Managment** são responsáveis por guardar as informações do usuário da plataforma. A tabela **SCOUSER** armazena as informações básicas do usuário, enquanto que a tabela **SCOUSERSOCIALNETWORK** armazena as informações sobre quais redes sociais este usuário possui associadas a sua conta de usuário na plataforma. O grupo **Shared** possui uma única tabela **SCOSHARED** que armazena a informação de quais *workflows* estão compartilhados com usuários específicos na plataforma SemanticSCo. Esta tabela também armazena a informação do nível de acesso do usuário a cada *workflow*.

O grupo **General** possui algumas tabelas de uso geral da plataforma. A tabela **SCOFIELD** armazena todos os arquivos na plataforma. Estes arquivos podem ser uma fonte

Figura 15 – Estruturas das tabelas do banco de dados da plataforma SemanticSCo.

de dados utilizada como dados de entrada para um serviço de análise, dados resultantes do processamento de um serviço, ou mesmo uma foto selecionada pelo usuário em seu cadastro na plataforma. A tabela **SCOMESSAGE** armazena as mensagens enviadas pelo administrador da plataforma para um usuário específico. A tabela **SCORECENTACTIVITY** armazena um registro das atividades recentes do usuário na plataforma. Por fim, a tabela **SCOGENERALPARAMETERS** armazena um conjunto de parâmetros gerais configuráveis relacionados à capacidade de armazenamento de dados na plataforma.

Em sua configuração inicial, a plataforma SemanticSCo Web utiliza um limite de 2 Gb de espaço alocado para cada usuário da plataforma. O usuário é informado sobre a quantidade de espaço em uso por seus experimentos na interface inicial. A contabilização do espaço utilizado é realizada levando-se em consideração apenas as fontes de dados (**Data Sources**) incluídas nos *workflows*. Os dados resultantes da execução de um *workflow* de análise não são contabilizados neste limite. A plataforma SemanticSCo armazena os resultados da execução de um *workflow* por 30 dias após a sua execução. Neste período, ao acessar um *workflow* previamente criado, o usuário poderá ter acesso aos dados resultantes sem que exista a necessidade de executar o *workflow* novamente. A qualquer

momento, independentemente dos dados terem sido removidos ou não, o *workflow* poderá ser executado para acesso aos dados. Esta medida foi adotada para otimizar o uso de espaço no servidor da plataforma. Estes parâmetros são configuráveis pelo administrador da plataforma, podendo ser ampliados ou reduzidos conforme a demanda e limitações da infraestrutura utilizada pela plataforma.

4.3.2 Workflow Execution Manager

O módulo **Workflow Execution Manager** é responsável pela execução de serviços mapeados para cada elemento (atividade de análise ou serviço de adaptação) contido no *workflow* do usuário. Cada serviço RESTful disponível na plataforma foi previamente publicado utilizando o módulo **Service Creation and Publication**. As informações publicadas sobre cada serviço são armazenadas no módulo **Service Registry**.

Workflow Execution Manager utiliza o *framework* Activiti (ACTIVITI, 2018) para a leitura e a execução de arquivos BPMN contendo a descrição do comportamento local de cada serviço. Em sua versão inicial, a integração da plataforma SemanticSCo com este *framework* era realizada por meio da codificação de uma classe específica para cada uma das operações locais contidas no arquivo BPMN. Adicionalmente, quando da reengenharia da plataforma SemanticSCo para o ambiente Web, limitações adicionais foram observadas devido à necessidade de interação entre o usuário e a plataforma para a solicitação de parâmetros utilizados na execução de um serviço. Neste sentido, a API SIBRI foi incorporada com a finalidade de sanar tais limitações. Adicionalmente, esta API foi estendida com a inclusão de um conjunto de classes otimizadas para os serviços de análise disponíveis na plataforma SemanticSCo Web.

A Figura 16 apresenta a arquitetura do módulo **Workflow Execution Manager**. Este módulo é formado pelo componente SIBRI, pelas classes de extensão da API e pelo *framework* Activiti, os quais interagem para garantir a execução dos serviços de análise. O módulo **Composite Service Enactment** solicita a execução de um serviço junto ao *framework* Activiti. Este *framework* por sua vez executa um serviço RESTful a partir da especificação BPMN de seu comportamento local. Durante a execução de cada operação deste serviço, uma classe de extensão da API SIBRI é invocada e realiza esta operação utilizando a API SIBRI e o repositório de serviços (GEAS). Quando necessário, a API

Figura 16 – Arquitetura do módulo Workflow Execution Manager.

SIBRI interage com o módulo **Composite Service Enactment**, sinalizando a necessidade de parâmetros de entrada para a execução dos serviços. Quando o usuário fornece tais parâmetros, o fluxo de execução do serviço é reiniciado.

O conjunto de classes que estende a API SIBRI foi definido a partir da constatação de que todos os serviços disponíveis no repositório GEAS possuíam uma estrutura similar, isto é, o conjunto de operações e/ou dinâmica de execução eram basicamente os mesmos. Assim, esta extensão foi definida para simplificar o mapeamento dos serviços publicados na plataforma SemanticSCo. Uma classe foi implementada para suportar cada uma das operações comuns do repositório GEAS: **generateID**, **sendFile**, **getStatus** e **getResult**. A partir da definição destas classes, cada operação BPMN deve ser mapeada diretamente para uma destas classes, e não para as classes genéricas da API SIBRI.

A Figura 17 apresenta um diagrama de classes contendo as classes incorporadas à plataforma SemanticSCo Web. A classe **AbstractServiceTask** pertence ao pacote **core** da API SIBRI e foi incluída aqui apenas para demonstrar a interdependência entre as classes incluídas na plataforma e a API SIBRI.

A operação **generateID**, responsável por gerar um identificador único de execução do serviço no repositório GEAS, é suportada pela classe **GenerateIdOpExecution**. Esta classe realiza uma invocação REST à operação **generateID** via método **GET**. O resultado da invocação do método retorna um identificador de sessão utilizado para as demais

Figura 17 – Interdependência entre as classes incorporadas à plataforma SemanticSCo Web.

invocações ao serviço. Este identificador é atribuído a uma variável de execução do serviço chamada *identifier*. Uma vez definida esta variável, a mesma pode ser reutilizada a qualquer momento por outras operações do serviço.

A operação *sendFile*, responsável por submeter um ou mais arquivos de análise, é suportada pelas classes *SendFileOpExecution* e *SendSingleFromMultipleFileOpExecution*. A classe *SendFileOpExecution* realiza uma invocação POST à operação *sendFile* de um serviço REST. O conteúdo do arquivo é enviado no corpo da requisição POST. Os parâmetros de entrada utilizados nas operações de serviços estão contidos no arquivo de mapeamento BPMN dentro do elemento *ioSpecification* do serviço mapeado. De forma padrão, o envio de arquivos por meio desta classe submete o arquivo para o parâmetro *wsdl:sendFile.file*, que é comum a todos os serviços do repositório GEAS. Contudo, este valor pode ser sobrescrito utilizando-se o parâmetro *sendFileNameParameter* para atribuir um novo valor ao parâmetro. A classe *SendFileOpExecution* também permite que vários arquivos sejam enviados em uma única execução da operação. Para isso, os nomes dos arquivos devem ser enviados utilizando-se o caracter ponto e vírgula (;) como delimitador. A classe *SendSingleFromMultipleFileOpExecution* especializa a classe concreta *SendFileOpExecution* para restringir o envio de mais de um arquivo por operação. Se for detectado que existe mais de um arquivo, é solicitado ao usuário que este selecione apenas um dos arquivos para concluir esta operação.

A operação *getStatus*, responsável por monitorar a execução da atividade de análise fornecida pelo serviço no repositório, é suportada pela classe *GetStatusOpExecution*. Esta classe realiza o monitoramento periódico da execução de um determinado serviço. O monitoramento ocorre analisando o *status* de retorno do serviço. Enquanto este *status*

tiver o valor **running**, a instância da classe `GetStatusOpExecution` irá aguardar até a conclusão da execução. Caso o *status* de retorno tenha o valor **failure**, o monitoramento é interrompido com a indicação de falha na execução do serviço. Caso o *status* de retorno tenha o valor **finished**, o monitoramento é interrompido com a indicação de sucesso na execução do serviço.

Por fim, a operação `getResult`, responsável por obter o resultado da execução do serviço de análise, é suportada pela classe `GetResultOpExecution`. Os parâmetros de saída utilizados nas operações de serviços estão contidos em um arquivo de mapeamento BPMN dentro do elemento `ioSpecification`. De forma padrão, a obtenção do resultado é feita por meio da leitura do parâmetro de saída `wsdl:getResultResponse.return`. O arquivo de saída resultante pode ter seu nome informado utilizando o parâmetro `outputResultResponseParameter`.

4.4 Suporte à Transparência

A plataforma SemanticSCo Web permite que *workflows* construídos na plataforma sejam compartilhados entre seus usuários. Este recurso simplifica a disseminação e a reprodução de um experimento construído na plataforma.

4.4.1 Compartilhamento de *workflows*

O proprietário (criador) de um *workflow* pode disponibilizar o acesso ao *workflow* a um ou mais usuários por meio de uma operação de compartilhamento de *workflow*. O proprietário deste *workflow* também pode optar por tornar o *workflow* público, não impondo nenhuma restrição de acesso. Ao optar pelo compartilhamento público, a plataforma SemanticSCo Web sugere ao proprietário do *workflow* o compartilhamento nas redes sociais Facebook, LinkedIn e Google+. O proprietário pode selecionar se deseja compartilhar ou não o *workflow* em uma ou mais destas redes sociais.

A Figura 18 apresenta uma listagem dos *workflows* criados por um usuário na plataforma SemanticSCo. A coluna **Visibility** indica o estado de compartilhamento de um *workflow*. O estado **Private** representa um *workflow* cujo acesso é restrito ao seu proprietário. O estado **Shared** representa um *workflow* compartilhado com um ou mais

Figura 18 – Lista de *workflows* criados pelo usuário na plataforma SemanticSCo. Em destaque, os botões utilizados para compartilhar um *workflow*.

usuários. Os botões utilizados para alterar o estado de compartilhamento de um *workflow* estão em destaque na figura.

Ao compartilhar um *workflow*, o proprietário é questionado sobre o tipo de compartilhamento desejado. Três opções podem ser selecionadas: **None**, **All users** e **Specific users**. **None** restringe o acesso para o *workflow* ao proprietário do mesmo, sendo a opção padrão definida para todos os *workflows* quando são criados. **All users** permite que um *workflow* seja compartilhado com todos os usuários da plataforma. Por fim, **Specific users** permite que um *workflow* seja compartilhado com um ou mais usuários específicos.

A plataforma suporta dois modos de acesso a um *workflow* compartilhado: **read-only access** e **write access**. O modo **read-only access** é o mais restritivo, permitido apenas que o usuário possa visualizar e executar (reproduzir) o *workflow* compartilhado. Nenhuma alteração realizada no *workflow* é persistida. O modo **write access** permite não apenas reproduzir um *workflow*, mas também realizar modificações no mesmo. De acordo com este modo de acesso, todas as alterações realizadas no *workflow* são persistidas.

O compartilhamento de um *workflow* com todos os usuários (opção **All users**) implica necessariamente na utilização do modo de acesso **read-only access**. Por outro lado, o compartilhamento de um *workflow* com um grupo específico de usuários (opção **Specific users**) permite a definição de um modo de acesso distinto para cada usuário

Figura 19 – Seleção de usuários e modo de acesso para um *workflow* compartilhado para um grupo de usuários.

(*read-only access* ou *write access*). Independentemente do modo em que o *workflow* foi compartilhado, o usuário que possui permissão de acesso ao mesmo pode realizar uma cópia do *workflow* original. Uma vez que realize esta cópia, este usuário poderá modificar livremente sua própria versão do *workflow*.

A Figura 19 apresenta a interface para o compartilhamento de *workflow* com um conjunto específico de usuários. Ao selecionar esta opção, a plataforma exibe uma interface para a busca por usuários cadastrados na plataforma. Esta busca pode ocorrer pelo email (opção **Search by email**) ou pelo nome (opção **Search by name**) do usuário. Um modo de acesso específico pode ser definido para cada usuário. A Figura 19 também apresenta um exemplo da atribuição de diferentes modos de acesso ao mesmo *workflow*. O usuário **User 4** tem acesso somente para leitura (*read-only access*), enquanto que os usuários **User 2** e **User 3** têm acesso de escrita (*write access*) ao *workflow* sendo compartilhado.

A plataforma SemanticSCo Web disponibiliza os *workflows* compartilhados em duas áreas específicas da plataforma: **Shared with me** e **Public sharing**. **Shared with me** apresenta os *workflows* compartilhados com o usuário corrente (“*logado*”) na plataforma. **Public sharing** apresenta os *workflows* compartilhados com todos os usuários da plataforma. A Figura 20 apresenta uma listagem dos *workflows* compartilhados com um dado usuário *logado*.

# Id	Name	Description	Access type	Owner	
17	Experiment 1	-	Write	User 1	
19	Experiment 2	-	Read only	User 1	
26	Study 1	-	Write	User 4	
27	Study 2	-	Read only	User 4	

© 2018 - SemanticSCo - Semantic Service Composition
Biomedical Systems and Services Laboratory - University of São Paulo - Brazil

Figura 20 – Listagem dos *workflows* compartilhados com o usuário *logado*.

4.4.2 Social Network Integration

Social Network Integration é o módulo responsável por permitir a integração da plataforma SemanticSCo com redes sociais. Durante o compartilhamento público de um *workflow*, um *link* de acesso interno é gerado para que o *workflow* possa ser acessado diretamente por usuários fora da plataforma. Quando tais usuários acessam este link, estes são redirecionados para a plataforma para que possam visualizar o *workflow*. Um usuário que não possui cadastro na plataforma SemanticSCo Web pode realizar seu cadastro utilizando informações providas pela própria rede social. Assim, o módulo Social Network Integration tem basicamente duas responsabilidades: compartilhar um *workflow* em um ambiente externo à plataforma e fornecer mecanismos para acesso à plataforma utilizando recursos da própria rede social.

A Figura 21 apresenta a arquitetura do módulo Social Network Integration. Este módulo é composto por quatro componentes: Share Workflow, Social Network Authentication, ScribeJava e Plugins JS. O componente Share Workflow é responsável por criar a URL para um *workflow* compartilhado na plataforma e prover o acesso a este *workflow* de maneira pública. O componente Plugins JS agrega um conjunto de *plugins* JavaScript para permitir o compartilhamento de URLs em uma dada rede social. Para cada rede social suportada, um *plugin* fornecido pela própria rede social foi

Figura 21 – Arquitetura do módulo Social Network Integration.

incorporado a este componente.

O componente **Social Network Authentication** é responsável pela integração da plataforma SemanticSCo com a rede social para realizar a autenticação de um usuário na plataforma. O módulo **Authentication** interage com o componente **Social Network Authentication** para realizar o controle do estado de autenticação de cada usuário. Adicionalmente, por meio do componente **Social Network Authentication**, também é possível solicitar informações básicas que podem ser utilizadas durante o cadastro do usuário. O componente **ScribeJava** (GROMOV; GARANINA, 2018) é uma biblioteca Java contendo um conjunto de métodos auxiliares para intermediar a comunicação entre o componente **Social Network Authentication** e as redes sociais. Este componente foi reutilizado na plataforma sem modificações. O componente **Share Workflow** e o módulo **Authentication** interagem com o banco de dados **MySQL** para obter e/ou persistir informações sobre o estado de um *workflow* e usuários cadastrados, respectivamente.

A Figura 22 ilustra por meio de um diagrama de sequência UML as interações de um usuário para acesso a plataforma SemanticSCo Web. Por meio de seu navegador web, o usuário pode optar por solicitar acesso à plataforma utilizando ou não as redes sociais. Ao optar pelo uso das redes sociais, o usuário interage com o módulo **Authentication** solicitando acesso à plataforma SemanticSCo. Uma solicitação de autenticação via uma dada rede social é enviada ao módulo **Social Network Integration**. Este módulo interage com a rede social solicitando a autenticação do usuário. Uma resposta contendo

Figura 22 – Interação do usuário para acesso a plataforma SemanticSCo Web.

uma interface de autenticação gerada pela própria rede social é retornada para o usuário. O usuário fornece as credenciais de acesso, que são enviadas pelo sistema de autenticação para a rede social. Esta, por sua vez, retorna o *status* de autenticação do usuário, que, quando realizada com sucesso, permite a liberação de acesso do usuário à plataforma. Ao optar por não utilizar as redes sociais, o mecanismo de autenticação é todo controlado pelo módulo **Authentication**, sem que exista a necessidade de interagir com o módulo **Social Network Integration**.

Uma vez autenticado, o usuário pode criar e compartilhar *workflows* na plataforma. A Figura 23 ilustra a interação do usuário para compartilhamento de um *workflow*. O usuário solicita ao módulo **Social Network Integration** o compartilhamento de um *workflow* previamente selecionado por ele. O **Social Network Integration** gera uma URL de acesso ao *workflow* e publica a URL na rede social escolhida pelo usuário.

Figura 23 – Interação do usuário para compartilhamento de *workflows* na plataforma SemanticSCo Web.

A versão atual da plataforma SemanticSCo permite a integração das redes sociais LinkedIn, Facebook e Google+. Porém, devido ao uso da biblioteca *ScribeJava* é possível no futuro ampliar a quantidade de redes. *ScribeJava* possui interfaces simplificadas de acesso a diversas outras redes sociais, tais como Twitter (TWITTER, 2108) e Flickr (FLICKR, 2018).

4.5 Melhorias Adicionais à Plataforma SemanticSCo

Esta seção apresenta modificações adicionais realizadas na plataforma SemanticSCo que não estão diretamente ligadas à acessibilidade, reprodutibilidade e transparência. Tais modificações foram introduzidas na plataforma para melhorar o processo de descoberta dos serviços e a utilização dos serviços de adaptação.

4.5.1 Descoberta de novas atividades de análise

Durante a criação de um *workflow* na plataforma SemanticSCo, o usuário recebe sugestões das atividades de análise que podem ser incluídas em seu *workflow*. Estas atividades são sugeridas com base nas informações semânticas associadas às fontes de dados e serviços já disponíveis em seu *workflow*.

A descoberta de serviços na plataforma SemanticSCo utiliza duas abordagens: *forward* e *backward*. Na abordagem *forward* são descobertos serviços que utilizem como entrada dados já disponíveis no *workflow* atual do usuário, sejam estes diretamente

disponíveis por meio de fontes de dados ou como resultado da execução de algum serviço de análise. Nesta abordagem, sempre todos os elementos contidos no *workflow* são levados em consideração para a descoberta de novos serviços. Na abordagem *backward* são descobertos serviços que possam gerar os dados de entrada necessários para um serviço específico. Nesta abordagem um único elemento é levado em consideração na descoberta de novos serviços.

O processo de descoberta de serviços utilizando a abordagem *forward* pode se tornar lento e computacionalmente custoso à medida em que aumenta o número de elementos presentes em um *workflow*. Este problema ocorre devido à quantidade de informações semânticas associadas a cada atividade de análise e fonte de dados contidos no *workflow* do usuário. Tais informações são consideradas pelo mecanismo de inferência utilizado pela plataforma para a sugestão de novas atividades de análise que podem ser incluídas no *workflow*.

Para minimizar este problema, a plataforma SemanticSCo Web definiu uma segunda abordagem para otimizar a descoberta de serviços *forward*. Na abordagem original, o usuário solicita a inclusão de novas atividades sem selecionar previamente nenhum serviço. Na abordagem otimizada, o usuário seleciona um determinado serviço antes de solicitar a inclusão de novas atividades de análise em seu *workflow*.

A abordagem otimizada permite a descoberta de novos serviços em um tempo relativamente curto, quando comparado a sua versão original (sem selecionar nenhum serviço). A solução utilizada pela plataforma SemanticSCo Web consiste na invocação do comando para descoberta de serviços utilizando apenas informações relacionadas às fontes de dados e ao serviço selecionado no *workflow* de análise.

4.5.2 Refatoração dos serviços de adaptação do repositório GEAS

A plataforma SemanticSCo utiliza um conjunto de serviços de adaptação fornecidos pelo repositório Gene Expression Analysis Services (GEAS) (GUARDIA, 2016). Tais serviços são responsáveis por realizar algum tipo de conversão ou manipulação em dados de expressão gênica utilizados nos serviços de análise. Em sua versão original, existiam seis serviços de adaptação, nomeados AS1 a AS6.

Após a análise das funcionalidades presentes nos diferentes serviços de adaptação,

constatou-se a necessidade de refatorar estes serviços a fim de extrair atividades comuns a mais de um serviço e que poderiam ser reutilizadas em diferentes contextos. Com esta finalidade, os seis serviços de adaptação originais foram refatorados e deram origem a dez novos serviços de adaptação. Adicionalmente, a refatoração destes serviços permitiu a atribuição de nomes mais significativos para os novos serviços.

Refatoração do serviço AS1

O serviço de adaptação **Splitter** foi criado a partir da renomeação do serviço **AS1**. O serviço **Splitter** é responsável por particionar um conjunto de dados normalizados em dois conjuntos de dados separados.

Splitter requer um conjunto de dados de entrada normalizados fornecido em arquivo texto no formato tabular. Cada linha do arquivo representa um gene com seus valores de expressão normalizados. O serviço exige como parâmetros de execução o nome da coluna que identifica os genes e os nomes das colunas que contém os valores de expressão que se deseja manter em cada conjunto de dados. Os dois conjuntos de dados resultantes são gerados com a primeira coluna contendo os identificadores dos genes e as demais colunas os valores selecionados para cada conjunto.

Refatoração do serviço AS2

A Figura 24 apresenta os serviços criados a partir da refatoração do serviço **AS2**. Este serviço deu origem a três novos serviços: **Gene Identification Mapping**, **Up Regulated Gene Extraction** e **Down Regulated Gene Extraction**.

Figura 24 – Refatoração do serviço AS2.

Gene Identification Mapping foi criado com base em uma operação comum utilizada pelos serviços AS2, AS3, AS4 e AS5. **Gene Identification Mapping** é responsável por converter os identificadores de genes utilizados em um conjunto de dados de um tipo origem para um tipo destino. Esta conversão de identificadores de gene permite a utilização dos dados por serviços que restrinjam os dados de entrada a um tipo específico de identificador de gene, diferente do tipo utilizado pelo conjunto de dados a ser convertido. **Gene Identification Mapping** requer dois conjuntos de dados de entrada fornecidos em arquivos de texto no formato tabular. O primeiro conjunto contém os dados dos genes que serão convertidos para o novo tipo de identificador. O segundo conjunto consiste de diferentes identificadores para cada um dos genes contidos no primeiro conjunto.

Up Regulated Gene Extraction e **Down Regulated Gene Extraction** são responsáveis por filtrar um conjunto de dados de expressão diferencial em um conjunto de dados de genes regulados positivamente (*Up Regulated*) e regulados negativamente (*Down Regulated*), respectivamente. Estes serviços requerem como entrada dados de expressão diferencial no formato tabular contendo informações sobre genes. O serviço realiza a filtragem dos dados sobre genes regulados positivamente e negativamente com base nos valores de estatística correspondentes aos genes. O conjunto de dados resultante consiste nos dados dos genes regulados positivamente (serviço **Up Regulated Gene Extraction**) ou regulados negativamente (serviço **Down Regulated Gene Extraction**), filtrados a partir do conjunto de dados de entrada.

Refatoração do serviço AS3

A Figura 25 apresenta a refatoração do serviço AS3. Além do serviço **Gene Identification Mapping**, a refatoração deu origem ao serviço **KEGG Pathway Data Filtering**. Este serviço é responsável por filtrar as vias biológicas da base de dados KEGG a partir de dois conjuntos de dados.

KEGG Pathway Data Filtering requer dois conjuntos de dados como entrada. O primeiro conjunto de dados contém os valores normalizados dos genes. O segundo conjunto de dados contém os dados enriquecidos destes mesmos genes. O conjunto de dados resultante consiste em vários arquivos representando as vias biológicas KEGG encontradas no conjunto de dados enriquecidos. Cada arquivo contém os genes associados a uma dada

Figura 25 – Refatoração do serviço AS3.

via biológica, cujos valores são os dados normalizados associados a cada gene.

Refatoração do serviço AS4

A Figura 26 apresenta os serviços criados a partir da refatoração do serviço AS4. Além do serviço **Gene Identification Mapping**, a refatoração deu origem a dois novos serviços: **Gene Filtering** e **Standard Deviation Normalization**.

Figura 26 – Refatoração do serviço AS4.

Gene Filtering tem por objetivo realizar uma seleção de genes e valores associados com base em dois conjuntos de dados no formato tabular. O primeiro conjunto contém os dados normalizados dos genes. O segundo conjunto contém uma lista de genes a ser utilizada como filtro. O conjunto de dados resultante irá conter apenas os genes do primeiro conjunto de dados, que também estão presentes no segundo conjunto de entrada, bem como seus respectivos valores de expressão gênica.

Standard Deviation Normalization realiza a normalização dos dados contidos em um arquivo de texto tabular com múltiplas colunas. **Standard Deviation Normalization Service** requer um único conjunto de dados de entrada no formato texto tabular. Neste conjunto, cada linha contida no arquivo representa um gene e cada coluna um valor associado. O serviço **Standard Deviation Normalization** irá normali-

zar os dados contidos no arquivo de entrada de acordo com o desvio padrão. O conjunto de dados resultante consiste em um arquivo texto em formato tabular com os valores dos genes normalizados.

Refatoração do serviço AS5

A Figura 27 apresenta o serviço criado a partir da refatoração do serviço AS5. Além do serviço **Gene Identification Mapping**, a refatoração deu origem ao serviço **Sample Data Filtering**. Este serviço é responsável por selecionar colunas em um conjunto de dados normalizados.

Figura 27 – Refatoração do serviço AS5.

Sample Data Filtering requer um conjunto de dados de entrada normalizados fornecido em arquivo texto no formato tabular. O serviço exige como parâmetro de execução o identificador das colunas que se deseja selecionar no conjunto de entrada e manter no conjunto de dados resultante.

Refatoração do serviço AS6

A Figura 28 apresenta os serviços criados a partir da refatoração do serviço AS6. A refatoração deu origem a dois novos serviços: **Cross Matching Up Regulated Identification** e **Cross Matching Down Regulated Identification**.

Figura 28 – Refatoração do serviço AS6.

Cross Matching Up Regulated Identification e Cross Matching Down Regulated Identification são responsáveis por filtrar um ou mais conjuntos de dados de expressão diferencial em um conjunto de dados de genes regulados positivamente (*Up Regulated*) e regulados negativamente (*Down Regulated*), respectivamente. Estes serviços requerem como entrada um ou mais conjuntos de dados de expressão diferencial no formato tabular contendo informações sobre genes regulados positivamente e negativamente. O conjunto de dados resultante consiste nos dados dos genes regulados positivamente (serviço Cross Matching Up Regulated Identification) e regulados negativamente (serviço Cross Matching Down Regulated Identification) comuns a todos os conjuntos de dados utilizados como entrada.

A Tabela 5 sumariza os serviços resultantes da refatoração dos adaptadores AS1 a AS6.

4.6 Considerações finais

Este capítulo apresentou os principais aspectos relacionados à ampliação do suporte à acessibilidade, bem como à introdução do suporte à reprodutibilidade e transparência na plataforma SemanticSCo Web. O suporte à acessibilidade foi ampliado por meio da construção de uma interface web. Esta interface permite o uso de forma simples e intuitiva da plataforma SemanticSCo Web. A integração com o ArrayExpress para obtenção de dados de expressão gênica também é um diferencial que facilita a construção de *workflows* na plataforma. O suporte à reprodutibilidade introduzido na plataforma permite que os *workflows* construídos possam ser persistidos em banco de dados. Estes *workflows* posteriormente podem ser recuperados e executados, permitindo que os experimentos construídos na plataforma possam ser reproduzidos da maneira em que foram originalmente concebidos. O suporte à transparência introduzido na plataforma permite que os *workflows* construídos possam ser compartilhados com outros usuários, com a finalidade de incentivar a colaboração científica entre os mesmos. Neste sentido, o compartilhamento em redes sociais de ampla utilização é um diferencial da plataforma.

O processo de descoberta de novas atividades de análise também foi melhorado na versão web da plataforma. A nova abordagem reduz o tempo de descoberta de novos serviços para *workflows* com muitos elementos. Por fim, a refatoração dos serviços de

Tabela 5 – Novos serviços criados com base nos adaptadores AS1 a AS6.

Nome do serviço	Funcionalidade
Splitter	Particiona um conjunto de dados normalizados em dois conjuntos distintos.
Gene Identification Mapping	Converte os identificadores de genes utilizados em um conjunto de dados por um outro tipo de identificador.
Up Regulated Gene Extraction	Seleciona os genes regulados positivamente a partir de um conjunto de dados de expressão diferencial.
Down Regulated Gene Extraction	Seleciona os genes regulados negativamente a partir de um conjunto de dados de expressão diferencial.
KEGG Pathway Data Filtering	Seleciona as vias biológicas da base de dados KEGG a partir de dois conjuntos de dados de entrada.
Gene Filtering	Seleciona os genes e valores associados com base em dois conjuntos de dados.
Standard Deviation Normalization	Normaliza um conjunto de dados de acordo com o desvio padrão.
Sample Data Filtering	Filtra de um conjunto de dados normalizados as condições experimentais de interesse.
Cross Matching Up Regulated Identification	Extraí de dois ou mais conjuntos de dados de expressão diferencial os genes regulados positivamente comuns a todos os conjuntos utilizados como entrada.
Cross Matching Down Regulated Identification	Extraí de dois ou mais conjuntos de dados de expressão diferencial os genes regulados negativamente comuns a todos os conjuntos utilizados como entrada.

adaptação produziu serviços com nomes mais intuitivos que realizam apenas uma única operação. Esta abordagem permite uma maior reutilização dos serviços de adaptação.

Estudo de caso

De modo a validar o funcionamento da plataforma SemanticSco Web, um estudo de análise de expressão gênica foi realizado com o intuito de demonstrar as principais funcionalidades implementadas na plataforma para permitir a colaboração científica. Este estudo envolveu a reprodução de parte dos resultados obtidos em um trabalho envolvendo a análise comparativa de dados de Microarray Agilent One-color de pacientes com diabetes tipo I e indivíduos saudáveis (controle) (LIMA et al., 2016). Estes dados estão publicamente armazenados no repositório ArrayExpress sob o identificador E-MTAB-2976.

O capítulo está estruturado da seguinte forma: a seção 5.1 apresenta uma visão geral do estudo utilizado para demonstrar o uso das novas funcionalidades da plataforma; a seção 5.2 apresenta os *workflows* construídos na plataforma para realização do estudo; a seção 5.3 apresenta os resultados obtidos com o estudo; e por fim, a seção 5.4 apresenta algumas conclusões.

5.1 Visão geral

O desenvolvimento deste estudo de caso foi dividido em três etapas. Na primeira etapa, um dos autores do estudo original formulou um conjunto de questões de interesse que deveriam ser respondidas por meio do estudo de reprodução. Na segunda etapa, um conjunto de quatro *workflows* foi criado para responder às questões formuladas. Finalmente na terceira etapa, um grupo na rede social Facebook foi criado e a este grupo foram adicionadas as pessoas envolvidas neste estudo de caso, incluindo o autor do estudo original que formulou as questões de interesse. Posteriormente, neste grupo foi compartilhado os *workflows*

desenvolvidos e as respostas às questões formuladas, de modo a permitir a análise e a reprodução dos resultados obtidos.

As questões elaboradas na primeira etapa são apresentadas a seguir:

Questão 1: Quantos genes estão regulados positivamente e regulados negativamente nas amostras de células mesenquimais de pacientes com **Diabetes Tipo-1 (T1D-MS)** em comparação com amostras de células mesenquimais de indivíduos controle (**C-MS**)?

Questão 2: O gene **ADRB3** está regulado positivamente nas amostras **T1D-MS** em comparação com as amostras **C-MS**?

Questão 3: O gene **CXCL12** está regulado positivamente nas amostras **T1D-MS** em comparação com as amostras **C-MS**? Quais outros genes tem perfil de expressão semelhante ao gene **CXCL12**?

Questão 4: O gene **VCAM1** está regulado negativamente nas amostras **T1D-MS** em comparação com as amostras **C-MS**?

Questão 5: Cite alguns genes de moléculas de adesão ("adhesion-related molecules") que estão reguladas positivamente nas amostras **T1D-MS** em comparação com as amostras **C-MS**?

Questão 6: Qual é o perfil de agrupamento dos genes diferencialmente expressos entre as amostras **T1D-MS** e **C-MS**?

Questão 7: Quais são as vias de sinalização **KEGG** enriquecidas dentre os genes regulados positivamente nas amostras **T1D-MS** em comparação com as amostras **C-MS**? E quais são as vias sinalização **KEGG** enriquecidas dentre os genes regulados negativamente?

Questão 8: As vias "**Neuroactive Ligand receptor Interaction**" e "**JAK Stat signaling pathway**", relacionadas ao sistema nervoso simpático, estão reguladas positivamente nas **T1D-MS** em comparação com as amostras **C-MS**?

5.2 Construção dos workflows

Diferentes *workflows* foram desenvolvidos para responder as questões propostas. Estes *workflows* são apresentados de acordo com o objetivo principal das atividades de análise.

5.2.1 Análise diferencial

De modo a responder as questões 1 a 5, um primeiro *workflow* foi desenvolvido para a análise diferencial dos dados de expressão gênica (*Workflow 1*). O processo para construção do *workflow* foi dividido em três etapas: 1) definição das fontes de dados; 2) análise diferencial; e 3) seleção de genes. A Figura 29 apresenta detalhes de cada uma destas etapas.

Figura 29 – Etapas para a criação do *workflow* para a análise diferencial dos dados de Microarray Agilent One-color.

Na etapa 1, um conjunto de dados foi adicionado ao *workflow*. Este conjunto consiste de dados normalizados de Microarray Agilent One-color, contendo amostras de pacientes com Diabetes tipo I e pacientes saudáveis (grupo controle). Estes dados foram obtidos via integração da plataforma SemanticSCo Web com o repositório ArrayExpress (identificador de acesso E-MTAB-2976), sendo anotados semanticamente como Log-transformed One-color Normalized Microarray Data.

A Figura 30 apresenta um conjunto de informações relacionadas ao processo de recuperação dos dados junto ao repositório ArrayExpress, bem como informações sobre a estrutura destes dados. A Figura 30a apresenta informações básicas exibidas ao usuário

sobre o experimento durante a interação da plataforma com o ArrayExpress. A Figura 30b apresenta o arquivo contido no experimento e selecionado para inclusão no *workflow*. A Figura 30c apresenta a estrutura do arquivo de dados normalizados. A coluna **Gene symbol** contém o identificador de cada gene. A coluna **Probe** apresenta o identificador (Probe Id) de uma sonda de DNA associada a um gene. As colunas D-PRE.DM01.127.1.1, D-PRE.DM13.127.1.2, D-PRE.DM07.127.1.3 e D-PRE.DM02.127.1.4 representam os valores normalizados de expressão dos genes dos pacientes com Diabetes tipo I. As colunas CTRL.NM01.452.1.4, CTRL.NM19.454.1.2, CTRL.NM27.454.1.4 e CTRL.NM20.454.1.3 representam os valores normalizados de expressão dos genes do grupo controle.

a)

Upload file

Find experiment

Name

Global gene expression of bone marrow multipotent mesenchymal stromal cells isolated from type 1 diabetes patients and healthy donors

Description

Microarray analysis of bone marrow multipotent mesenchymal stromal cells isolated from type 1 diabetes patients and healthy donors.

Last update

Feb 2, 2017

Organism

Homo sapiens

Provider

Kalil Alves de Lima (kalil.delima@gmail.com)

Next

Cancel

b)

Upload file

Files available

☒ delima2013array.txt

1

Next

Cancel

c)

Gene symbol	Probe	D-PRE.DM01.127.1.1	D-PRE.DM13.127.1.2	D-PRE.DM07.127.1.3	D-PRE.DM02.127.1.4	CTRL.NM01.452.1.4	CTRL.NM19.454.1.2	CTRL.NM27.454.1.4	CTRL.NM20.454.1.3
1/2-SBSRNA4	A_24_P744148	6,471811639	5,917912224	5,938026952	6,320338123	5,45568574	5,34341427	5,511272114	5,37458771
A1BG	A_23_P38816	7,113252165	8,820446547	8,496894857	8,286908988	9,094808626	7,274414015	6,443458061	7,455924877
A1BG-AS1	A_24_P721699	4,742123437	7,359599927	6,755804895	6,309443986	6,756366945	5,716213659	5,386337066	5,771491875
A2LD1	A_32_P110872	8,185945675	7,456577737	7,933397377	8,783763591	8,302830955	8,000466216	7,907129037	7,874764521
A2M	A_23_P116898	5,196804111	6,251664652	8,279235703	6,61432413	11,31529864	9,83561421	7,676494214	10,04622289
...	...	...	...	...	...	...	...	...	...

Figura 30 – Detalhamento da integração com o repositório ArrayExpress para obtenção dos dados do experimento E-MTAB-2976.

Na etapa 2, os dados de expressão gênica dos pacientes são associados ao serviço **Splitter**, responsável por separar os dados normalizados em dois conjuntos distintos, que representam os dados dos pacientes com Diabetes tipo I (conjunto 1) e dados do grupo controle (conjunto 2). Estes dois conjuntos foram então associados ao serviço **Two-sample Student's T-test Analysis**, responsável por realizar a análise diferencial dos dados.

Na etapa 3, os dados resultantes da análise diferencial são associados aos serviços **Up Regulated Gene Extraction** e **Down Regulated Gene Extraction**. Estes serviços são responsáveis por selecionar respectivamente os genes regulados positivamente e negativamente no conjunto de dados de entrada. A Figura 31 apresenta o *Workflow 1*. A Tabela 6 descreve os parâmetros utilizados na execução de cada serviço deste *workflow*.

Figura 31 – *Workflow* para a análise diferencial dos dados de Microarray Agilent One-color.

5.2.2 Clusterização

De modo a responder a questão 6, os *workflows* 2 e 3 foram desenvolvidos. Em um primeiro momento, o objetivo era utilizar os dados resultantes da análise diferencial obtidos no *Workflow 1* para a construção de um único *workflow* para realizar a clusterização. Contudo, dada a grande quantidade de genes diferencialmente expressos (2799 regulados positivamente e 527 regulados negativamente) obtidos a partir do *Workflow 1*, optou-se por criar um *workflow* adicional, com parâmetros de execução mais restritos, de modo a diminuir a quantidade de genes diferencialmente expressos.

O recurso de cópia de *workflows* foi utilizado para criar um novo *workflow* (*Workflow 2*) a partir do *Workflow 1*. O *Workflow 2* difere do *Workflow 1* apenas pelos parâmetros de

Tabela 6 – Parâmetros utilizados para a execução do *Workflow* 1.

Serviço	Parâmetros de execução
Splitter	<ul style="list-style-type: none"> • Label column = Gene symbol • Experimental conditions 1 = D-PRE.DM01.127.1.1;D-PRE.DM13.127.1.2;D-PRE.DM07.127.1.3;D-PRE.DM02.127.1.4 • Experimental conditions 2 = CTRL.NM01.452.1.4;CTRL.NM19.454.1.2;CTRL.NM27.454.1.4;CTRL.NM20.454.1.3
Two-sample Student's T-test Analysis	<ul style="list-style-type: none"> • Correction method = BH (Benjamini Hochberg) • Cut off value = 1 • Significance level = 0.05
Up Gegulated Gene Extraction	<ul style="list-style-type: none"> • Fold change value = 2 • p-value = 0.05
Down Gegulated Gene Extraction	<ul style="list-style-type: none"> • Fold change value = 2 • p-value = 0.05

execução utilizados para os serviços Up Regulated Gene Extration e Down Regulated Gene Extration. Especificamente, o valor do p-value de **0.05** foi alterado para **0.01** em ambos os serviços. Com base nesta alteração, o número de genes diferencialmente expressos foi reduzido significativamente (611 regulados positivamente e 103 regulados negativamente).

O *Workflow* 2 foi então executado e os resultados desta execução utilizados para criar um *workflow* para realizar a clusterização dos dados (*Workflow* 3). O processo para construção do *Workflow* 3 foi dividido em três etapas: 1) definição das fontes de dados; 2) filtragem e normalização; e 3) visualização. A Figura 32 apresenta detalhes de cada uma destas etapas.

Na etapa 1, dois conjuntos de dados foram adicionados ao *workflow*. O primeiro conjunto consiste de dados processados a partir do conjunto de amostras de pacientes com Diabetes tipo I e pacientes saudáveis (grupo controle). Dado que determinados genes possuíam múltiplas sondas de DNA associadas, a média aritmética dos valores de expressão gênica das diferentes sondas de um mesmo gene foi calculada. Este procedimento

Figura 32 – Etapas para a criação do *workflow* para a geração de um dendrograma a partir da clusterização de dados de Microarray Agilent One-color.

foi realizado de forma análoga ao descrito no estudo original (LIMA et al., 2016). O conjunto de dados resultante foi então adicionado ao *workflow* e nomeado **Average Probes Microarray Data**. O segundo conjunto é composto por um arquivo com a listagem de genes regulados positivamente e negativamente obtidos a partir do *Workflow 2* via os serviços de análise **Up Regulated Gene Extraction** e **Down Regulated Gene Extraction**. Este conjunto foi enviado à plataforma e nomeado **Up + Down Regulated Gene**.

Na etapa 2, ambos os conjuntos de dados são associados ao serviço **Gene Filtering**, responsável por selecionar os genes e valores de expressão que estejam presentes em todos os conjuntos de dados utilizados como entrada. Na sequência, os genes e valores de expressão são associados ao serviço **Standard Deviation Normalization**, responsável por normalizar um conjunto de dados de acordo com o desvio padrão.

Na etapa 3, os dados normalizados são então associados ao serviço **Hierarchical Clustering Of Microarray Data**, responsável por realizar a clusterização de um conjunto de dados. Finalmente, o resultado da clusterização é então associado ao serviço **Dendrogram Generation**, responsável por gerar uma imagem para visualização dos dados na forma de um dendrograma.

A Figura 33 apresenta o *workflow 3*. A Tabela 7 descreve os parâmetros utilizados na execução de cada serviço deste *workflow*. Os serviços **Gene Filtering**, **Standard Deviation Normalization** e **Dendrogram Generation** não foram incluídos nesta tabela pois não são parametrizados.

Figura 33 – *Workflow* para a clusterização de dados de Microarray Agilent One-color.

Tabela 7 – Parâmetros utilizados para a execução do *Workflow* 3.

Serviço	Parâmetros de execução
Hierarchical Clustering Of Microarray Data	<ul style="list-style-type: none"> Distance measure = euclidean distance Clustering method = pairwise average-linkage

5.2.3 Enriquecimento funcional

Para responder as questões 7 e 8, um *workflow* para a análise de enriquecimento funcional dos dados de expressão gênica foi desenvolvido (*Workflow* 4). O processo para a construção do *Workflow* 4 foi dividido em três etapas: 1) definição das fontes de dados; 2) substituição de identificadores; e 3) análise de enriquecimento funcional. A Figura 34 apresenta detalhes de cada uma destas etapas.

Na etapa 1, três conjuntos de dados foram adicionados ao *workflow*. Os dois primeiros conjuntos consistem dos genes regulados positivamente e negativamente obtidos a partir do *Workflow* 1 via os serviços de análise Up Regulated Gene Extraction e Down Regulated Gene Extraction, respectivamente.

O terceiro conjunto de dados adicionado ao *workflow* contém um mapeamento entre três tipos de identificadores de genes: Probe Id, Entrez Id e Gene Symbol. Estes dados foram adicionados ao *workflow* a partir de um arquivo local, sendo anotados semanticamente como Gene Identifiers Mapping.

Figura 34 – Etapas para a criação do *workflow* para a análise funcional dos dados de Microarray Agilent One-color.

Na etapa 2, os dados dos genes regulados positivamente e negativamente são então associados a instâncias distintas do serviço **Gene Identification Mapping**, responsável por realizar a troca dos identificadores dos genes (**Gene Symbol** para **Entrez ID**). Esta troca é necessária porque o serviço de enriquecimento funcional disponível na plataforma, baseado na ferramenta DAVID (HUANG; SHERMAN; LEMPICKI, 2009), não aceita o formato de identificador utilizado pelos dados de expressão.

Na etapa 3, os dados resultantes da substituição dos identificadores de genes são associados a instâncias distintas do serviço **Single Enrichment Analysis Of Gene Expression Data**, responsável por realizar a análise de enriquecimento funcional.

A Figura 35 apresenta o *workflow* 4. A Tabela 8 descreve os parâmetros utilizados na execução de cada serviço deste *workflow*. As diferentes instâncias para os serviços **Gene Identification Mapping** e **Single Enrichment Analysis Gene Expression Data** foram incluídos apenas uma vez pois utilizam os mesmos valores de parâmetros.

5.3 Resultados

Os *workflows* foram construídos e executados utilizando os parâmetros descritos na seção anterior. Os resultados obtidos do processamento de cada um deles foram então utilizados para responder a cada uma das questões propostas pelo estudo de caso:

- **Resposta questão 1:** Foram encontrados 2799 genes regulados positivamente e

Figura 35 – *Workflow* para a análise funcional dos dados de Microarray Agilent One-color.

Tabela 8 – Parâmetros utilizados para a execução do *Workflow* 4.

Serviço	Parâmetros de execução
Gene Identification Mapping	<ul style="list-style-type: none"> • Source map column = 2 (Gene Symbol) • Target map column = 1 (Entrez Id)
Single Enrichment Analysis Gene Expression Data	<ul style="list-style-type: none"> • Gene identifier type = ENTREZ_GENE_ID • DAVID-registered e-mail = wilson@usp.br • Threshold = 0.1 • Count = 2 • Enrichment categories = KEGG_PATHWAY

527 genes regulados negativamente.

- **Resposta questão 2:** O gene ADRB3 foi encontrado no conjunto de genes regulados positivamente nas amostras T1D-MSD em comparação com as amostras C-MSD.
- **Resposta questão 3:** O gene CXCL12 não aparece como diferencialmente expresso.
- **Resposta questão 4:** O gene VCAM1 não aparece como diferencialmente expresso.
- **Resposta questão 5:** Os genes CDH9, COL9A2, SEMA4A, ITGA9, CEACAM1, ESAM, SELPLG, ICAM4, COL20A1, ITGAL e HSPG2 foram identificados como regulados positivamente.

- **Resposta questão 6:** Por meio do processamento de dados utilizando dois *workflows* foi possível identificar que os genes formam dois clusters: um cluster que agrupa amostras C-MSD e outro que agrupa amostras T1D-MSD. Dada a impossibilidade de incluir no texto a representação gráfica do perfil de agrupamento gerado devido à quantidade de genes, um *workflow* adicional, chamado de *Workflow 3a* foi criado com base em um subconjunto dos genes diferencialmente expressos obtidos no *Workflow 2*. Este subconjunto foi formado pelos 40 genes com maior variação diferencial (20 regulados positivamente e 20 regulados negativamente). O *Workflow 3a* foi criado a partir de uma cópia do *Workflow 3*, tendo como única alteração a fonte de dados Up + Down Regulated Gene, que foi substituída pelo conjunto dos 40 genes com maior variação diferencial. A Figura 36 ilustra o dendrograma gerado a partir da clusterização destes 40 genes diferencialmente expressos.
- **Resposta questão 7:** Foram encontradas 35 vias reguladas positivamente ao todo, das quais 21 vias possuem $p\text{-value} < 0.05$. As 10 vias com o $p\text{-value}$ mais significativo (menor $p\text{-value}$) foram: Oxidative phosphorylation, Alzheimer's disease, Huntington's disease, Neurotrophin signaling pathway, Fc gamma R-mediated phagocytosis, Parkinson's disease, Thyroid hormone signaling pathway, Estrogen signaling pathway, hProlactin signaling pathway, e Peroxisome. Também foram encontradas 4 vias reguladas negativamente ao todo, das quais 2 vias possuem $p\text{-value} < 0.05$: mRNA surveillance pathway e Endocytosis.
- **Resposta questão 8:** A via "Neuroactive Ligand receptor Interaction" foi identificada como regulada positivamente, enquanto a via "JAK Stat signaling pathway" não foi identificada.

Após a análise da execução de cada *workflow*, as respostas para cada uma das questões foram incluídas como anotações em cada um dos *workflows* envolvidos. Na sequência, um grupo privado (grupo secreto) para discussão dos resultados foi criado utilizando a rede social Facebook. Este grupo foi chamado **Study MSD-T1D vs MSD-C** e continha pessoas envolvidas neste estudo de caso. Os *workflows* desenvolvidos e as respostas às questões formuladas foram então compartilhadas de modo a permitir a análise e a eventual reprodução dos resultados obtidos. A Figura 37 apresenta um dos *workflows* compartilhados no grupo do Facebook. A imagem do *workflow* apresentada é meramente

Figura 36 – Dendrograma gerado a partir do resultado da clusterização dos 40 genes com maior variação diferencial.

ilustrativa, sendo utilizada por todos os *workflows* compartilhados pela plataforma SemanticSCo Web. Quando o usuário clica sobre o *workflow*, este é redirecionado para o site da plataforma, onde é possível visualizar e executar o mesmo.

Figura 37 – *Workflow 4* compartilhado no grupo Study MSC-T1D vs MSC-C do Facebook.

5.4 Considerações finais

Neste capítulo foi apresentado um estudo de análise de expressão gênica com o intuito de demonstrar as principais funcionalidades implementadas na plataforma para permitir a colaboração científica. Recursos como a construção e execução de *workflows* foram descritos de maneira a reproduzir parte dos resultados obtidos do trabalho publicado na literatura, envolvendo a análise comparativa de dados de Microarray Agilent One-color (LIMA et al., 2016). Para a reprodução de parte do trabalho, uma divisão do estudo em diferentes partes permitiu a modelagem de *workflows* mais intuitivos, além de permitir o compartilhamento individual dos resultados de análise associados aos diferentes *workflows*.

Em linhas gerais, a maior parte dos resultados encontrados foi compatível com o estudo original. Alguns genes do estudo (LIMA et al., 2016), como por exemplo, os genes CXCL12, CD86, JAM2, CNTN1 e VCAM1 não aparecem como diferencialmente expressos em nosso estudo de reprodução. Tal fato pode ser explicado pelo uso de diferentes valores

de parâmetros na execução da ferramenta responsável pela análise diferencial (Pacote R Limma), uma vez que estes valores não estão detalhados no estudo original. Ainda, cabe ressaltar que a quantidade de genes diferencialmente expressos identificados no nosso estudo de reprodução não pode ser diretamente comparada com a quantidade de sondas de DNA diferencialmente expressas informada no estudo original, uma vez que múltiplas sondas de DNA podem estar relacionadas a um mesmo gene.

Os recursos disponíveis na plataforma SemanticSCo Web simplificam a obtenção de dados por meio da integração com o repositório ArrayExpress. O compartilhamento de *workflows* utilizando as redes sociais permite a divulgação dos resultados e favorece a interação entre diferentes usuários para trocas de experiências. Cada *workflow* construído na plataforma pode ser reproduzido por qualquer usuário com acesso aos mesmos.

Outros estudos adicionais também foram realizados com base em experimentos realizados na versão original da plataforma SemanticSCo e documentados em (GUARDIA, 2016). A reprodução destes estudos neste trabalho teve por objetivo, além de testar a nova versão da plataforma, verificar se os resultados produzidos na versão web são similares aos obtidos na versão original da plataforma SemanticSCo. Estes estudos estão descritos nos apêndices B, C e D.

Conclusão

Este projeto teve como objetivo propor um conjunto de extensões à plataforma SemanticSCo para ampliar sua acessibilidade e prover suporte à pesquisa científica colaborativa. Este capítulo apresenta as principais contribuições e limitações do trabalho, além de realizar uma comparação com outras plataformas de análise genômica. Por fim, uma lista de trabalhos futuros é apresentada.

O capítulo está estruturado da seguinte forma: a seção 6.1 apresenta as principais contribuições do trabalho; a seção 6.2 discute o diferencial, as limitações e principais aplicações do trabalho; por fim, a seção 6.3 apresenta os trabalhos futuros.

6.1 Principais contribuições

As principais contribuições deste trabalho incluem a criação da API SIBRI e a criação da nova versão da plataforma SemanticSCo.

6.1.1 API SIBRI

A plataforma SemanticSCo utiliza a linguagem BPMN para representar a ordem na qual as operações de um serviço RESTful devem ser executadas. Na versão original da plataforma, diferentes classes eram utilizadas no suporte à execução de cada operação de um dado serviço. Assim, a inclusão de novos serviços exigia a codificação de classes específicas, com pouco ou nenhum reuso das classes já existentes.

A API SIBRI foi desenvolvida para simplificar e padronizar a adição de novos serviços à plataforma SemanticSCo. Esta API possui classes genéricas e customizáveis que tornam desnecessária a codificação de classes específicas para a adição de um novo serviço na plataforma. O uso da API simplifica o processo de invocação dos serviços RESTful utilizando o *framework* Activiti para diferentes domínios de aplicação.

Adicionalmente, a API SIBRI provê mecanismos padrão para a interação com um usuário em um ambiente web durante a execução de um serviço. Esta API também permite o desenvolvimento de soluções específicas de interação para as situações onde as classes genéricas não proveem suporte adequado para um dado problema. Todos estes mecanismos contribuíram para o desenvolvimento de uma versão mais acessível e adaptável da plataforma SemanticSCo Web.

6.1.2 Plataforma SemanticSCo Web

A incorporação de suporte à reprodutibilidade e transparência, aliado a uma maior acessibilidade, contribuíram para ampliar e simplificar o uso da plataforma SemanticSCo Web.

A acessibilidade da plataforma foi ampliada por meio do desenvolvimento de uma nova interface web, simples e intuitiva. Tal interface elimina a necessidade de execução de uma aplicação local para a execução da plataforma e favorece a evolução de futuras versões, centralizando em um único ponto todas eventuais alterações e a inclusão de novas funcionalidades. Esta abordagem também permite que novos serviços publicados sejam distribuídos imediatamente para todos os usuários.

Melhorias no processo de construção de *workflows* de análise também foram introduzidas na plataforma SemanticSCo. A nova interface provê componentes exclusivos para a criação do *workflow*, que, em conjunto com uma maior visibilidade dos detalhes relacionados a cada um dos elementos contidos no *workflow* (conjuntos de dados de entrada e saída, parâmetros de execução e anotações), proporcionam uma visão ampla do processo de criação e execução de um *workflow*.

A integração com o repositório ArrayExpress facilita a obtenção de dados de um experimento para que estes possam ser utilizados em um *workflow* de análise. Adicionalmente, a criação de um conversor de identificador de acesso do GEO para ArrayExpress

permite que os usuários da plataforma utilizem dados contidos nestes dois repositórios de forma transparente e centralizada.

A reprodução de um experimento permite que um dado estudo de expressão gênica possa ser repetido por qualquer usuário com os mais diversos propósitos. Adicionalmente, todos os *workflows* criados na plataforma podem ser compartilhados na própria plataforma ou em redes sociais, como Facebook, LinkedIn e Google+. O compartilhamento de um estudo realizado na plataforma contribui para a divulgação dos resultados obtidos, permitindo que outras pessoas acessem os estudos e possam reproduzir e/ou modificar os estudos originais.

Adicionalmente, melhorias foram introduzidas no processo de descoberta de serviços na versão web da plataforma SemanticSCo. Estas melhorias possibilitam um melhor desempenho deste processo, principalmente quando do aumento da quantidade de elementos presentes no *workflow*. Ainda, a flexibilidade de utilizar informações semânticas de um único elemento selecionado ou de todos os elementos contidos no *workflow* permite que o usuário escolha a melhor estratégia frente as suas necessidades durante o processo de construção do *workflow*.

Por fim, a refatoração dos serviços de adaptação do repositório GEAS criou novos serviços com nomes mais intuitivos e que podem ser reutilizados com maior facilidade. Estes serviços estão disponíveis tanto durante a construção de *workflows* na plataforma SemanticSCo Web, quanto pela invocação direta via serviço RESTful ao repositório GEAS.

6.2 Discussão

Este trabalho visou o desenvolvimento de uma nova versão da plataforma SemanticSCo com vistas à ampliação da acessibilidade e introdução de suporte para a pesquisa científica colaborativa. Um estudo de caso também foi desenvolvido para demonstrar como as novas funcionalidades facilitaram o uso da plataforma e permitiram a reprodução e o compartilhamento dos experimentos desenvolvidos.

Plataformas de análise com alta acessibilidade, tais como Galaxy (AFGAN et al., 2016), RAP (D'ANTONIO et al., 2015), TRUFA (KORNOBIS et al., 2015) e Babelomics (ALONSO et al., 2015), proporcionam um ambiente adequado ao uso por usuários sem

conhecimentos computacionais avançados. Por outro lado, plataformas de análise com baixa acessibilidade, tais como Bio-jETI (LAMPRECHT; MARGARIA; STEFFEN, 2009) e Taverna (WOLSTENCROFT et al., 2013) exigem de seus usuários o conhecimento de diferentes aspectos sintáticos e semânticos das ferramentas utilizadas. O desenvolvimento de uma interface web simples e intuitiva, assim como a introdução de suporte para a integração com o ArrayExpress ampliaram a acessibilidade da plataforma, facilitando o uso e o aprendizado das funcionalidades contidas na mesma.

Embora a reprodutibilidade de um estudo seja altamente desejável, nem todas as plataformas de análise de expressão gênica proveem tal funcionalidade. Por exemplo, as plataformas RAP, TRUFA e Babelomics não proveem suporte à reprodutibilidade por terceiros dos experimentos criados em seus ambientes. A introdução de suporte a reprodutibilidade na plataforma SemanticSCo Web tornou possível a recuperação e reprodução dos experimentos desenvolvidos nesta plataforma, tal qual estes foram concebidos e executados pela última vez. Para que um experimento de análise possa ser reproduzido, basta que o mesmo seja compartilhado.

Ao permitir que estudos realizados dentro de uma plataforma integrada de análise possam ser compartilhados com outros usuários, uma plataforma aumenta a transparência das informações e dos resultados obtidos em seus experimentos. Esta transparência pode ser encontrada em plataformas como Galaxy e Taverna. A plataforma Galaxy possui uma área interna onde os seus experimentos podem ser compartilhados com outros usuários, enquanto que a plataforma Taverna compartilha seus *workflows* utilizando a rede social myExperiment (GOBLE et al., 2010). A plataforma SemanticSCo Web, além de contar com uma área interna onde os *workflows* podem ser compartilhados, pode realizar o compartilhamento de seus *workflows* em redes sociais de ampla utilização. Tal facilidade é inédita em plataformas integradas de análise de expressão gênica e representa uma facilidade adicional para o compartilhamento de um experimento, dado o uso crescente de redes sociais de propósito geral entre pesquisadores.

Apesar da introdução de um conjunto de facilidades/recursos para a ampliação da acessibilidade e da introdução de suporte à reprodutibilidade e transparência, uma avaliação sistemática da usabilidade da plataforma não foi realizada. Esta limitação não invalida o trabalho pois a realização de tal avaliação tem por objetivo identificar pontos de eventuais melhorias, não descartando todas as facilidades e recursos já introduzidos.

A utilização simultânea da plataforma por usuários processando grandes conjuntos de dados exige uma infraestrutura de processamento e memória robustos, incomuns à infraestrutura encontrada em outros sistemas web. Embora a robustez da plataforma não tenha sido avaliada, isso poderá ser realizado futuramente quando a plataforma se tornar operacional, facilitando a identificação e correção de eventuais problemas. Nesta avaliação, deverão ser considerados os tamanhos médios de *workflows* construídos, além da quantidade de usuários simultâneos utilizando a plataforma.

A possibilidade de modificações nas interfaces com serviços providos por terceiros à plataforma SemanticSCo Web pode comprometer seu bom funcionamento. Tal limitação também não invalida a extensão proposta, dado que manutenções são comuns a sistemas que incorporam recursos fornecidos por terceiros. No evento de alterações nestas interfaces, a plataforma deverá ser modificada para manter o suporte adequado. Adicionalmente, por se tratar de um sistema web, a manutenção da plataforma SemanticSCo Web é facilitada dado que eventuais correções, uma vez disponibilizadas, estão imediatamente disponíveis a todos os usuários.

Na plataforma SemanticSCo Web, quando um usuário compartilha publicamente um *workflow* em uma rede social, este não sabe quais usuários acessam seu *workflow*, uma vez que a informação de acesso não é registrada pela plataforma. O compartilhamento público abre muitas possibilidades, como, por exemplo, ao escrever um artigo, os autores poderiam disponibilizar publicamente seu experimento na plataforma. Com isso, um potencial avaliador deste trabalho poderia acessar o *workflow* com a finalidade de reproduzir e validar os resultados obtidos. Por outro lado, outros pesquisadores também poderiam utilizar o *workflow* disponibilizado como base para o desenvolvimento de novas pesquisas a partir dos resultados já alcançados.

6.3 Trabalhos futuros

Acreditamos que as evoluções incorporadas para a plataforma SemanticSCo Web representam uma solução adequada para a pesquisa científica reproduzível e colaborativa. Apesar dos grandes avanços introduzidos, a plataforma ainda pode ser aprimorada de diferentes formas.

A primeira melhoria a ser introduzida na plataforma seria a implementação do conceito de estudo de expressão gênica. Um estudo de expressão gênica tem por objetivo englobar um conjunto de *workflows* relacionados. Tal facilidade seria análoga ao conceito de projeto, encontrado na maior parte dos ambientes integrados de desenvolvimento de *software*. Nestes ambientes, um projeto é composto por vários arquivos. Quando um estudo de expressão gênica fosse compartilhado, todos os *workflows* desenvolvidos já estariam disponíveis de forma organizada.

Atualmente, a plataforma também carece de mecanismos para a administração de usuários e serviços. De maneira a sanar esta limitação, uma área administrativa pode ser construída a fim de visualizar e gerenciar os usuários cadastrados, assim como prover ferramentas de interação entre o administrador da plataforma e os usuários cadastrados. Nesta área também poderá ser criada uma interface para configurar as parametrizações da plataforma, como por exemplo, o tamanho máximo de espaço físico disponível para os *workflows* dos usuários. Todas as estruturas de dados para gerenciar estas atividades foram criadas e na versão atual precisam ser manipuladas diretamente. Adicionalmente, mecanismos para a inclusão de novos serviços de análise na plataforma por qualquer usuário poderiam ser incluídos. Na versão atual, apenas o administrador da plataforma pode incluir novos serviços de análise para que fiquem disponíveis a todos usuários. A inclusão destes mecanismos aumentaria a capacidade de análise provida pela plataforma.

Pretendemos realizar uma avaliação sistemática da plataforma por um conjunto de potenciais usuários. Esta avaliação permitirá aprimorar ainda mais a experiência do usuário com a plataforma. Para este objetivo, pretendemos utilizar uma metodologia para a avaliação da usabilidade de sistemas computacionais, como, por exemplo, a metodologia System Usability Scale (SUS) (BROOKE, 2013).

Finalmente, pretendemos aprimorar o mecanismo de descoberta de serviços da plataforma. As melhorias introduzidas na versão web da plataforma contribuíram para a melhoria do desempenho da plataforma quando da introdução de novos serviços/atividades de análise. Porém, novas formas de composição de serviços e novos algoritmos podem ser investigados.

Referências

ACTIVITI. *Activiti BPM Platform*. 2018. Disponível em: <<http://activiti.org>>. Acesso em: 10 ago. 2018.

AFGAN, E. et al. The galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Research*, v. 44, n. W1, p. W3–W10, July 2016.

ALONSO, R. et al. Babelomics 5.0: functional interpretation for new generations of genomic data. *Nucleic acids research*, v. 43, n. W1, p. W117–21, jul 2015.

ALUR, D.; CRUPI, J.; MALKS, D. *Core J2EE Patterns: Best Practices and Design Strategies*. [S.l.: s.n.], 2003. ISBN 978-0131422469.

ATTWOOD, T. et al. *Concepts, Historical Milestones and the Central Place of Bioinformatics in Modern Biology: A European Perspective*. [S.l.: s.n.], 2011. 1–38 p. ISBN 978-9533072821.

BARRETT, T. et al. NCBI GEO: mining tens of millions of expression profiles–database and tools update. *Nucleic Acids Research*, v. 35, n. Database, p. D760–D765, 2007.

BELL, G.; HEY, T.; SZALAY, A. Computer science: Beyond the data deluge. *Science*, v. 323, n. 5919, p. 1297–1298, 2009.

BLEDA, M. et al. CellBase, a comprehensive collection of RESTful web services for retrieving relevant biological information from heterogeneous sources. *Nucleic acids research*, v. 40, n. Web Server issue, p. W609–14, jul 2012.

BOLCHINI, D. et al. Better bioinformatics through usability analysis. *Bioinformatics (Oxford, England)*, v. 25, n. 3, p. 406–12, 2009.

BRAZMA, A. Minimum Information About a Microarray Experiment (MIAME) – Successes, Failures, Challenges. *The Scientific World JOURNAL*, v. 9, p. 420–423, 2009.

BRAZMA, A. et al. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat Genet*, v. 29, n. december, p. 365–371, 2001.

BROOKE, J. SUS: A Retrospective. *Journal of Usability Studies*, v. 8, p. 29–40, 2013.

BUTTE, A. The use and analysis of microarray data. *Nature reviews. Drug discovery*, v. 1, n. 12, p. 951–960, dec 2002.

- D'ANTONIO, M. et al. RAP: RNA-Seq Analysis Pipeline, a new cloud-based NGS web application. *BMC genomics*, BioMed Central Ltd, v. 16, n. Suppl 6, p. S3, 2015.
- DROPBOX. *Dropbox*. 2018. Disponível em: <<https://www.dropbox.com>>. Acesso em: 10 ago. 2018.
- EMBL-EBI. *Annotare 2.0*. 2012. Disponível em: <<http://www.ebi.ac.uk/fg/annotare>>. Acesso em: 10 ago. 2018.
- EMBL-EBI. *ArrayExpress – functional genomics data*. 2015. Disponível em: <<http://www.ebi.ac.uk/arrayexpress>>. Acesso em: 10 ago. 2018.
- EMBL-EBI. *European Nucleotide Archive (ENA) - Supported read file formats*. 2018. Disponível em: <<https://www.ebi.ac.uk/ena/submit/read-file-formats>>. Acesso em: 10 ago. 2018.
- FACEBOOK. *Facebook*. 2018. Disponível em: <<http://www.facebook.com>>. Acesso em: 10 ago. 2018.
- FGED. *MINSEQE: Minimum Information about a high-throughput SEQuencing Experiment*. 2008. Disponível em: <<http://fged.org/projects/minseque>>. Acesso em: 10 ago. 2018.
- FGED. *MAGE-TAB Specification Version 1.1*. [S.l.]: FGED, 2011.
- FLICKR. *Flickr*. 2018. Disponível em: <<https://www.flickr.com/>>. Acesso em: 10 ago. 2018.
- GOBLE, C. A. et al. myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic acids research*, v. 38, n. Web Server issue, p. W677–82, jul 2010.
- GOECKS, J. et al. Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, v. 11, n. 8, p. R86, 2010.
- GOOGLE. *Google+*. 2018. Disponível em: <<https://plus.google.com>>. Acesso em: 10 ago. 2018.
- GROMOV, S.; GARANINA, I. *Scribe Java (OAuth library)*. 2018. Disponível em: <<https://github.com/scribejava>>. Acesso em: 10 ago. 2018.
- GUARDIA, G. D. A. *Suporte ao desenvolvimento e à composição de serviços web semânticos para análise de expressão gênica*. Tese (Doutorado) — Universidade de São Paulo, 2016.
- GUARDIA, G. D. A. et al. SemanticSCo: A platform to support the semantic composition of services for gene expression analysis. *Journal of Biomedical Informatics*, v. 66, p. 116–128, 2017.
- GUARDIA, G. D. A. et al. A Methodology for the Development of RESTful Semantic Web Services for Gene Expression Analysis. *Plos One*, v. 10, n. 7, p. e0134011, 2015.
- HEY, T.; TANSLEY, S.; TOLLE, K. *The Fourth Paradigm: Data-Intensive Scientific Discovery*. 1. ed. [S.l.]: Microsoft Research, 2009. 17–31 p. ISBN 978-0982544204.

- HUANG, D.; SHERMAN, B. T.; LEMPICKI, R. A. Systematic and integrative analysis of large gene lists using DAVID Bioinformatics Resources. *Nature*, p. 44–57, 2009.
- IETF. *The OAuth 2.0 Authorization Framework*. 2012. Disponível em: <<https://tools.ietf.org/html/rfc6749>>. Acesso em: 10 ago. 2018.
- ISO; IEC; IEEE. *Systems and software engineering – Vocabulary*. [S.l.: s.n.], 2010. ISBN 978-0738162058.
- JQUERY. *jQuery Foundation*. 2018. Disponível em: <<https://jquery.com>>. Acesso em: 10 ago. 2018.
- KOLESNIKOV, N. et al. ArrayExpress update—simplifying data submissions. *Nucleic acids research*, v. 43, n. Database issue, p. D1113–6, 2015.
- KORNOBIS, E. et al. TRUFA: A User-Friendly Web Server for de novo RNA-seq Analysis Using Cluster Computing. *Libertas Academia - Freedom to Research*, p. 97–104, 2015.
- LAMPRECHT, A.-L.; MARGARIA, T.; STEFFEN, B. Bio-jETI: a framework for semantics-based service composition. *BMC Bioinformatics*, v. 10, n. Suppl 10, p. S8, 2009.
- LAURETTE, P. et al. Transcription factor mitf and remodeller brg1 define chromatin organisation at regulatory elements in melanoma cells. *eLife*, 2015.
- LIMA, K. A. de et al. Transcriptional profiling reveals intrinsic mrna alterations in multipotent mesenchymal stromal cells isolated from bone marrow of newly-diagnosed type 1 diabetes patients. *Stem Cell Research & Therapy*, 2016.
- LINKEDIN. *LinkedIn*. 2018. Disponível em: <<https://www.linkedin.com>>. Acesso em: 10 ago. 2018.
- LOPES, J. S.; ABREU, R. A. de; OLIVEIRA, R. F. Brain transcriptomic response to social eavesdropping in zebrafish (*danio rerio*). *PLOS ONE*, 2015.
- MARGARIA, T.; KUBCZAK, C.; STEFFEN, B. Bio-jETI: a service integration, design, and provisioning platform for orchestrated bioinformatics processes. *BMC bioinformatics*, v. 9 Suppl 4, p. S12, 2008.
- MESIROV, J. P. Accessible reproducible research. *Science*, v. 327, n. 5964, p. 415–416, 2010.
- MISLOVE, A. et al. Measurement and Analysis of Online Social Networks. *5th ACM/USENIX Internet Measurement Conference (IMC'07)*, v. 40, n. 6, p. 29, 2007.
- MYSQL. *MySQL*. 2018. Disponível em: <<https://www.mysql.com>>. Acesso em: 10 ago. 2018.
- NCBI. *Gene Expression Omnibus*. 2016. Disponível em: <<http://www.ncbi.nlm.nih.gov/geo>>. Acesso em: 10 ago. 2018.
- NOORDEN, R. V. Online collaboration: Scientists and the social network. *Nature*, v. 512, n. 7513, p. 126–129, 2014.
- OASIS. *Universal Description Discovery and Integration (UDDI) Version 3.0.2*. [S.l.]: Oasis, 2004.

OLIVEIRA, G. L. V. D. et al. Bone marrow mesenchymal stromal cells isolated from multiple sclerosis patients have distinct gene expression profile and decreased suppressive function compared with healthy counterparts. *Cell Transplantation*, v. 24, p. 151–165, 2015.

ORACLE. *JavaServer™ Faces Specification Version 2.2*. [S.l.]: Oracle, 2013.

ORACLE. *Java™ Platform, Enterprise Edition (Java EE) Specification, v7*. [S.l.]: Oracle, 2013.

ORACLE. *JSR 338: Java™ Persistence API, Version 2.1*. [S.l.]: Oracle, 2013.

ORACLE. *JSR 345: Enterprise JavaBeans™, Version 3.2 EJB Core Contracts and Requirements*. [S.l.]: Oracle, 2013.

ORACLE. *The Java Language Specification Java SE 8 Edition*. [S.l.]: Oracle, 2015.

PRIMEFACES. *PrimeFaces Ultimate UI Framework for Java EE*. 2018. Disponível em: <<http://www.primefaces.org>>. Acesso em: 10 ago. 2018.

RESEARCHGATE. *ResearchGate*. 2018. Disponível em: <<https://www.researchgate.net>>. Acesso em: 10 ago. 2018.

SHEARER, E.; GOTTFRIED, J. *News Use Across Social Media Platforms 2017*. 2017. Disponível em: <<http://www.journalism.org/2017/09/07/news-use-across-social-media-platforms-2017>>. Acesso em: 10 ago. 2018.

SLONIM, D. K.; YANAI, I. Getting started in gene expression microarray analysis. *PLoS Computational Biology*, v. 5, n. 10, p. e1000543, 2009.

TWITTER. *Twitter*. 2108. Disponível em: <<https://www.twitter.com>>. Acesso em: 10 ago. 2018.

UCSC. *UCSC Genome Browser Home*. 2002. Disponível em: <<https://genome.ucsc.edu>>. Acesso em: 10 ago. 2018.

W3C. *Semantic Annotations for WSDL and XML Schema*. [S.l.]: W3C, 2007.

W3C. *CSS Snapshot 2015*. [S.l.]: W3C, 2015.

W3C. *HTML5*. 2017. Disponível em: <<https://www.w3.org/TR/html5>>. Acesso em: 10 ago. 2018.

WIT, P. D. et al. The simple fool's guide to population genomics via RNA-Seq: an introduction to high-throughput sequencing data analysis. *Molecular Ecology Resources*, v. 12, n. 6, p. 1058–1067, 2012.

WOLF, J. B. W. Principles of transcriptome analysis and gene expression quantification: An RNA-seq tutorial. *Molecular Ecology Resources*, v. 13, n. 4, p. 559–572, 2013.

WOLSTENCROFT, K. et al. The taverna workflow suite: designing and executing workflows of web services on the desktop, web or in the cloud. *Nucleic Acids Research*, v. 41, n. W1, p. W557–W561, 2013.

ZHANG, H.; CHOUDHURY, M. D.; GRUDIN, J. Creepy but inevitable?: the evolution of social networking. *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing - CSCW '14*, p. 368–378, 2014.

Plataforma SemanticSCo Web

Este apêndice apresenta uma visão geral da interface gráfica da plataforma SemanticSCo Web, descrevendo de maneira simplificada as funcionalidades contidas nas principais interfaces do sistema.

A.1 Interface de autenticação

A Figura 38 apresenta a interface de autenticação da plataforma SemanticSCo Web. O usuário pode optar por fornecer email e senha para acesso, ou mesmo realizar a autenticação

The screenshot displays the login page of the SemanticSCo Web platform. At the top, a dark blue header contains the 'SEMANTIC SCo' logo and the text 'Semantic Services Composition'. The main content area has a light blue background. On the left, a text block states 'Built for Biologists and Bioinformaticians' and describes the platform's support for RESTful services in bioinformatics. On the right, a white box titled 'Log Into SemanticSCo' contains a login form with fields for 'Email address' and 'Password', a 'Login' button, and social media login options (LinkedIn, Facebook, Google). Below the form, a link says 'Don't have an account? Create one now!'. The footer is dark blue and includes copyright information for 2017, the Biomedical Systems and Services Laboratory at the University of São Paulo, and the USP logo.

Figura 38 – Interface de autenticação da plataforma SemanticSCo.

utilizando uma das redes sociais suportadas: LinkedIn, Facebook ou Google+. Caso o usuário não possua um cadastro prévio na plataforma, também poderá realizar seu cadastro, fornecendo email e senha, ou mesmo utilizando suas informações de cadastro já disponíveis em uma das redes sociais suportadas.

A.2 Interface inicial

A Figura 39 apresenta a interface inicial da plataforma SemanticSCo Web. Na parte superior são apresentados o título da plataforma, nome e foto do usuário logado. Ao clicar sobre o nome ou a foto, o usuário pode editar suas informações pessoais ou sair da plataforma. Logo abaixo da barra superior existe uma barra de menus contendo as opções **Dashboard**, **Workflow**, **Shared** e **Documentation**. A opção **Dashboard** redireciona o usuário para a interface inicial da plataforma. A opção **Workflow** redireciona para a listagem de *workflows* contruídos pelo usuário. A opção **Shared** dá acesso aos *workflows* compartilhados com o usuário. Finalmente, a opção **Documentation** exibe a documentação da plataforma.

Figura 39 – Interface inicial da plataforma SemanticSCo.

Na região central da interface é exibido a quantidade de *workflows* criados pelo usuário (**MY WORKFLOWS**), a quantidade de *workflows* compartilhados pelo usuários (**SHARED BY**

ME), a quantidade de *workflows* compartilhados com o usuário (SHARED WITH ME), a quantidade de mensagens enviadas pelo administrador do sistema para o usuário (MESSAGES FOR YOU) e o espaço utilizado pelos *workflows* do usuário (MEGABYTES USED). Logo abaixo destas informações são apresentadas as atividades recentes do usuário na plataforma (Recent activities) e o conteúdo das mensagens enviadas pelo administrador da plataforma para o usuário (Messages for you).

A.3 Interface com a lista de workflows criados pelo usuário

A Figura 40 apresenta uma interface com os *workflows* criados por um usuário da plataforma. O botão **Create workflow**, localizado na parte superior esquerda da interface, permite a criação de um novo *workflow*. Logo abaixo, encontram-se listados todos os *workflows* criados pelo usuário. Os botões do lado direito de cada *workflow* listado são utilizados para realizar diferentes operações sobre o *workflow*. Os botões disponíveis da esquerda para a direita são: **Edit workflow**, **Rename workflow**, **Copy workflow**, **Change workflow visibility** e **Remove workflow**.

Figura 40 – Interface com a listagem de *workflows* criados pelo usuário da plataforma SemanticSCo.

O botão **Edit workflow** realiza a edição de um *workflow*. A plataforma permite a alteração/execução de um *workflow* por meio desta opção. O botão **Rename workflow** permite alterar o nome de um *workflow*. O botão **Copy workflow** realiza uma cópia integral de todos os elementos de um *workflow*. Esta opção é utilizada para criar um novo *workflow* a partir de um *workflow* existente. O botão **Change workflow visibility** permite criar ou remover o compartilhamento de um *workflow* com os demais usuários da plataforma ou com as redes sociais. A coluna **Visibility** indica o estado de compartilhamento de um *workflow*. O estado **Private** representa um *workflow* cujo acesso é restrito ao seu proprietário. O estado **Shared** representa um *workflow* compartilhado com um ou mais usuários. Por fim, o botão **Remove workflow** remove todos os elementos de um *workflow*.

A.4 Interface para criação de workflows

A Figura 41 apresenta a interface para construção de um *workflow*. A edição de um *workflow* é controlada principalmente por meio de um conjunto de seis botões contidos na barra de ferramentas localizada na parte superior esquerda da interface. O botão **Add Data Source** inclui uma nova fonte de dados. O botão **Add Activity** inclui uma nova atividade de análise. O botão **Service Discovery** descobre os serviços de análise que podem produzir o tipo de dado necessário como entrada para um elemento selecionado. O botão **Remove** apaga um elemento selecionado no *workflow*. Qualquer elemento pode ser apagado, com a exceção do elemento **Start**. As associações entre os elementos (arestas) podem ser removidas clicando sobre elas e na sequência selecionando a opção **Remove**. O botão **Execute** é responsável pela execução de um serviço de análise selecionado. Finalmente, o botão **Close** encerra a edição e visualização do *workflow*.

O centro da interface contém o painel de edição do *workflow*. O elemento **Start** indica o ponto de início de execução do *workflow*, sendo incluído automaticamente em todos *workflows* construídos. Um retângulo com bordas arredondadas representa uma atividade de análise (serviço). Um retângulo com um ícone na parte superior direita representa um serviço de adaptação. Serviços de adaptação são frequentemente utilizados para a manipulação de dados dentro da plataforma. Um cilindro com múltiplos tons de cinza representa uma fonte de dados utilizada como entrada para as atividades de análise e serviços de adaptação. As arestas entre os elementos indicam a sequência de execução

Figura 41 – Interface para criação de *workflows* na plataforma SemanticSCo.

do *workflow*. O ícone azul ao lado de cada elemento indica o ponto de origem utilizado para criar uma aresta entre dois elementos. Para realizar a ligação, o usuário deve clicar sobre o ícone azul e arrastar a aresta até qualquer parte do elemento que deseja utilizar como destino da conexão.

Do lado direito da interface existe um painel de detalhes que permite visualizar e alterar informações dependendo do tipo de elemento selecionado. Por exemplo, o usuário pode alterar o nome de cada um dos elementos contidos no *workflow*, assim como os parâmetros utilizados para a execução dos serviços de análise. O painel de detalhes ainda permite recuperar os arquivos de saída resultantes da execução de um determinado serviço. Este painel pode ser exibido ou ocultado por meio do ícone azul localizado na parte superior direita do painel.

A.5 Interface para acesso aos workflows compartilhados

A Figura 42 apresenta a interface contendo os *workflows* compartilhados com o usuário logado na plataforma. Na parte superior esquerda são apresentadas dois botões: **Shared with me** e **Public sharing**. O botão **Shared with me** apresenta os *workflows* compartilhados diretamente com o usuário atualmente *logado* na plataforma, enquanto que o botão **Public sharing** apresenta os *workflows* compartilhados de forma pública. A coluna **Access type** apresenta o tipo de acesso a um *workflow* compartilhado. Nesta coluna são apresentados os valores **Read only**, que permite o acesso somente leitura e execução para um *workflow*, e **Write** que, além da leitura e execução, permite ao usuário realizar modificações no *workflow*. A coluna **Owner** apresenta o nome do proprietário (criador) do *workflow*.

The screenshot shows the 'Semantic Services Composition' web interface. At the top, there's a header with the logo and navigation links: Dashboard, Workflow, Shared, and Documentation. Below the header, there are two tabs: 'Shared with me' (selected) and 'Public sharing'. The main content area displays a table titled 'Shared workflows' with the following data:

# Id	Name	Description	Access type	Owner	
17	Experiment 1	-	Write	User 1
19	Experiment 2	-	Read only	User 1
26	Study 1	-	Write	User 4
27	Study 2	-	Read only	User 4

At the bottom of the table, there is a pagination control showing '1' of 10 items. The footer contains copyright information: © 2018 - SemanticSCo - Semantic Service Composition, Biomedical Systems and Services Laboratory - University of São Paulo - Brazil, and the USP logo.

Figura 42 – Interface para acesso aos *workflows* compartilhados na plataforma SemanticSCo.

A.6 Interface para edição dos dados do perfil do usuário

A Figura 43 apresenta a interface para edição dos dados do perfil do usuário. A plataforma permite alterar informações básicas como nome do usuário, email, senha de acesso e a foto utilizada. Caso o usuário cadastrado utilize uma ou mais redes sociais para acessar a plataforma, esta rede social aparecerá entre as redes associadas ao seu perfil na área **Associated social networks**. Para remover a associação com uma determinada rede, basta clicar sobre o ícone “x”, ao lado do nome da rede social.

The screenshot shows the 'Personal Information' section of the SemaSCo user profile. At the top, there's a header with the SemaSCo logo and navigation links: Dashboard, Workflow, Shared, and Documentation. The user's name 'Wilson D Silva' and profile picture are displayed. Below the profile picture is a '+ Choose' button and a trash icon. The form fields for 'Name', 'Email', 'Password', and 'Confirm password' are shown, with the current values being 'Wilson D Silva', 'sco@sco.com', and both passwords set to 'Password (not changed)'. The 'Associated social networks' section lists 'Facebook', 'LinkedIn', and 'Google', each with a removal 'x' icon. An 'Update' button is at the bottom of the form. The footer contains copyright information: '© 2018 - SemanticSCo - Semantic Service Composition Biomedical Systems and Services Laboratory - University of São Paulo - Brazil' and the USP logo.

Figura 43 – Interface para edição dos dados do perfil do usuário na plataforma SemanticSCo.

Reprodução do estudo para análise de dados de Microarray Agilent

Este apêndice apresenta o *workflow* construído para a reprodução do experimento envolvendo a análise de dados de Microarray Agilent One-color realizada na versão original da plataforma SemanticSCo (GUARDIA, 2016). Este estudo consiste de uma análise comparativa de expressão gênica de pacientes com esclerose múltipla submetidos a transplante autólogo de células-tronco hematopoiéticas (OLIVEIRA et al., 2015). A primeira parte do estudo consiste de uma análise comparativa entre os pacientes com a doença e pacientes saudáveis (grupo controle), enquanto que a segunda parte do estudo consiste de uma análise comparativa entre pacientes pré e pós transplante.

Com intuito de reproduzir este estudo na plataforma SemanticSCo Web, optou-se por dividi-lo em duas partes, cada qual modelada por meio de um mesmo *workflow*, mas com diferentes conjuntos de dados de entrada. Tal divisão racionaliza a modelagem do estudo e permite o compartilhamento individual dos resultados de análise associados aos diferentes conjuntos de dados. O processo para construção do *workflow* foi dividido em cinco etapas: 1) definição das fontes de dados; 2) normalização; 3) análise diferencial; 4) análise de enriquecimento funcional; e 5) visualização. A Figura 44 apresenta detalhes de cada uma destas etapas.

Na etapa 1, dois conjuntos de dados foram adicionados ao *workflow*. O primeiro conjunto consiste de dados brutos de Microarray Agilent One-color, contendo amostras de pacientes com esclerose múltipla pré transplante e amostras de pacientes saudáveis (grupo controle). Estes dados foram obtidos via integração da plataforma SemanticSCo Web com o repositório ArrayExpress (identificador de acesso E-MEXP-3905), sendo ano-

Figura 44 – Etapas para a criação do *workflow* para a análise de dados de Microarray Agilent One-color.

tados semanticamente como *Agilent One-color Raw Microarray Data*. Os dados de expressão gênica de ambos os conjuntos de amostras utilizam os identificadores das sondas de DNA da plataforma Agilent (identificador *Probe ID*).

O segundo conjunto de dados adicionado ao *workflow* contém um mapeamento entre três tipos de identificadores de genes: *Probe Id*, *Entrez Id* e *Gene Symbol*. Estes dados foram adicionados ao *workflow* a partir de um arquivo local, sendo anotados semanticamente como *Gene Identifiers Mapping*.

Na etapa 2, os dados de expressão gênica dos pacientes são associados ao serviço *Agilent One-color Microarray Data Preprocessing*, responsável por realizar a normalização de dados de Microarray Agilent One-color.

Na etapa 3, os dados normalizados são associados ao serviço *Splitter*, responsável por separar os dados normalizados em dois conjuntos distintos, os quais representam os dados dos pacientes pré transplante (conjunto 1) e dados do grupo controle (conjunto 2). Estes dois conjuntos de dados normalizados foram então associados ao serviço *Two-sample Student's T-test Analysis*, responsável por realizar a análise diferencial dos dados.

Na etapa 4, os dados resultantes da análise diferencial são associados ao serviço *Up Regulated Gene Extraction*, responsável por selecionar, dentre o conjunto de dados diferencialmente expressos, os genes regulados positivamente. Na sequência, os dados dos genes regulados positivamente são associados ao serviço *Gene Identification Mapping*,

responsável por realizar a troca dos identificadores dos genes (Probe ID para Entrez ID). Esta troca é necessária porque o serviço de enriquecimento funcional disponível na plataforma, baseado na ferramenta DAVID (HUANG; SHERMAN; LEMPICKI, 2009), não aceita o formato de identificador utilizado pelos dados de expressão. Após a substituição dos identificadores dos genes, os dados resultantes são então associados ao serviço **Single Enrichment Analysis Of Gene Expression Data**, responsável por realizar a análise de enriquecimento funcional.

Na etapa 5, os dados normalizados são primeiramente associados a uma nova instância do serviço **Gene Identifiers Mapping**, de modo a realizar a troca dos identificadores dos genes (Probe ID para Entrez ID). A seguir, este conjunto de dados e o conjunto de dados produzido pela análise de enriquecimento funcional são então associados ao serviço **KEGG Pathway Data Filtering**, responsável por filtrar os dados das vias biológicas KEGG. Finalmente, os dados resultantes da filtragem são então associados ao serviço **Gene Expression Data Rendering Into KEGG Pathway Graph**, responsável por gerar uma imagem (grafo) de uma determinada via biológica KEGG de interesse.

Figura 45 – *Workflow* para análise de dados de Microarray Agilent One-color.

A Figura 45 apresenta o *workflow* construído na plataforma SemanticSCo Web para a reprodução do estudo de análise de dados de Microarray Agilent One-color. Este *workflow* foi então executado utilizando os parâmetros indicados em (GUARDIA, 2016) e os resultados desta execução foram comparados aos resultados obtidos na versão original da plataforma. Esta análise comparativa constatou que os resultados das duas execuções

são similares.

A Tabela 9 descreve os parâmetros utilizados na execução de cada serviço. Os serviços Agilent One-color Microarray Data Preprocessing e KEGG Pathway Data Filtering não foram incluídos nesta tabela pois não são parametrizados. Adicionalmente, foi incluído apenas uma entrada para o serviço Gene Identification Mapping pois as duas instâncias deste serviço utilizam os mesmos valores de parâmetros.

Tabela 9 – Parâmetros utilizados para a execução do *workflow* para a análise de dados de Microarray Agilent.

Serviço	Parâmetros de execução
Splitter	<ul style="list-style-type: none"> • Label column = ID • Experimental conditions 1 = control1;control2;control3;control4 • Experimental conditions 2 = pre1;pre2;pre3;pre4
Two-sample Student's T-test Analysis	<ul style="list-style-type: none"> • Correction method = none • Cut off value = 2 • Significance level = 0.05
Up Gegulated Gene Extraction	<ul style="list-style-type: none"> • Fold change value = 2 • p-value = 0.05
Gene Identification Mapping	<ul style="list-style-type: none"> • Source map column = 0 (Probe Id) • Target map column = 1 (Entrez Id)
Single Enrichment Analysis Of Gene Expression Data	<ul style="list-style-type: none"> • Gene identifier type = ENTREZ_GENE_ID • Enrichment categories = GOTERM_BP_FAT,GOTERM_CC_FAT,GOTERM_MF_FAT,KEGG_PATHWAY • DAVID-registered e-mail = wilson@usp.br • Threshold = 0.1 • Count = 2
Gene Expression Data Rendering Into KEGG Path Graph	<ul style="list-style-type: none"> • Gene identifier type = ENTREZ_GENE_IDENTIFIER • File = hsa05033 • Pathway identifier = 05033

Reprodução do estudo para análise de dados de Microarray Affymetrix

Este apêndice apresenta o *workflow* construído para a reprodução do experimento envolvendo a análise de dados de Microarray Affymetrix One-color realizada na versão original da plataforma SemanticSCo (GUARDIA, 2016). Este estudo consiste de um conjunto de dados obtidos de amostras cerebrais de peixes-zebra divididos em quatro grupos. O grupo *Bystanders to Interacting Conspecifics* (BIC) possui indivíduos que observaram de forma atenta uma situação de conflito entre outros indivíduos. O grupo *Bystanders Attentive to Non-Interacting Conspecifics* (BANIC) possui indivíduos que observaram de forma atenta uma situação de não-conflito entre outros indivíduos. O grupo *Bystanders Inattentive to Non-Interacting Conspecifics* (BINIC) possui indivíduos que observaram de forma inatenta uma situação de não-conflito entre outros indivíduos. E por fim, o grupo *Isolated Fish* (ISOL) possui indivíduos isolados socialmente (grupo controle). O objetivo do estudo foi investigar alterações no perfil de expressão gênica de cada grupo (LOPES; ABREU; OLIVEIRA, 2015).

Com intuito de reproduzir este estudo na plataforma SemanticSCo Web optou-se por dividi-lo em duas partes, cada qual modelada em um *workflow* específico devido à quantidade de atividades de análise realizadas no estudo original. Tal divisão facilitou a modelagem do estudo.

C.1 Workflow 1

O processo para construção do *workflow* 1 foi dividido em quatro etapas: 1) definição das fontes de dados; 2) análise diferencial; 3) filtragem e normalização; e 4) visualização. A Figura 46 apresenta detalhes de cada uma destas etapas.

Figura 46 – Etapas para a criação do *workflow* 1 para a análise de dados de Microarray Affymetrix One-color.

Na etapa 1, dois conjuntos de dados foram adicionados ao *workflow*. O primeiro conjunto consiste de dados normalizados de Microarray Affymetrix One-color, contendo amostras cerebrais de peixes-zebra divididos nos grupos BIC, BANIC BINIC e ISOL. Estes dados foram adicionados a partir de um arquivo local, sendo anotados semanticamente como **Log-transformed One-color Normalized Microarray Data**. Os dados de expressão gênica do conjunto de amostras utilizam os identificadores das sondas de DNA da plataforma Affymetrix (identificador **Probe ID**).

O segundo conjunto de dados adicionado ao *workflow* contém um mapeamento entre três tipos de identificadores de genes: **Probe Id**, **Entrez Id** e **Gene Symbol**. Estes dados foram adicionados ao *workflow* a partir de um arquivo local, sendo anotados semanticamente como **Gene Identifiers Mapping**.

Na etapa 2, os dados de expressão das amostras cerebrais de peixes-zebra são associados a três instâncias diferentes do serviço **Splitter**, responsável por separar os dados normalizados em dois conjuntos distintos, os quais representam as amostras cerebrais de peixes-zebra de diferentes grupos. Para a primeira instância (**Splitter**) foram

separadas as amostras referentes aos grupos ISOL (conjunto 1) e BIC (conjunto 2). Para a segunda instância (**Splitter(2)**) foram separadas as amostras referentes aos grupos ISOL (conjunto 1) e BANIC (conjunto 2). Por fim, para a terceira instância (**Splitter(3)**) foram separadas as amostras referentes aos grupos ISOL (conjunto 1) e BINIC (conjunto 2). Na sequência, cada um dos conjuntos de dados obtidos a partir das diferentes instâncias do serviço **Splitter** são associados a diferentes instâncias do serviço **Two-sample Student's T-test Analysis**, responsável por realizar a análise diferencial dos dados.

Na etapa 3, os dados resultantes da análise diferencial e os dados normalizados contendo as amostras cerebrais de peixes-zebra (incluídos na etapa 1) são associados ao serviço **Gene Filtering**, responsável por selecionar os genes e valores normalizados associados com base nos conjuntos de dados de entrada. Na sequência, os dados dos genes e valores selecionados são associados ao serviço **Standard Deviation Normalization**, responsável por normalizar um conjunto de dados de acordo com o desvio padrão.

Na etapa 4, os dados normalizados são associados ao serviço **Gene Identification Mapping**, de modo a realizar a troca dos identificadores dos genes (**Probe ID** para **Gene Symbol**). A seguir, este conjunto de dados é então associado ao serviço **Hierarchical Clustering Of Microarray Data**, responsável por realizar o agrupamento hierárquico de dados de Microarray. Finalmente, os dados resultantes do agrupamento são então associados ao serviço **Dendrogram Generation**, responsável por gerar uma imagem contendo um dendrograma a partir dos dados do agrupamento hierárquico associados a entrada do serviço.

A Figura 47 apresenta o *workflow* construído na plataforma SemanticSCo Web para a reprodução do estudo de análise de dados de Microarray Affymetrix One-color. Este *workflow* foi então executado utilizando os parâmetros indicados em (GUARDIA, 2016) e os resultados desta execução foram comparados aos resultados obtidos na versão original da plataforma. Esta análise comparativa constatou que os resultados das duas execuções são similares, exceto pelos resultados obtidos na execução dos serviços de análise diferencial. Segundo GUARDIA (2016), o valor 1.1 deveria ser utilizado como *cutoff* para o serviço de análise diferencial. Contudo, o uso deste valor resultou em uma saída diferente da esperada. Ao modificar o valor de *cutoff* para 1.5, o resultado da execução das instâncias dos serviços **Two-sample Student's T-test Analysis** foi idêntico ao resultado descrito no estudo original.

A Tabela 10 descreve os parâmetros utilizados na execução de cada serviço. Os serviços **Gene Filtering**, **Standard Deviation Normalization** e **Dendrogram Generation** não foram incluídos nesta tabela pois não são parametrizados. Adicionalmente, apenas uma entrada para o serviço **Two-sample Student's T-test Analysis** foi incluída pois as três instâncias deste serviço utilizam os mesmos valores de parâmetros.

Figura 47 – *Workflow* 1 para análise de dados de Microarray Affymetrix One-color.

C.2 Workflow 2

O processo para construção do *workflow* 2 foi dividido em duas etapas: 1) definição das fontes de dados; e 2) enriquecimento. A Figura 48 apresenta detalhes de cada uma destas etapas.

Na etapa 1, três conjuntos de dados foram adicionados ao *workflow*. O primeiro conjunto consiste de dados normalizados de Microarray Affymetrix One-color, contendo amostras cerebrais de peixes-zebra divididos nos grupos BIC, BANIC BINIC e ISOL. Estes dados foram adicionados a partir de um arquivo local, sendo anotados semanticamente como **Log-transformed One-color Normalized Microarray Data**. Os dados de expressão gênica do conjunto de amostras utilizam os identificadores das sondas de DNA da plataforma Affymetrix (identificador **Probe ID**).

O segundo conjunto consiste de dados sobre vias biológicas. Estes dados foram adicionados ao *workflow* a partir de um arquivo local, sendo anotados semanticamente

Tabela 10 – Parâmetros utilizados para a execução do *workflow* 1 para análise de dados de Microarray Affymetrix.

Serviço	Parâmetros de execução
Splitter	<ul style="list-style-type: none"> • Label column = ID_REF • Experimental conditions 1 = ISOL_1;ISOL_2;ISOL_3 • Experimental conditions 2 = BIC_1;BIC_2;BIC_3
Splitter(2)	<ul style="list-style-type: none"> • Label column = ID_REF • Experimental conditions 1 = ISOL_1;ISOL_2;ISOL_3 • Experimental conditions 2 = BANIC_1;BANIC_2;BANIC_3
Splitter(3)	<ul style="list-style-type: none"> • Label column = ID_REF • Experimental conditions 1 = ISOL_1;ISOL_2;ISOL_3 • Experimental conditions 2 = BINIC_1;BINIC_2;BINIC_3
Two-sample Student's T-test Analysis	<ul style="list-style-type: none"> • Correction method = none • Cut off value = 1.5 • Significance level = 0.05
Gene Identification Mapping	<ul style="list-style-type: none"> • Source map column = 0 (Probe Id) • Target map column = 2 (Gene Symbol)
Hierarchical Clustering Of Microarray Data	<ul style="list-style-type: none"> • Distance measure = city-block distance • Clustering method = pairwise average-linkage

como Gene Set.

O terceiro conjunto de dados adicionado ao *workflow* contém um mapeamento entre três tipos de identificadores de genes: Probe Id, Entrez Id e Gene Symbol. Estes dados foram adicionados ao *workflow* a partir de um arquivo local, sendo anotados semanticamente como Gene Identifiers Mapping.

Figura 48 – Etapas para a criação do *workflow* 2 para a análise de dados de Microarray Affymetrix One-color.

Na etapa 2, os dados de expressão gênica contendo as amostras cerebrais de peixes-zebra são associados ao serviço **Gene Identification Mapping**, responsável por realizar a troca dos identificadores dos genes (Probe ID para Gene Symbol). Esta troca é necessária porque o serviço de enriquecimento de dados disponível na plataforma não aceita o formato de identificador utilizado pelos dados de expressão. Após a substituição dos identificadores dos genes, os dados resultantes, em conjunto com os dados sobre vias biológicas, são então associados ao serviço **Gene Set Enrichment Analysis Of Gene Expression Data**, responsável por realizar a análise de enriquecimento de dados de expressão gênica.

A Figura 49 apresenta o *workflow* construído na plataforma SemanticSCo Web para a reprodução do estudo de análise de dados de Microarray Affymetrix One-color. Este *workflow* foi então executado utilizando os parâmetros indicados em (GUARDIA, 2016) e os resultados desta execução foram comparados aos resultados obtidos na versão original da plataforma. Esta análise comparativa constatou que os resultados das duas execuções são idênticos. A Tabela 11 descreve os parâmetros utilizados na execução de cada serviço.

Figura 49 – Workflow 2 para análise de dados de Microarray Affymetrix One-color.

Tabela 11 – Parâmetros utilizados para a execução do workflow 2 para análise de dados de Microarray Affymetrix.

Serviço	Parâmetros de execução
Gene Identification Mapping	<ul style="list-style-type: none"> • Source map column = 0 (Probe Id) • Target map column = 2 (Gene Symbol)
Gene Set Enrichment Analysis Of Gene Expression Data	<ul style="list-style-type: none"> • FDR adjustment = TRUE • Experimental design = unpaired • Reference experimental conditions = 1;2;3 • Target experimental conditions = 10;11;12

Reprodução do estudo para análise de dados de RNA-Seq Illumina

Este apêndice apresenta o *workflow* construído para a reprodução do experimento envolvendo a análise de dados de RNA-Seq Illumina realizada na versão original da plataforma SemanticSCo (GUARDIA, 2016). Este estudo consiste de uma análise comparativa entre dois conjuntos distintos envolvendo dados de expressão gênica de amostras de melanócitos tumorais da linhagem 501Mel e melanócitos normais imortalizados da linhagem Hermes 3A obtidas a partir de condições experimentais distintas: supressão si/shRNA do gene BRG1, supressão si/shRNA do gene MITF e não-supressão dos genes BRG1/MITF (controle) (LAURETTE et al., 2015). A primeira parte do estudo foi consiste de uma análise comparativa do perfil de expressão gênica dos melanócitos tumorais. A segunda parte do estudo foi realizado uma análise comparativa dos perfis de expressão gênica dos melanócitos normais.

Com intuito de reproduzir este estudo na plataforma SemanticSCo Web, optou-se por dividi-lo em duas partes, cada qual modelada com um mesmo *workflow*, mas com diferentes conjuntos de dados de entrada. Tal divisão racionaliza a modelagem do estudo e permite o compartilhamento individual dos resultados de análise associados aos diferentes conjuntos de dados. O processo para construção do *workflow* foi dividido em três etapas: 1) definição das fontes de dados; 2) análise diferencial; e 3) análise de enriquecimento funcional. A Figura 50 apresenta detalhes de cada uma destas etapas.

Na etapa 1, dois conjuntos de dados foram adicionados ao *workflow*. O primeiro conjunto consiste de dados pré processados de RNA-Seq Illumina, contendo amostras de supressão si/shRNA do gene BRG1 e não-supressão dos genes BRG1 (controle). O segundo

Figura 50 – Etapas para criação do *workflow* para a análise de dados de RNA-Seq.

conjunto consiste de dados pré processados de RNA-Seq Illumina, contendo amostras de supressão si/shRNA do gene MITF e não-supressão dos genes MITF (controle). Estes dados foram obtidos via integração da plataforma SemanticSCo Web com o repositório ArrayExpress (identificador de acesso GSE61966¹), sendo anotados semanticamente como RNA-Seq Count Data Table. Os dados de expressão gênica de ambos os conjuntos de amostras utilizam os identificadores dos genes no formato Ensemb Gene Id.

Na etapa 2, os dados das amostras de supressão si/shRNA do gene BRG1 e não-supressão do gene BRG1 (controle) são associados ao serviço Differential Analysis Of RNA-Seq Data, responsável por realizar a análise diferencial dos dados. Paralelamente, os dados das amostras de supressão si/shRNA do gene MITF e não-supressão do gene MITF (controle) são associados a uma segunda instância do mesmo serviço para análise diferencial. Na sequência, os dados resultantes da análise diferencial de ambas as instâncias do serviço Differential Analysis Of RNA-Seq Data são associados ao serviço Cross Matching Up Regulated Identification, responsável por selecionar os genes regulados positivamente comuns a ambos os conjuntos de dados de entrada.

Na etapa 3, os dados dos genes regulados positivamente são associados ao serviço Single Enrichment Analysis Of Gene Expression Data. Este serviço é responsável por realizar a análise de enriquecimento funcional do conjunto de dados de entrada.

A Figura 51 apresenta o *workflow* construído na plataforma SemanticSCo Web para a reprodução do estudo de análise de dados de RNA-Seq Illumina. Este *workflow* foi então executado utilizando os parâmetros indicados em (GUARDIA, 2016) e os re-

¹ O identificador de acesso GSE61966 do repositório GEO foi utilizado para a pesquisa pelos dados do experimento. A plataforma SemanticSCo Web converte automaticamente este identificador para o formato correspondente do ArrayExpress (E-GEOD-61966) antes de realizar a pesquisa pelos dados do experimento.

Figura 51 – *Workflow* para análise de dados de RNA-Seq.

sultados desta execução foram comparados aos resultados obtidos na versão original da plataforma. Esta análise comparativa constatou que os resultados das duas execuções são similares, exceto pelos resultados obtidos na execução do serviço **Single Enrichment Analysis Of Gene Expression Data**, que foram ligeiramente diferentes aos observados em (GUARDIA, 2016), o que pode estar relacionado a eventuais alterações na ferramenta DAVID (HUANG; SHERMAN; LEMPICKI, 2009), ferramenta esta encapsulada pelo serviço **Single Enrichment Analysis Of Gene Expression Data**.

A Tabela 12 descreve os parâmetros utilizados na execução de cada serviço. O serviço **Cross Matching Up Regulated Identification** não foi incluído nesta tabela pois não é parametrizado.

Tabela 12 – Parâmetros utilizados para a execução do *workflow* para análise de dados de RNA-Seq.

Serviço	Parâmetros de execução
Differential Analysis Of RNA-Seq Data	<ul style="list-style-type: none"> • Reference samples = 501_shSCR1_rep1.txt;501_shSCR1_rep2.txt;501_shSCR1_rep3.txt • Perform FDR adjustment of result = true • Significance level = 0.05 • Target samples = 501_shBRG1_rep1.txt;501_shBRG1_rep2.txt; 501_shBRG1_rep3.txt • Cut off value = 1.0
Differential Analysis Of RNA-Seq Data(1)	<ul style="list-style-type: none"> • Reference samples = 501_shSCR2_rep1.txt;501_shSCR2_rep2.txt • Perform FDR adjustment of result = true • Significance level = 0.05 • Target samples = 501_shMITF_rep1.txt;501_shMITF_rep2.txt • Cut off value = 1.0
Single Enrichment Analysis Of Gene Expression Data	<ul style="list-style-type: none"> • Gene identifier type = ENSEMBL_GENE_ID • Enrichment categories = GOTERM_BP_FAT • DAVID-registered e-mail = wilson@usp.br • Count = 2 • Threshold = 0.1