UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTMENTO DE INFORMÁTICA

SESAME: CLUSTERING WITH SEMANTIC SIMILARITY BASED ON MULTIPLE ONTOLOGIES

Carlos Alexandre Lourenço dos Santos

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

Dissertação Orientada por:
Professora Doutora Cátia Luísa Santana Calisto Pesquita

2016

Acknowledgments

I am much obliged to professor Cátia Pesquita for her assertive supervising, broad knowledge and cooperative spirit. These characteristics provided valuable confidence in achieving a successful and appreciated work contributing to the feeling that the effort was worthwhile. I would also like to thank my colleague João Rebelo who selflessly helped to collect useful data for results' analysis. Finally, this work benefited of the help obtained, in crucial moments, by email from Sébastien Harispe – author of the Semantic Measures Library – Eibe Frank – author of the Waikato Environment for Knowledge Analysis – and the Universal Protein Resource's helpdesk.

Dedicated to my family!

Resumo

Muitas das técnicas de prospecção de dados actualmente utilizadas funcionam de um modo “cego”, limitando-se ao que pode ser extraído directamente a partir dos dados, sem compreender o seu significado e, de um modo geral, deixando a interpretação dos resultados para peritos humanos. É, no entanto, amplamente reconhecido que codificar um maior número de relações entre objectos melhora o desempenho de abordagens de prospecção de dados. Isto, tipicamente, envolve a feitura de regras que sejam capazes de expressar conhecimento relativamente à forma como objectos de dados se relacionam entre si, mas o surgimento de tecnologias de *Semantic Web* e a sua aplicação em domínios diversificados como as ciências da vida, a astronomia ou a geografia, está a disponibilizar uma grande quantidade de dados enriquecidos com conhecimento de domínio na forma de múltiplas ontologias. Este cenário apresenta oportunidades únicas para a combinação do poder de abordagens de prospecção de dados e aprendizagem máquina com o conhecimento codificado em ontologias. O presente trabalho tem por objectivo abordar o desafio apresentado por esta mudança de paradigma através do desenvolvimento de novas abordagens para a descoberta de conhecimento alavancadas em tecnologias de *Semantic Web* e na abundância de conhecimento tornado disponível por intermédio das mesmas. Neste contexto, a semelhança semântica surge como um possível caminho para fazer a ponte entre os dois mundos, uma vez que pode ser usada para produzir uma medida de distância entre dois conceitos de uma ontologia ou entre duas entidades anotadas com conjuntos de conceitos de uma ontologia. Tendo em consideração que a distância é uma pedra angular de um número considerável de abordagens de aprendizagem máquina, incluindo diversas abordagens de segmentação (como, por exemplo, *k-Means* e *Farthest First*), a integração de semelhança semântica em algoritmos representativos do estado da arte da aprendizagem máquina disponibiliza uma forma de explorar dados usando o conhecimento contido em ontologias.

Tendo em vista atingir os objectivos descritos, foi implementada uma estrutura que utiliza duas bibliotecas de software do mais alto nível de desenvolvimento: a Biblioteca de Medidas Semânticas (SML) para o cálculo de semelhança semântica e o Ambiente Waikato para Análise de Conhecimento (WEKA) para algoritmos de aprendizagem máquina. A SML foi ainda estendida tendo em vista permitir a computação de semelhança semântica usando múltiplas ontologias. Pela disponibilização de informação acrescida relativamente a relações entre entidades, o recurso a referências semânticas provenientes de mais do que uma ontologia representa uma oportunidade para reforçar a qualidade potencial de processos de segmentação. Lidar com a integração de múltiplas ontologias numa única medida de semelhança semântica é um desafio conhecido. Neste trabalho foram usadas duas abordagens simples: Híper-grafo e Média Ponderada. Para se obter um híper-grafo na SML, é necessário levar a efeito um processo de redefinição de raízes em que uma raiz virtual é criada para ligar os grafos carregados com cada uma das ontologias envolvidas. A abordagem de média ponderada combina os valores de semelhança semântica pela ponderação dos contributos de cada ontologia. No que diz respeito ao interface com o utilizador, para além de uma opção simples baseada em texto e da possibilidade de execução com especificação de parâmetros em linha de comando foi feita a integração das novas opções no explorador gráfico do WEKA e desenvolvido um ambiente gráfico próprio. Os resultados de cada execução são disponibilizados num ficheiro cujo conteúdo visa essencialmente disponibilizar toda a informação relativa a essa execução com o máximo de clareza incluindo, nomeadamente, uma designada matriz de confusão identificando o número de instâncias de cada classe de dados afetado a cada segmento.

O conjunto de dados usado na avaliação da aplicação de segmentação desenvolvida foi obtido a partir de caminhos metabólicos presentes no repositório *Reactome* que disponibiliza uma lista de proteínas envolvidas para cada um dos caminhos metabólicos. A avaliação foi focada em três tipos de conjuntos de caminhos metabólicos humanos com anotações na ontologia de genes (GO) e/ou na ontologia de entidades químicas de interesse biológico (ChEBI): (1) Sem Ligação, ou seja, grupos distantes de caminhos metabólicos, sem qualquer ligação entre si; (2) Com Ligação, ou seja, diferentes grupos de caminhos metabólicos com uma ligação entre si e (3) Mesmo Grupo, ou seja, caminhos metabólicos pertences a um mesmo grupo de caminhos. Para cada conjunto foram efectuados oito testes, cada um deles com dezasseis tarefas de segmentação, com tamanhos de dados e números de segmentos alvo diversificados. A aplicação inclui dois algoritmos de segmentação, *SimplekMeans* e *Farthest First*, e foi testada com duas bem conhecidas medidas de semelhança semântica, a medida semântica de comparação directa de grupos de anotações por cada duas entidades *SimGIC* e a medida semântica de comparação indirecta de grupos de anotações por cada duas entidades baseada na medida de comparação de pares de conceitos *Lin* com uma estratégia de agregação Média de Melhores Correspondências. Uma linha de base – referência para os resultados de segmentação tendo em vista capturar a influência da utilização de distâncias semânticas em contraponto às distâncias convencionalmente usadas em segmentação – foi estabelecida cujas anotações foram tratadas como palavras usando filtro disponibilizado pelo WEKA que converte um atributo de cadeia de caracteres num vector representativo das frequências de ocorrência de palavras. Tendo em conta o tipo (dos três atrás descritos) de conjunto de caminhos metabólico, o uso de semelhança semântica é claramente benéfico tanto para o tipo Sem

Ligação como para o tipo Com Ligação, com aumentos de desempenho que vão desde $+3\%$ a $+11\%$. No que diz respeito ao conjunto Mesmo Grupo, a linha de base tem um desempenho em média melhor do que as abordagens baseadas em semelhança semântica. Os resultados usando ambas as ontologias ou apenas a GO revelam desempenhos muito semelhantes para as mesmas abordagens de segmentação e semelhança semântica, o que não acontece quando é usada unicamente a ontologia ChEBI. Uma pequena parte das proteínas usadas nos conjuntos de dados são anotadas com conceitos da ontologia ChEBI (apenas cerca de 5 a 10%) e a estrutura daquela ontologia é maioritariamente plana, com uma grande proporção de nodos folhas, o que se confirmou diminuir o impacto da utilização de medidas de semelhança semântica. Foi possível confirmar a conhecida tendência em algoritmos de segmentação baseados no *k-Means* para uma diminuição do desempenho da segmentação associada ao aumento do número alvo de segmentos e ainda, verificar que essa tendência se agrava consideravelmente se, com um elevado número de segmentos alvo, se conjugar um muito elevado número de instâncias a segmentar. Mostrou-se também que esta conjugação se revela, como seria de esperar, causadora de piores tempos de execução com a curiosidade de tal se verificar quando é usado o *SimplekMeans* mas não com o *Farthest First*. O primeiro foi, nas mesmas condições de teste, sempre mais lento que o segundo assim como a medida *SimGIC* foi sempre mais rápida do que a baseada na medida *Lin*.

Foi então possível demonstrar que a utilidade de empregar semelhança semântica depende não só da diversidade e qualidade das anotações existentes nos conjuntos de dados, mas também da estrutura das ontologia usadas e do grau em que as mesmas são capazes de acrescentar informação útil para identificar instâncias semelhantes. O presente trabalho constitui-se como um primeiro contributo que abre caminho a esforços futuros complementares em frentes diversas como, por exemplo: (1) Avaliar melhor as suas potencialidades com testes adicionais com diferentes combinações e números de ontologias usadas bem como diferentes fontes de dados; (2) Explorar algoritmos de segmentação, incluindo métodos de inicialização de centróides, alternativos; (3) Considerar medidas de semelhança semântica mais complexas e (4) Investigar aspectos relacionados com a eficiência computacional no uso de múltiplas ontologias. Em última análise, a abordagem proposta pode vir a ser usada para analisar conjuntos de dados diversos compostos tanto por anotações semânticas como por valores numéricos, através da sua combinação com as abordagens convencionais já disponíveis.

Palavras-chave: ontologia, semelhança semântica, segmentação, prospecção de dados.

Abstract

Many of the currently employed data mining techniques work in a blind mode, limiting themselves to what can be extracted directly from the data, without understanding its meaning. It is, however, widely recognized that encoding more relations between objects increases the performance of data mining approaches. This typically involves the handcrafting of rules that are able to express knowledge about how data objects relate to each other, but the emergence of semantic web technologies and their application in diverse domains is providing a wealth of data that is enriched with domain knowledge in the form of multiple ontologies.

The present work aims at addressing the challenge presented by this paradigm shift by integrating semantic similarity into machine learning algorithms to explore data using the knowledge contained in ontologies. A software application was developed that utilizes two state of the art libraries: The Semantic Measures Library (SML) for semantic similarity calculations and The Waikato Environment for Knowledge Analysis (WEKA) for machine learning algorithms. SML was further extended to allow the computation of semantic similarity using multiple ontologies.

The data-set used in the application's evaluation was derived from the metabolic pathways present in Reactome, which provides a list of involved proteins for each of the pathways. The evaluation focused on three types of sets of human pathways with annotations to GO and ChEBI: (1) No Link, not linked pathways' groups; (2) Link, pathways' groups with one link and (3) Same Group, pathways in the same group. It was shown that the usefulness of employing semantic similarity depends not only on the diversity and quality of the data-sets annotations, but also on the structure of the ontologies employed, and the degree to which they are able to impart useful information to identify similar instances. Ultimately, the proposed approach can be used to analyze diverse data-sets composed of both semantic annotations and numerical values, by combining it with the conventional approaches already available.

Keywords: ontology, semantic similarity, clustering, knowledge discovery from data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

During the past three decades the techniques to process and analyze data have been a subject of intense research and development. Their improvement during this period has been amazing and motivated by several factors among which two must be underlined:

- The increasing acknowledgment of data harnessing importance in all sectors, from leisure to economics passing through management, sciences, defense or even politics;
- The necessity to effectively and efficiently analyze huge volumes of data accumulated in emerging internet-based global information such as the world wide web and various kinds of interconnected, heterogeneous databases.

The present work is part of the mentioned research efforts in this case driven by the intention to contribute to the improvement of data mining techniques particularly those concerning the semantic web. The project specifically aims to explore concepts and entities' semantic similarity calculation as support to the data mining technique of clustering instances according to their degree of similarity.

1.1 Motivation

Although phenomenally successful in terms of size and number of users, a world wide web which content consists mainly of distributed hypertext and hypermedia accessed via a combination of keyword based search and link navigation is fundamentally a relatively simple artifact. This simplicity has been one of its great strengths favoring popularity and growth since even naive users are able to use it and can even create their own content. However, the explosion in both the range and quantity of web content has highlighted some serious shortcomings in the hypertext paradigm: the required content becomes increasingly difficult to locate using search and browse and answering more complex queries – along with more general information retrieval, integration, sharing and processing – can be difficult or even impossible [1]. Nowadays, a paradigm-shift is being witnessed. With the ultimate goal of allowing data to be shared effectively by wider communities and to be processed automatically by tools as well as manually, semantic web and its technologies are empowering the access to background knowledge – once scarce and difficult to explore – in the form of ontologies and more and more data are being released as linked data.

The computer science field of data mining – the process of discovering interesting patterns and knowledge from large amounts of data – is inseparable from these developments, particularly web content, structure and usages mining [2]. Similar relevance is assumed by machine learning, a branch of artificial intelligence in which, using computing, systems are designed that can learn from data in a manner of being trained. These systems may learn and improve with experience, and with time, refine a model that can be used to predict outcomes of questions based on the previous learning [3]. A recent leap in data mining and machine learning has been the emergence of deep learning [4]. These techniques have greatly improved performance on a number of unsupervised learning tasks, however, as many of the previous techniques they work in a blind mode, limiting themselves to what can be extracted directly from the data, without understanding its meaning, mostly leaving interpretation of the results to human experts [5].

It is widely recognized that encoding more relations between objects increases the performance of data mining approaches [6]. This typically involves the handcrafting of rules that are able to express knowledge about how data objects relate to each other, but the emergence of semantic web technologies and their application in diverse domains such as life sciences, astronomy or geography, is providing multiple ontologies with data that is enriched with domain knowledge. This panorama presents unique opportunities in combining the power of data mining and machine learning approaches with the knowledge encoded in ontologies. In particular, data-sets annotated with multiple ontologies are becoming increasingly common, for instance, proteins whose function is described using the Gene Ontology (GO) [7], ligands using the ontology for Chemical Entities of Biological Interest (ChEBI) [8] and phenotypic effects using the Human Phenotype Ontology [9] or electronic health records which use different terminologies and ontologies to describe diagnostics, symptoms and medical procedures.

The present work aims at addressing the challenge presented by the aforementioned paradigm shift by developing novel approaches for knowledge discovery that leverage on semantic web technologies and the abundance of knowledge made available through them. Semantics enriched data can be explored by the design and development of data mining algorithms that are able to make use of the background knowledge expressed in data ontology annotations in tandem with the information imparted by the data values. In this context semantic similarity emerges as a possible avenue to bridge the two worlds, since it can be used to produce a measure of distance between two ontology concepts or two entities annotated with ontology concept sets. Considering that distance is a cornerstone of a number of machine learning approaches, including several clustering approaches (e.g., k-means, farthest-first, etc.) where only semantics unaware distance measures like Euclidean, Manhattan or Chebyshev keep on being used, the integration of semantic similarity into state of the art machine learning algorithms provides a way to explore data using the knowledge contained in ontologies. Despite its unquestionable usefulness, robust measurement of semantic similarity aiming for automatically assessing a numerical score between a pair of terms according to the semantic evidence observed in knowledge sources (used as semantic background) remains a challenging and motivating task [10]. Particularly in the case of using multiple ontologies as semantics' sources because these approaches provide complementary views of reality so that the incompleteness, errors and subjective interpretations, usual in single ontology approaches, are mitigated [11]. Numerous communities like, for instance, bioinformatics, natural language processing or artificial intelligence are involved in the study of semantic similarity measures.

This work's final purpose is to, more than harnessing the power of the semantic web in a data mining context, to do so in a manner that can be easily used by other researchers or analysts to extract valuable knowledge from their data. A popular machine learning suite was integrated with a state of the art library for semantic similarity, which has been extended to handle multiple ontologies. The integrated system has been subject to a primary evaluation using a data-set of proteins annotated with multiple ontologies.

1.2 Goals

This research project's global goals are:

1. Develop new clustering strategies based on the exploration of the semantic space making use of semantic similarity measures;
2. Implement those strategies in integration with a data mining library;
3. Assessment of those strategies using real data.

1.3 Contributions

1. Integration of semantic similarity measures into the popular data mining framework WEKA;
2. Extension of state of the art semantic measures library to handle multiple ontologies;
3. Providing preliminary test's results of the developed software solution for clustering with semantic similarity based on multiple ontologies;

4. Release of the implemented software on github.

1.4 Document Structure

This document is organized in the following way:

- Chapter 1, Introduction, where the motivation, goals and contributions are outlined;
- Chapter 2, Related Work and State of the Art, introduces some of the most relevant documented work concerning the context of continuous scientific research and development in the areas of data mining, machine learning and semantic similarity;
- Chapter 3, Clustering with Semantic Based Distances, describes the requirements, design and implementation strategies undertaken to achieve the integration of semantic similarity distance into clustering algorithms.
- Chapter 4, Semantic Similarity with Multiple Ontologies, characterizes the possible scenarios to compute semantic similarity using multiple ontologies, and the specificities of their exploration in the developed software solution;
- Chapter 5, Evaluation, provides a detailed description of the procedures adopted to test the developed application and discussion of those tests main results;
- Chapter 6, Conclusion, presents the major conclusions and possible directions for future work.

Chapter 2

Related Work and State of the Art

This work involved three important information technology research fields associated to the process of Knowledge Discovery from Data (KDD), ie, the automated or convenient extraction of patterns representing knowledge implicitly stored or captured in large databases, data warehouses, the web, other massive information repositories or data streams [12]: data mining, machine learning and semantic similarity. All of these areas are continuously suffering scientific research and development which constitutes a favorable context and contribute to new achievements in data mining based on semantic similarity itself. In this chapter some of the most relevant documented work concerning the aforementioned context is identified.

2.1 Data Mining

Being a truly interdisciplinary subject, data mining can be defined in many different ways, figure 1, for instance, illustrates it as a step in the process of KDD shown as an iterative sequence of [12]:

1. Data cleaning – to remove noise and inconsistent data;
2. Data integration – where multiple data sources may be combined;
3. Data selection – where data relevant to the analysis task are retrieved from the database;
4. Data transformation – where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations;
5. Data mining – an essential process where intelligent methods are applied to extract data patterns;
6. Pattern evaluation – to identify the truly interesting patterns representing knowledge based on *interestingness measures*;

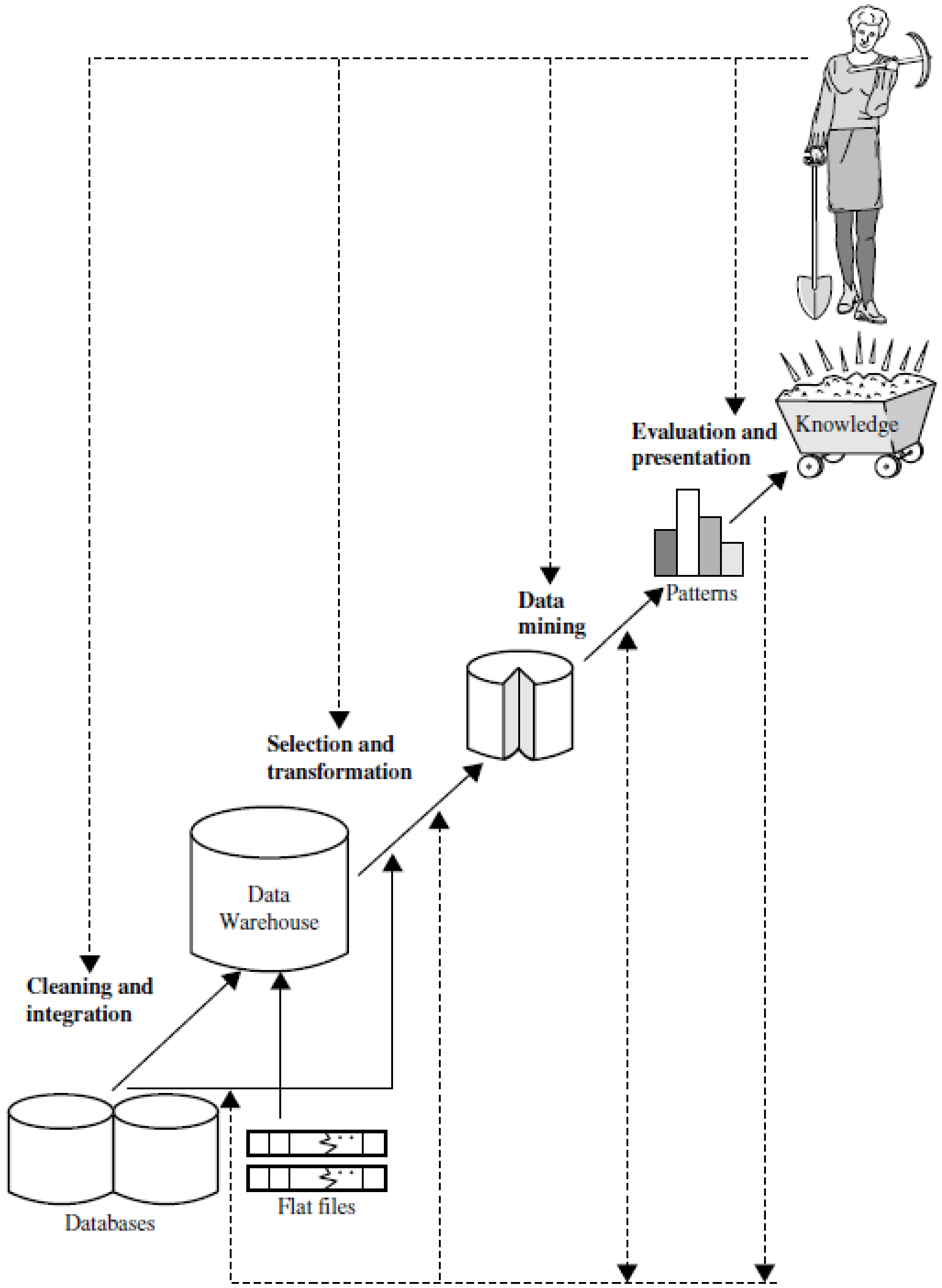


Figure 1: Data mining as a step in the process of knowledge discovery.

7. Knowledge presentation – where visualization and knowledge representation techniques are used to present mined knowledge to users.

However, in industry, in media and in the research milieu, the designation data mining is often used to refer to the entire knowledge discovery process (perhaps because it is shorter than KDD). Therefore, it seems appropriate to adopt a broad view of data mining functionality defining it as the process of discovering interesting patterns and knowledge from large amounts of data.

2.1.1 Clustering

In this work clustering analysis was the main data mining functionality adopted and was applied to ontology data. It is a fundamental technique in unsupervised learning, since it is able to discover the natural groupings of a set of unlabelled objects. It can be applied to a number of tasks including natural classification (of objects into classes), understanding the underlying structure of data, to support for instance anomaly detection, and as a method for compressing and summarizing data [13].

The clustering method k-means is the base of the clustering algorithms used in the present work. It is one of the most well-known and commonly used methods of the simplest and most fundamental version of cluster analysis, partitioning. These methods organize the objects of a set into several exclusive groups or clusters being the number of target clusters – starting point parameter for partitioning methods – given.

Formally, given a data-set, D, of n objects, and k, the number of clusters to form, a partitioning algorithm organizes the objects into k partitions $(k \leq n)$, where each partition represents a cluster. The clusters are formed to optimize an objective partitioning criterion, such as a dissimilarity function based on distance, so that the objects within a cluster are "similar" to one another and "dissimilar" to objects in other clusters in terms of the data-set attributes [12].

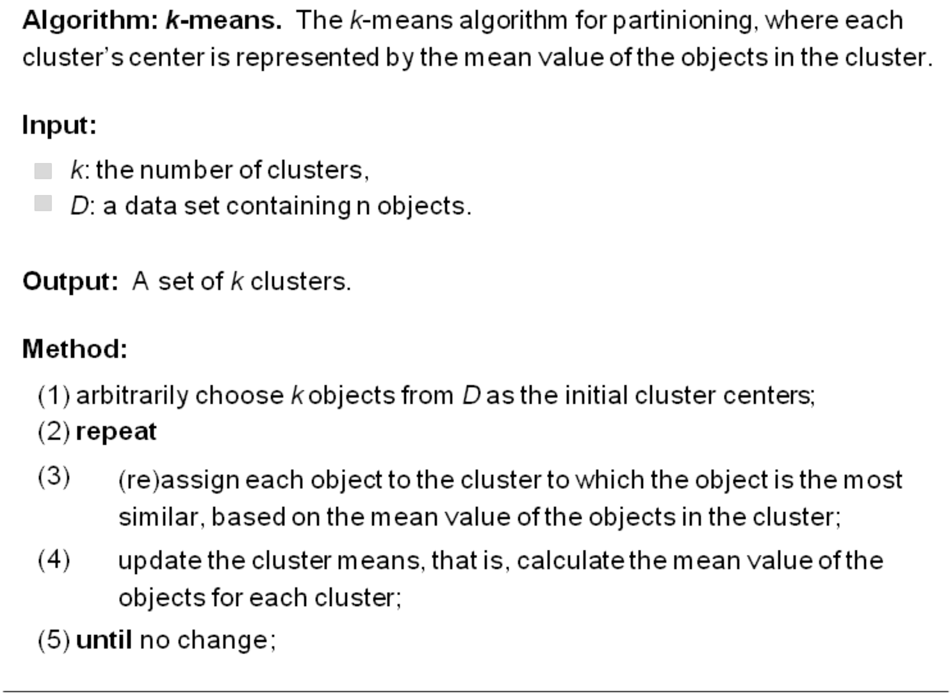
2.1.1.1 k-Means, a Centroid-Based Technique

This is a centroid-based partitioning technique. It uses the centroid of a cluster, C_i, to represent that cluster. Conceptually, the centroid of a cluster is its center point. The centroid can be defined in various ways such as by the mean or medoid of the objects (or points) assigned to the cluster. The difference between an object $\mathbf{p} \in C_i$ and $\mathbf{c_i}$, the representative of the cluster, is measured by $dist(\mathbf{p}, \mathbf{c_i})$, where $dist(\mathbf{x}, \mathbf{y})$ is the Euclidean distance between two points $\mathbf{x}$ and $\mathbf{y}$. The quality of cluster C_i can be measured by the within-cluster variation, which is the sum of squared error between all objects in C_i and the centroid $\mathbf{c_i}$, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(\mathbf{p}, \mathbf{c_i})^2$$

where E is the sum of the squared error for all objects in the data set. In other words, for each object in each cluster, the distance from the object to its cluster center is squared, and the distances are summed. This objective function tries to make the resulting k clusters as compact and as separate as possible.

Optimizing the within-cluster variation is computationally challenging. In the worst case, it would be necessary to enumerate a number of possible partitionings that are exponential to the number of clusters, and check the within-cluster variation values. It has been shown that the problem is NP-hard in general Euclidean space even for two clusters (i.e., $k = 2$). Moreover, the problem is NP-hard for a general number of clusters k even in the 2-D Euclidean space. If the number of clusters k and the dimensionality of the space d are fixed, the problem can be solved in time $O(n^{dk+1} \log n)$, where n is the number of objects. To overcome the prohibitive computational cost for the exact solution, greedy approaches are often used in practice. A prime example is the k-means algorithm, which is simple and commonly used.



Algorithm: *k*-means. The *k*-means algorithm for partinioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

- *k*: the number of clusters,
- *D*: a data set containing n objects.

Output: A set of *k* clusters.

Method:

```
(1) arbitrarily choose k objects from D as the initial cluster centers;
(2) repeat
(3)     (re)assign each object to the cluster to which the object is the most
        similar, based on the mean value of the objects in the cluster;
(4)     update the cluster means, that is, calculate the mean value of the
        objects for each cluster;
(5) until no change;
```

Figure 2: The k-means partitioning algorithm.

The k-means algorithm defines the centroid of a cluster as the mean value of the points within the cluster. It proceeds as follows. First, it randomly selects k of the objects in data set D, each of which initially represents a cluster mean or center. For each of the remaining objects, an object is assigned to the cluster to which it is the most similar, based on the Euclidean distance between the object and the cluster mean. The k-means algorithm then iteratively improves the within-cluster variation. For each cluster, it computes the new mean using the objects assigned to the cluster in the previous iteration. All the objects are then reassigned using the updated means as the new cluster centers. The iterations continue until the assignment is stable, that is, the clusters formed in the current round are the same as those formed in the previous round. The k-means procedure is summarized in figure 2 [12].

Example of Clustering by k-Means Partitioning Considering a set of objects located in 2-D space, as depicted in figure 3(a). Let $k = 3$, that is, the user would like the objects to be partitioned into three clusters.

According to the algorithm in figure 2, we arbitrarily choose three objects as the three initial cluster centers, where cluster centers are marked by a $+$. Each object is assigned to a cluster based on the cluster center to which it is the nearest. Such a distribution forms silhouettes encircled by dotted curves, as shown in figure 3(a).

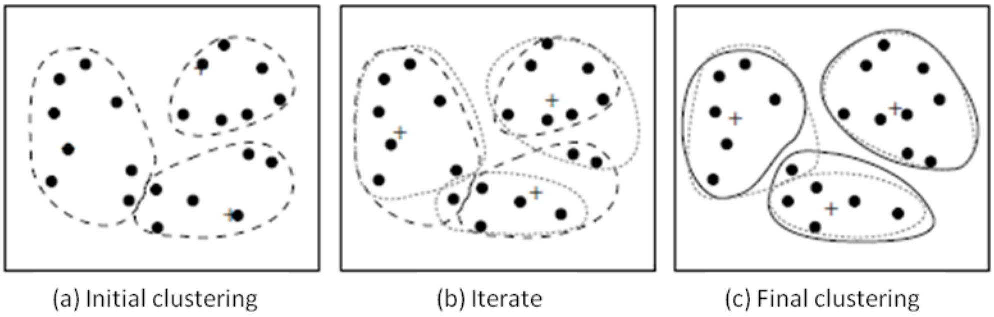


Figure 3: Clustering of a set of objects using the k-means method; for (b) update cluster centers and reassign objects accordingly (the mean of each cluster is marked by a $+$).

Next, the cluster centers are updated. That is, the mean value of each cluster is recalculated based on the current objects in the cluster. Using the new cluster centers, the objects are redistributed to the clusters based on which cluster center is the nearest. Such a redistribution forms new silhouettes encircled by dashed curves, as shown in figure 3(b).

This process iterates, leading to figure 3(c). The process of iteratively reassigning objects to clusters to improve the partitioning is referred to as iterative relocation. Eventually, no reassignment of the objects in any cluster occurs and so the process terminates. The resulting clusters are returned by the clustering process [12].

2.2 Machine Learning

How can we build computer systems that automatically improve with experience, and what are the fundamental laws that govern all learning processes [14]? Tom M. Mitchell defines this as the central question studied by machine learning. Since 1985, when there were almost no commercial applications, until now a huge progress is said to have occurred in machine learning and that it can be measured by its significant real-world applications which include: speech recognition, computer vision, bio-surveillance and robot control. The field's methods are identified as being the best available for developing particular types of software, namely where the application: is too complex for people to manually design the algorithm or requires that the software customize to its operational environment after it is fielded. Another role of machine learning is stressed by the author, its potential to reshape the way Computer Science is viewed by shifting the question from how to program computers to how to allow them to program themselves emphasizing the design of self monitoring systems that self-diagnose and self-repair, and approaches that model their users and take advantage of the steady stream of data flowing through the program rather than simply processing it. Substantial progress is suggested to have already been made in the development of machine learning algorithms and their underlying theory. For instance, there are a variety of algorithms for supervised learning of classification and regression functions, i.e., for learning some initially unknown function $f : X \to Y$ given a set of labeled training examples $\{(x_i, y_i)\}$ of inputs x_i and outputs $y_i = f(x_i)$. There are, of course, many other types of learning problems and associated algorithms like the one most relevant to the present work, unsupervised clustering (e.g., cluster genes based on their time series expression patterns), and others like anomaly detection, reinforcement learning or data modeling.

Machine learning algorithms play a very important role in data mining processes since these require techniques for finding and describing structural patterns in data, as a tool for helping to explain that data and make predictions from it [15].

2.2.1 Data Mining Tools with Machine Learning Algorithms

In a recent poll regarding the use of data mining tools in real projects, it is interesting to observe that in the top five there is only one commercial tool: Excel. The domination of free tools probably stems from the maturity and availability of a large number of machine learning algorithm implementations [16]. The most popular freely available data mining tools that have grown more efficient and useful over the years, some even comparable or better in certain aspects than their commercial counterparts, include:

- Waikato Environment for Knowledge Analysis (WEKA) – Java based, open source data mining platform developed at the University of Waikato, New Zealand. Has had mostly stable popularity over the years, which is mainly due to its user friendliness and the availability of a large number of implemented algorithms. It is still not as popular as other tools, both in business and academic circles, mostly because of some slow and more resource demanding implementations of data mining algorithms. It is still quite powerful and versatile [17].

- R – This open-source tool and programming language of choice for statisticians is also a strong option for data mining tasks. The source code is written in C++, Fortran, and in R itself. It is an interpreted language and is mostly optimized for matrix based calculations, comparable in performance to commercially available Matlab. Offers very fast implementations of many machine learning algorithms, comparable in number to WEKA (from which a large number of algorithms is borrowed), and also the full prospect of statistical data visualization methods [18].

- Konstanz Information Miner (KNIME) – A general purpose tool based on the Eclipse platform. It is open-source, though commercial licenses exist for companies requiring professional technical support. According to the official website, it is used by over three thousand organizations in more than sixty countries, and there seems to be a considerable community support. One of the greatest strengths of KNIME is the integration with WEKA and R. Although extensions have to be installed to enable the integration, the installation itself is trivial [19].

- scikit-learn – A free package in Python that extends the functionality of NumPy and SciPy packages with numerous data mining algorithms. It also uses the matplotlib package for plotting charts. The package keeps improving by accepting valuable contributions from many contributors. One of its main strong points is a well written online documentation for all of the implemented algorithms. Well written documentation is a requirement for any contributor and is valued more than a lot of poorly documented algorithm implementations [20].

2.3 Semantic Web

Nowadays the availability of electronic resources is permanently increasing making their organization and efficient access difficult. The semantic web initiative is precisely about adding formal structures and semantics (meta-data and knowledge) to web content for easy management and access. To make resources machine-understandable, it proposes in particular to enrich them with descriptions called annotations [21]. The concept annotation is defined in the oxford dictionary as "a note by way of explanation or comment added to a text or diagram". Besides this basic meaning, a semantic annotation (also called conceptual annotation) has two more important features: machines can read and process it and contains a set of formal and shared terms for a certain domain. A semantic annotation provides formal meaning to the data object, in a machine readable format, typically in the form of an attribution of a class Internationalized Resource Identifier (IRI) to an entity [22]. For instance, the human protein for hemoglobin can be described as having the molecular function "oxygen binding" (http://purl.obolibrary.org/obo/GO_0019825).

Semantic annotations use formal knowledge to capture annotator's knowledge and then act as a knowledge carrier to enrich annotated object's semantics [22]. The used formal knowledge may assume the form of an ontology which, in its classical sense is a philosophical discipline, a branch of philosophy that deals with the nature and the organization of being, but in computer science it refers to an engineering artifact, describing a formal, shared conceptualization of a particular domain of interest [23]. The three main components of an ontology are:

- Classes (or concepts) – Provide the abstraction mechanism for grouping resources with similar characteristics. Classes have an intentional meaning (their underlying concept) which is related but not equal to their extension (the instances that compose the class). Classes are typically identified by a unique code in the form of an IRI;

- Relations – An ontology relation is a binary relation established between classes (or concepts) like, for instance, class-subclass or part-whole;

- Instances (or individuals) – Are individual objects, each pertaining to a domain.

Ontologies are usually represented by labeled graphs where nodes represent the classes and edges the relations between them [24].

2.3.1 Semantic Similarity

As mentioned before, every data mining functionality depends on some kind of distance/similarity measuring between data instances. This work focused on exploring data semantics using semantic similarity measures so the following definitions of commonly used expressions are important:

- Semantic relatedness – Strength of the semantic interactions between two elements without restriction regarding the types of semantic links considered;

- Semantic similarity – Specializes the notion of semantic relatedness, by only considering taxonomical relationships in the evaluation of the semantic strength between two elements;

- Semantic distance – Generally considered as the inverse of semantic relatedness, all semantic interactions between the compared elements are considered.

In other words, semantic similarity measures compare elements regarding the properties they share and the properties which are specific to them [25]. Table 1 shows a quite updated summary of term semantic similarity measures — information content (IC), maximum informative common ancestor (MICA), all common ancestors (ACA) and vector space models (VSM) are used acronyms. It was extracted from an important work where an updated overview of term semantic similarity measures as well as their assessment and comparison is made.

Pairwise measures are those quantifying the similarity of two terms, whereas measures able to describe the relatedness of two sets of terms, yielding a global similarity of sets, are referred to as groupwise measures [57]. Lin [45] is an example of a pairwise node-based semantic similarity measure, it measures the similarity between two terms c_1 and c_2 as:

$$sim_{Lin}(c_1, c_2) = \frac{2 \times IC(c_{MICA})}{IC(c_1) + IC(c_2)}$$

Type	Name	Ref.	Term	MICA	ACA	Path	Term	VSM
			IC			Length	Depth	
Groupwise	Ali and Deane	[26]	✗	✓	✗	✗	✗	✗
	Cho	[27]	✓	✓	✗	✗	✗	✗
	Cosine	[28]	✗	✗	✗	✗	✗	✓
	Czekanowski-Dice	[29]	✗	✗	✓	✗	✗	✗
	Dice	[28]	✗	✗	✓	✗	✗	✗
	FMS	[30]	✓	✗	✗	✗	✗	✗
	IntelliGO	[31]	✓	✓	✗	✓	✓	✓
	Jaccard	[28]	✗	✗	✓	✗	✗	✗
	Kappa statistics	[32]	✗	✗	✓	✗	✗	✗
	NTO	[33]	✗	✗	✓	✗	✗	✗
	PL	[34]	✗	✗	✗	✓	✗	✗
	simGIC	[35]	✓	✗	✓	✗	✗	✗
	simLP	[36]	✗	✓	✗	✗	✓	✗
	simNLP	[37]	✗	✓	✗	✗	✓	✗
	simUI	[36]	✗	✗	✓	✗	✗	✗
	SSA	[38]	✓	✓	Depends on measure used			
	TO	[39]	✗	✗	✓	✗	✗	✗
	TAS	[40]	✗	✓	✗	✗	✗	✗
	Weighted cosine	[41]	✓	✗	✗	✗	✗	✓
	WJ	[28]	✓	✗	✓	✗	✗	✗
Pairwise	Annotation cosine	[42]	✗	✗	✗	✗	✗	✓
	G-SESAME	[32]	✗	✗	✓	✓	✗	✗
	GraSM	[43]	✓	✓	✗	✗	✗	✗
	Jiang and Conrath	[44]	✓	✓	✗	✗	✗	✗
	Lin	[45]	✓	✓	✗	✗	✗	✗
	Othman	[46]	✓	✓	✗	✓	✓	✗
	PS or PK-TS	[47]	✗	✓	✗	✓	✓	✗
	Resnik	[48]	✓	✓	✗	✗	✗	✗
	RSS	[49]	✗	✓	✗	✓	✓	✗
	SB-TS	[50]	✗	✗	✗	✗	✓	✗
	simIC	[51]	✓	✓	✗	✗	✗	✗
	simRel	[52]	✓	✓	✗	✗	✗	✗
	SSM	[53]	✓	✓	✗	✓	✓	✗
	TCSS	[54]	✓	✓	✗	✗	✗	✗
	Wu	[30]	✗	✗	✓	✗	✗	✗
	Wu-Palmer	[55]	✗	✓	✗	✓	✓	✗
	XOA	[56]	Depends on measure used					✓

Columns Term IC, Some common ancestors (MICA), All common ancestors, Path length, Term depth and VSM refer to the features of the measures described in the text. NTO, normalized term overlap; PL, path length; PS or PK-TS, pekar-staab term similarity; SSA, semantic similarity of annotations; TO, term overlap; TAS, total ancestry similarity; WJ, weighted Jaccard; XOA, cross ontological analysis.

Table 1: Summary of term semantic similarity measures [57].

On the other hand, $simGIC$ (GIC standing for graph information content) [35] is an example of a groupwise graph-based semantic similarity measure in which each term is weighted by its IC. It was developed to explore gene ontology (for annotating gene products) terms, being A and B two proteins with terms t, it is given by:

$$simGIC(A, B) = \frac{\sum_{t \in A \cap B} IC(t)}{\sum_{t \in A \cup B} IC(t)}$$

The IC of a concept provides an estimation of its degree of generality/concreteness, a dimension which enables a better understanding of concept's semantics. As a result, IC has been successfully applied to the automatic assessment of the semantic similarity between concepts. Sánchez [58] proposed a new intrinsic IC computational model where the IC of a concept c is defined as:

$$IC(c) = -log\left(\frac{\frac{|leaves(c)|}{|subsumers(c)|} + 1}{max\_leaves + 1}\right)$$

Where $leaves(c)$ and $subsumers(c)$ contain, respectively, the taxonomical concepts above and bellow the concept c and max_leaves represents the number of leaves corresponding to the root node of the hierarchy.

Semantic similarity measures can be used to compute the similarity between data annotated with ontologies [59]. These measures are able to compare ontology classes or entities annotated with ontology classes and return a numerical value reflecting their closeness in meaning. Many measures make use of the concept of information content, which describes how meaningful an ontology class is either based on structural properties, corpora usage or a combination of both.

Several semantic similarity measures have been proposed in the last decade, and more recently there have been efforts in extending these measures to work for entities annotated with multiple ontologies. The exploitation of multiple ontologies provides additional knowledge that can improve the similarity estimation and solve cases in which terms are not represented in an individual ontology. This is especially interesting in domains such as the biomedical one, in which several big and detailed ontologies are available, offering overlapping and complementary knowledge about the same topics [60].

While presenting semantic similarity calculation improvement opportunities the use of multiple ontologies also poses new issues. For instance, some concepts related to a concept in a given ontology may not be seen in that ontology however, these related concepts exist in other ontologies. The issue that stands out here is that the ontologies have different granularity degrees, and so, each ontology reflects a different similarity scale. For measuring cross-ontology similarity of concepts, variables like the granularity of ontologies must be taken into account [61].

2.3.1.1 Software Tools for Calculating Semantic Similarity

A natural demand in research is the development of software tools that implement available methods for calculating semantic similarity of terms in an ontology and that of entities annotated with an ontology. So far, there have been quite a few such software tools available, with examples including, among many others:

- SimPack – A framework of similarity measures adapted to the use in ontologies. It offers a variety of different semantic similarity measures, is generic, i.e., it can be applied to different data structures given the existence of appropriate data assessors and, it is implemented in Java, thus portable [62];

- seGOsa – User-friendly cross-platform system to support large-scale assessment of GO driven similarity among gene products. Using information-theoretic approaches, the system exploits both topological features of the GO and statistical features of the model organism databases annotated to GO to assess semantic similarity among gene products [63];

- DOSim – R-based software package to compute the similarity between diseases and to measure the similarity between human genes in terms of diseases. It incorporates an enrichment analysis function based on the disease ontology and uses this function to explore the disease feature of an independent gene set [63];

- Semantic Measures Library (SML) – To date the most complete on this area. It is a generic (i.e., not domain-specific) and open source Java library and command line software dedicated to the computation and analysis of knowledge-based semantic measures. It can be used to compare a pair of concepts or two groups of concepts defined in a semantic graph and supports various types of formats and languages used to express knowledge representations, e.g., Resource Description Framework (RDF), Open Biomedical Ontologies (OBO) and Web Ontology Language (OWL) [25].

2.4 Clustering with Ontologies

Several clustering techniques rely on the definition of a distance metric, which is used by the clustering algorithms to find the best possible groupings. Typical distance metrics operate over numerical data (e.g. Euclidian, Manhattan) or categorical data (e.g., Mahalanobis), but are unable to handle the semantic content of data objects to perceive their similarity. For instance, imagine the following scenario, where there are three patients (A, B and C):

- A has been diagnosed with Type II diabetes;

- B with insulin resistance;
- C with estrogen resistance.

Each patient is thus described by their diagnosis. Using a typical categorical distance, all patients are equally distant. Using a string similarity based distance, B and C are more closely related. However, medically speaking, A and B are actually much more similar, since insulin resistance is a precursor to the development of Type II diabetes. This kind of similarity can be captured by using ontologies, since they model the concepts and relations in a given domain.

Some existing works employ ontology-based clustering. Maedche et al. [23] proposed an approach based on hierarchical clustering using similarities between ontology instances along three dimensions: taxonomy, relation and attribute similarity. The authors carried out an empirical evaluation due to the lack of ontological background knowledge. Another relevant related work presents a complementary approach to pure hierarchical clustering making use of the classification hierarchy common to ontologies. Ontologies encoded in an extended form of RDF/RDFS are combined with an established hierarchical clustering system to achieve results that, on one hand hold promise for applications of dictionary-based ontologies in information retrieval tasks and, on the other hand, raise an important question: how to quantify the significance of ontological clustering beyond the similar effects of the meta-word search? The results establish a baseline in which hierarchical clustering using ontologies is at least as good as meta-word search [64]. More recently, an approach that combines semantic similarity of variables with hierarchical clustering has been shown to produce good results on a set of linguistic benchmarks [65]. These works all share two limitations, they provide tailored semantic similarity measures, preventing their easy adaptation to other domains that may well necessitate a different metric, and moreover, they only work using a single ontology.

Chapter 3

Clustering with Semantic Based Distances

The purpose of this work is to extend typical distance-based clustering approaches with semantic distance. The overall framework consists in allowing the computation of semantic distances for data annotated with multiple ontologies, which are then used as distance values for clustering approaches. Figure 4 schematically represents the global framework, challenges and expected achievements. The goal is a system from which knowledge can be extracted using semantic information from more than one ontology to implement clustering based on semantic distances. There were a few requirements concerning the software libraries to be used on the system's development, they should be:

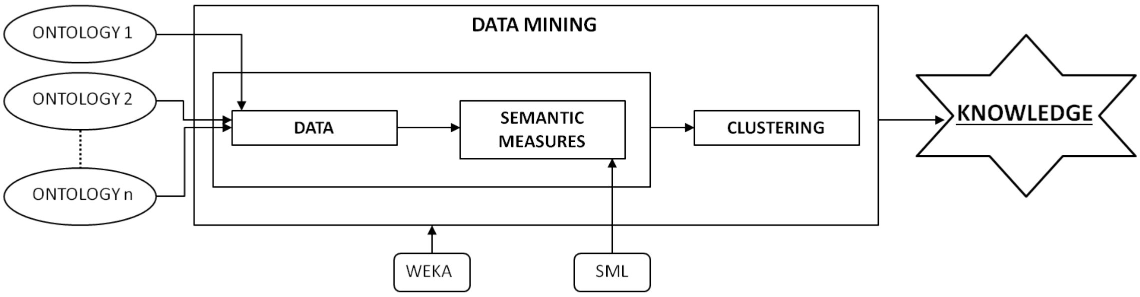


Figure 4: Schematic representation of the software solution's global framework.

- Free, open source;
- State of the art references;
- Easy to integrate with each other;
- Extensible;
- Very well documented.

In line with these requirements, the resulting implementation utilizes the two following libraries:

- WEKA for machine learning algorithms – Considered a landmark system, widely adopted by machine learning and data mining communities as an educational tool and also widely used in commercial settings. Its main features are, in summary, data preprocessing, classification, clustering, attribute selection and data visualization;

- SML for semantic similarity calculations – It is, from several available software solutions for the computation of knowledge-based measures (type of semantic measures adequate to the intention to use ontologies as the form of knowledge representation from which to extract the semantics associated to the compared elements), the most complete. In this work, it was further extended to allow the computation of semantic similarity using multiple ontologies.

This software solution – designated SESAME as an invocation of the magical phrase from the story of "Ali Baba and the forty thieves", here in the sense that it aims to favor the integration of semantic space potential in data mining processes – was completely developed using Java programming language in Eclipse Integrated Development Environment (IDE) with SML and WEKA Java libraries added in Java Archive (JAR) format.

3.1 SESAME

The components resulting from the software development efforts to create SESAME as a data mining application program for clustering based on semantic measures, are represented in the diagram of figure 5.

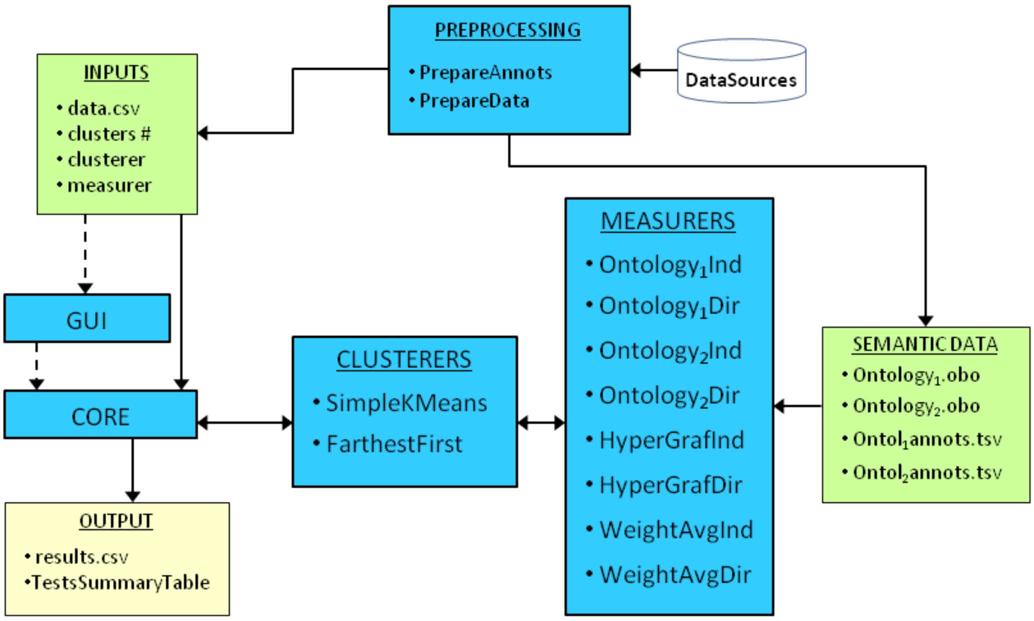


Figure 5: SESAME's components.

3.1.1 Semantic Data and Inputs

SESAME takes two types of data, the variable inputs defined by the user for each application run and the semantic data files that must be constantly available in the same file system path as the application itself. The run specific, user defined inputs are:

- File in Comma-Separated Values (CSV) format, *data.csv*, having in its lines the instances to cluster, table 2 shows an example – the formation of these files requires yet another data source concerning the instances' class assignments;
- Target number of clusters which must coincide with the number of different classes of instances in the corresponding input file (seven in the example shown in table 2);

Entry	Chebi	Class
http://SESAME/P60709	No	SignalingbyEGFR
http://SESAME/P63261	No	SignalingbyEGFR
http://SESAME/O14672	Yes	SignalingbyEGFR
http://SESAME/P46109	No	SignalingbyPDGF
http://SESAME/P09958	Yes	SignalingbyPDGF
http://SESAME/Q14451	No	SignalingbyPDGF
http://SESAME/P16333	No	SignalingbyPDGF
http://SESAME/P08559	Yes	SignalingbyRetinoicAcid
http://SESAME/P11177	Yes	SignalingbyRetinoicAcid
http://SESAME/O00330	No	SignalingbyRetinoicAcid
http://SESAME/Q15118	No	SignalingbyRetinoicAcid
http://SESAME/Q06187	Yes	SignalingbyRhoGTPases
http://SESAME/O43684	No	SignalingbyRhoGTPases
http://SESAME/P13498	No	SignalingbyRhoGTPases
http://SESAME/P04839	Yes	SignalingbyRhoGTPases
http://SESAME/P06756	No	SignalingbyVEGF
http://SESAME/P05771	Yes	SignalingbyVEGF
http://SESAME/Q05513	No	SignalingbyVEGF
http://SESAME/Q15759	Yes	SignalingbyVEGF
http://SESAME/Q9HCK8	No	SignalingbyWnt
http://SESAME/P18545	No	SignalingbyWnt
http://SESAME/P25963	No	SignalingbyNGF
http://SESAME/O14920	No	SignalingbyNGF
http://SESAME/P51617	Yes	SignalingbyNGF
http://SESAME/Q15418	Yes	SignalingbyNGF

Table 2: Example of an input file's content, instances to cluster. Entries are proteins identifiers in the semantic graph, the attribute Chebi just informs about the existence of ChEBI annotation and the attribute Class identifies each instance's metabolic pathway.

- Clusterer – clustering algorithm option;
- Measurer – semantic measuring configuration, i.e. semantic measure plus graph loading combination option.

The necessary semantic data elements, used to load semantic graphs and calculate semantic distances according to user selected semantic measure plus graph loading combination, are:

- Ontologies in OBO format;
- Annotations files in Tab-Separated Values (TSV) format.

3.1.2 Preprocessing

This component concerns to what could be designated as logistic tasks: necessary preparations before running the application when it is necessary to collect the desired data from the selected data sources and then:

- Prepare annotations files – Annotations files are produced by converting files extracted from the chosen data sources to TSV. For each chosen ontology one file is necessary with a line per entity. Each line with two columns, one with the entity's identifier and the other with its annotations with the corresponding ontology. These preparations are made only once for all runs corresponding to clustering tasks using the same ontologies.
- Prepare *data.csv* files – Files containing each a list of instances of a specific class are joined in a CSV formatted file. As shown in the example of table 2, each line of this file contains an instance having in a first column instances' identifiers concatenated with the string *http://SESAME/* to form a proper Uniform Resource Identifier (URI), in a second (optional) column chosen verification information about the data and, in a third column, that instance's class. In the process of creating these CSV files, there are no missing attribute values, instance repetitions are avoided and their annotations are checked using the annotations files (to avoid not annotated data objects, which would result in runtime errors, and to know with which ontology each instance is annotated). SESAME converts these CSV files to WEKA's Attribute-Relation File Format (ARFF) so they must also have a header line identifying each of the three columns of data.

3.1.3 SESAME's GUI

The application includes a Graphical User Interface (GUI) to provide its users a kind of interface that is indispensable if not for the entire scientific community, at least for those who dislike command line instructions or text-based interfaces. WEKA provides a GUI but this one is exclusively dedicated to SESAME's functionalities therefore provides a more specific interaction not subject to WEKA's version broader characteristics. It is based on *WindowBuilder*, a plug-in for Eclipse IDE which makes it very easy to create Java GUI applications without spending a lot of time writing code. Using a visual designer and layout tools simplifies adding controls using drag-and-drop, adding event handlers to those controls or changing various properties of controls using a property editor. The corresponding Java code is automatically generated. Figure 6 shows the created JFrame.

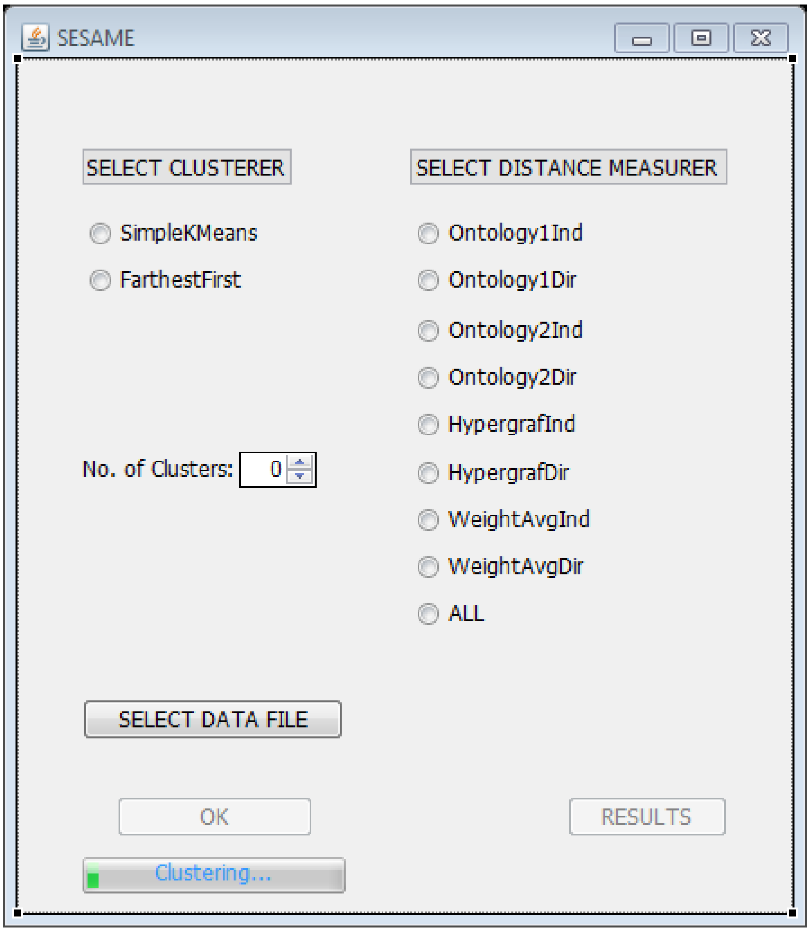


Figure 6: SESAME's GUI.

The user must select a clusterer and a distance measurer (only one of each can be selected), the target number of clusters (less than 51 and if 0 a rule of thumb is used) and a data file (through a common file system navigator). Only then the OK button becomes available allowing to proceed to clustering. When OK is pressed and until the clustering ends a progress bar shows. Eventually, a button to visualize the TSV formatted results file in a spreadsheet is also made available. The distance measurer option ALL allows to run the application with the selected clusterer, number of target clusters, data file and all the available measurers.

3.1.3.1 Alternative User Interface Options

SESAME's GUI is its only user interface that allows an unrestricted number of clustering tasks, with different options or not, to be run in the same application session. Users who consider this feature unimportant may chose one of the following interface options.

Command Line A simple text-based interface is provided if the application is run from command line without specifying the input parameters. This way the user makes clusterer, measurer and number of target clusters choices. The data file must be in the application's path and be named "data.csv". To specify input parameters, the following sequence must be observed:

```
N [clusters #] -t [data file path] - A [measurer option] -C [clusterer option]
```

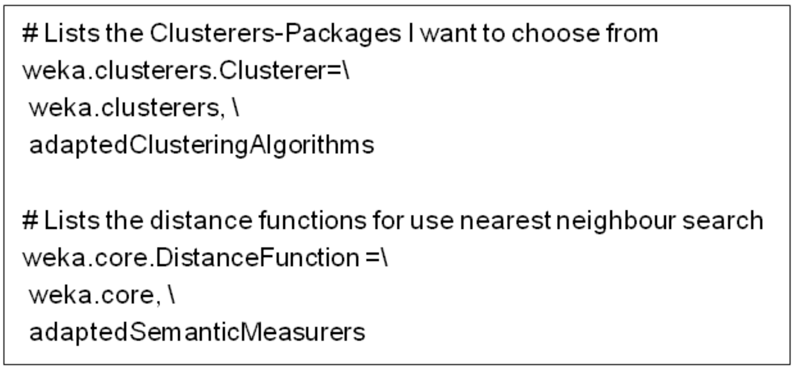
In either possibility:

- The number of target clusters must be an integer in [0,50] (0 for a "rule of thumb" number of target clusters definition);
- The data file path must comply to windows operating system requirements;
- The available measurer options are 1 for *Ontology1Ind*, 2 for *Ontology1Dir*, 3 for *Ontology2Ind*, 4 for *Ontology2Dir*, 5 for *HypergraphInd*, 6 for *HypergraphDir*, 7 for *WeightAvgInd* and 8 for *WeightAvgDir*;
- The available clusterer options are 1 for adapted SimplekMeans and 2 for adapted Farthest First.

WEKA Explorer GUI The Explorer is an important component of WEKA that provides a graphical environment from which users may configure and launch all the available data mining options. One of WEKA's most relevant strengths is its adaptability and extensibility and the explorer is not an exception.

As of version 3.4.4 it is possible for WEKA to dynamically discover classes at runtime. To enable or disable dynamic class discovery, the relevant file to edit is *GenericPropertiesCreator.props* (GPC) which can be obtained from the *weka.jar* archive. All that is required, is to change the *UseDynamic* property in this file from *false* to *true* (for enabling it) or the other way around (for disabling it). After being changed, the file must be placed in the home directory [66]. For the present work, this property was set to true and the system's environment variable *CLASSPATH* (which tells Java where to look for classes) was configured to include WEKA and SML used libraries in JAR format as well as SESAME's main class, *MyFirstCluster*, also in JAR format.

Since version 3.4.4, WEKA can also display multiple class hierarchies in its GUI which makes adding new functionalities quite easy. In the present work, adapted clusterers and distance functions were developed and located in packages *adaptedClusteringAlgorithms* and *adaptedSemanticMeasurers*, respectively. So, the file GPC add to be changed accordingly, as shown in figure 7.



```
# Lists the Clusterers-Packages I want to choose from
weka.clusterers.Clusterer=\
 weka.clusterers, \
 adaptedClusteringAlgorithms

# Lists the distance functions for use nearest neighbour search
weka.core.DistanceFunction =\
 weka.core, \
 adaptedSemanticMeasurers
```

Figure 7: *GenericPropertiesCreator.props* file's lines including SESAME's packages.

3.1.4 SESAME's Core

This component is the application's central hub. It receives the inputs, either from the GUI or command line, uses them to instruct clustering algorithms what to do and treats the results to produce an output file for each clustering task.

After some maintenance instructions, intended to control the flow of multiple clustering tasks in the same session (when the GUI is used), an output file name is defined reflecting the user's input options. Next, the chosen CSV format data file is loaded, converted to ARFF and only then used to set the data instances.

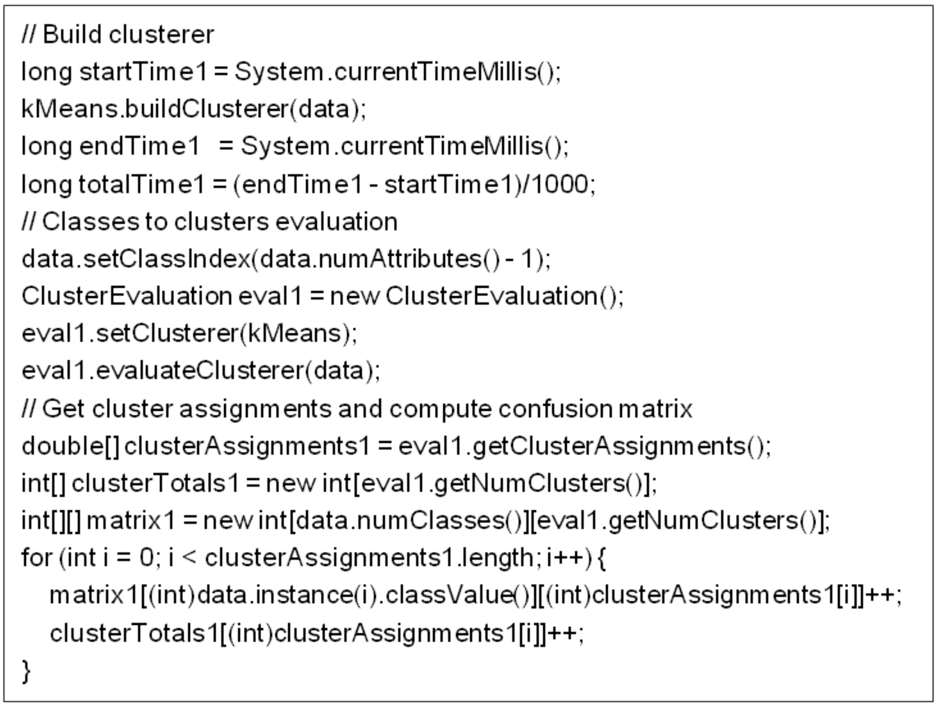
At this point, concrete clustering procedures are initiated by setting:

- The clustering algorithm;
- The semantic measuring configuration;
- The target number of clusters;
- The seed value – Used to initialize the random number generator. k-Means based clusterers like SimplekMeans and Farthest First set initial cluster's centroids by randomly selecting instances from the data. In SESAME, 42 is defined as the seed value;
- The option not to replace missing values – Used to set the replacement of all missing values for nominal and numeric attributes with the modes and means of the data. In SESAME it is set to *true* because the data is supposed to have no missing values.

Finally the clusterer is built, the build time counted and clustering evaluation can then be made as well as a consequent gathering of results' information like:

- Clustering assignments and confusion matrix;
- Statistics (clustering time, percentage and number of incorrectly clustered instances and percentage and number of instances per cluster);
- Classes assigned to clusters.

The source code instructions for this are shown in the excerpt of figure 8.



```
// Build clusterer
long startTime1 = System.currentTimeMillis();
kMeans.buildClusterer(data);
long endTime1   = System.currentTimeMillis();
long totalTime1 = (endTime1 - startTime1)/1000;
// Classes to clusters evaluation
data.setClassIndex(data.numAttributes() - 1);
ClusterEvaluation eval1 = new ClusterEvaluation();
eval1.setClusterer(kMeans);
eval1.evaluateClusterer(data);
// Get cluster assignments and compute confusion matrix
double[] clusterAssignments1 = eval1.getClusterAssignments();
int[] clusterTotals1 = new int[eval1.getNumClusters()];
int[][] matrix1 = new int[data.numClasses()][eval1.getNumClusters()];
for (int i = 0; i < clusterAssignments1.length; i++) {
   matrix1[(int)data.instance(i).classValue()][(int)clusterAssignments1[i]]++;
   clusterTotals1[(int)clusterAssignments1[i]]++;
}
```

Figure 8: SESAME's source code excerpt.

The last step of this component consists on registering all the computed clustering tasks' results in an output file.

3.1.5 Clustering Algorithms

Due to their distance based nature, suitable to the problem at hand, two of WEKA's clustering algorithms, identified as adaptation-prone, were selected for integration of semantic distance measuring options [67]:

- SimplekMeans – clusters data using k-means, a method which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean;
- Farthest First – a variant of k-means that places each cluster center, in turn, at the point (within the data area) farthest from the existing cluster centers.

3.1.5.1 Integrating SML into WEKA

SML provides the means to calculate semantic similarity values, bounded in $[0, 1]$, between pairs of ontology concepts or entities annotated with ontology concepts. Being normalized, these semantic similarity (sim) values, between two data instances (ent_a and ent_b) can be converted to semantic distances ($dist$) using the relation:

$$dist(ent_a, ent_b) = 1 - sim(ent_a, ent_b)$$

To implement clustering based on semantic measures these semantic distances must be made available to distance based clustering algorithms the same way conventionally used distance metrics are. WEKA provides a few clustering algorithms which are prepared to calculate Manhattan, Euclidean or Chebyshev distances. In programmatic terms this corresponds to having Java classes for each of these distance metrics all of them implementing the same interface, *DistanceFunction*. So, what's necessary to add SML's semantic distances calculation to WEKA is to develop new classes, each based on a desired reference graph configuration and semantic measure option (along with an associated IC specification), which implement the previously mentioned *DistanceFunction* interface. According to user options, the clustering algorithms then have the possibility to call any of the available distance calculation methods, either conventional or semantic based.

3.1.6 Semantic Distance

At its present state, SESAME is ready to use two ontologies and two corresponding annotations files so, the adapted clustering algorithms mentioned in the previous section were enriched with eight options for semantic distance measuring combining four reference graph loading options:

- Ontology$_1$ – reference graph loaded with first chosen ontology;
- Ontology$_2$ – reference graph loaded with second chosen ontology;
- HyperGraf – reference hyper-graph loaded with both chosen ontologies;
- WeightAvg – two reference graphs, one loaded with the first chosen ontology and the other with the second chosen ontology (final distance value based on weighted average values obtained from individual distances calculated using each graph),

(the latter two will be explained in greater detail in the next chapter) and two possible semantic measures (both using ontology-based IC computation [58]):

- Direct groupwise semantic measure SimGIC [35] (used in Ontology$_1$Dir, Ontology$_2$Dir, HyperGrafDir and WeightAvgDir);

- Indirect groupwise measure based on Lin's pairwise measure [45] with Best Match Average – in which each term of the first entity is paired only with the most similar term of the second one and vice versa [68] – aggregation strategy (used in Ontology$_1$Ind, Ontology$_2$Ind, HyperGrafInd and WeightAvgInd).

Nevertheless, the number of the application's semantic measuring configurations available may easily be increased if more than two ontologies ought to be used.

The two semantic similarity measures chosen fulfill the requirements established for the present work, where sets of concepts are to be compared implying the use of groupwise measures:

- One measure of the direct and another of the indirect approach types;
- Information theoretical (i.e., consider the IC of the concepts);
- Based on graph analysis;
- Reference measures of the respective types.

3.1.7 Output

SESAME's output consists in a file for each run which outline focus mainly in clearly providing the user all the information pertaining that run's results. Its content is divided in the following four parts:

1. Run – identifying the inputs (clusterer, measurer, number of target clusters and seed), clustering time and number and percentage of incorrectly clustered instances;
2. Instances – a list of all the clustered instances and, for each, the respective attributed cluster;
3. Clusters – the list of resulting numbered clusters and, for each, the respective centroid URI, number and percentage of instances and attributed class;
4. Confusion matrix – identifying the number of instances from each class assigned to each cluster, an example is shown in table 3.

It is common for a same user to run several tasks and be interested in having all the generated results available to analysis in a user-friendly way. Of course that just having a file for each run with all the aforementioned content would be cumbersome. SESAME includes a tool to, having a set of clustering tasks' results, facilitate their global analysis and evaluation. This useful tool creates a summary table of tests' results, like the example in table 4, in a TSV formatted file. Here, key application run results' values (clustering time and number and percentage of incorrectly clustered instances) and descriptors (clusterer,

Confusion Matrix:					
assigned to cluster –>	cluster 0	cluster 1	cluster 2	cluster 3	cluster 4
CellCycle	309	27	176	11	36
ChromatinOrganization	23	11	131	1	21
DevelopmentalBiology	333	101	166	40	22
DNARepair	1	0	172	0	0
ProgrammedCellDeath	48	16	18	5	1

Table 3: Example of output confusion matrix.

measurer, number of clusters and seed) for several runs of previously defined clustering tasks and groups of tasks are saved providing a simpler way to compare results than to open each run's complete results file one at a time.

NO LINK						
Test17-1242Prot_2clusters						
Clusterer	Measurer	Clusters (#)	Seed	Clustering Time (secs)	Incorrectly Clustered Instances (#)	Incorrectly Clustered Instances (%)
FF	GODir	2	42	69	464	37.35909823
FF	GOInd	2	42	69	464	37.35909823
SKM	WAvgDir	2	42	81	475	38.24476651
SKM	WAvgInd	2	42	90	520	41.86795491
Test18-944Prot_3clusters						
Clusterer	Measurer	Clusters (#)	Seed	Clustering Time (secs)	Incorrectly Clustered Instances (#)	Incorrectly Clustered Instances (%)
FF	ChEBIDir	3	42	17	33	32.67326733
FF	ChEBIInd	3	42	20	33	32.67326733
SKM	ChEBIInd	3	42	18	32	31.68316832
SKM	HypGrafDir	3	42	137	510	54.02542373
SAME GROUP						
Test15-1040Prot_7clusters						
Clusterer	Measurer	Clusters (#)	Seed	Clustering Time (secs)	Incorrectly Clustered Instances (#)	Incorrectly Clustered Instances (%)
FF	GODir	7	42	69	394	37.88461538
SKM	WAvgDir	7	42	95	571	54.90384615
SKM	WAvgInd	7	42	118	576	55.38461538

Table 4: Example of a summary table.

Chapter 4

Semantic Similarity with Multiple Ontologies

In response to the ever-increasing amount and variety of data, ontologies are becoming widely used to make information more computable. This context of ontology's construction and usage dissemination constitutes an important contribution to their increasing relevance particularly because:

- Different interpretations of reality can lead to complementary ontologies;
- A variety of domains of knowledge are getting represented as ontologies, especially by those more competent to do so, i.e. people with a background knowledge in these domains.

Semantic similarity can be computed with a single ontology, however, ontologies are often incomplete, due to the intrinsic uncertainty associated with their respective scientific field, they can also contain errors, or even follow a certain view of reality that is not shared by everyone. Multiple ontologies approaches provide complementary views of reality so that incompleteness, errors and subjective interpretations are mitigated.

It is important to stress that using multiple ontologies to compute semantic similarity may configure two scenarios [11]:

- Multiple ontologies, single domain similarity – two or more ontologies representing the same domain are used in a complementary way to improve semantic similarity results;
- Multiple ontologies, multiple domains similarity – represents a step beyond the previous approach since it uses multiple ontologies from distinct domains in order to compare concepts in a multidisciplinary context.

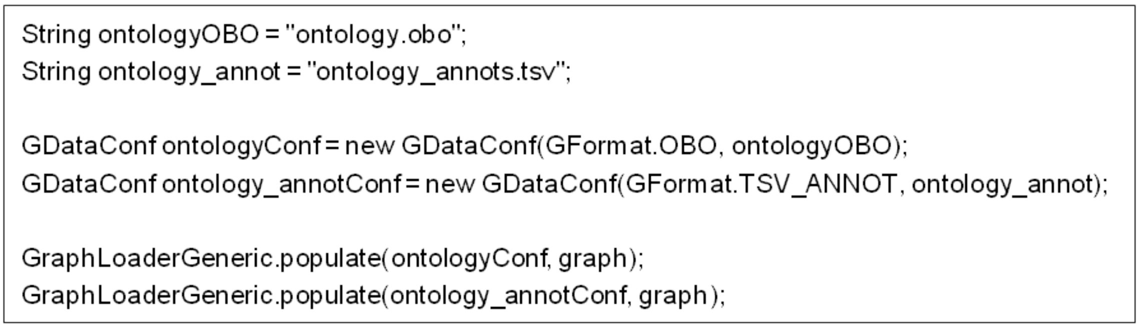
Given the multidisciplinarity of available information resources, implementing measures of similarity that can handle all the relevant domains is imperative. The hypothesis that multi-domain semantic similarity has some advantages compared to classical single-ontology measures when dealing with multidisciplinary resources has already been demonstrated [11] so, in the present work, this reality has been taken into consideration and specifically addressed.

4.1 Handling Multiple Ontologies in SESAME

While dealing with semantic similarity in single ontology or multiple ontologies single domain contexts is somehow frequent, the present work provides an approach prepared for both those approaches but also for multiple ontologies multiple domains, depending on user options.

Using multiple ontologies to calculate similarity is a process that strongly depends on links between the ontologies. This work relies on the fact that the instances being annotated with concepts from the involved ontologies may represent that necessary link between them. Using graph-based semantic measures, the computed semantic similarity values use semantic annotations from more than one ontology if they exist or only from one otherwise, imposing that at least with one of the ontologies all the instances must be annotated.

The developed application foresees the incorporation of as many ontologies as the users may find needed to their work. Notwithstanding the fact that at its present version only OBO format ontologies and TSV format annotations files are readily usable, future versions can accommodate additional formats among those supported by SML like RDF and OWL, in the case of ontologies, or GO Annotation File Format (GAF) in the case of annotations. Figure 9 shows the lines of code, present in every Java class of SESAME's *adaptedSemanticMeasures* package, defining the ontology and annotation formats (assuming ontology file *ontology.obo* and corresponding annotations file *ontology_annots.tsv* are in the same path as the application) and loading them to a graph, later used as reference for all semantic similarity calculations.



```
String ontologyOBO = "ontology.obo";
String ontology_annot = "ontology_annots.tsv";

GDataConf ontologyConf = new GDataConf(GFormat.OBO, ontologyOBO);
GDataConf ontology_annotConf = new GDataConf(GFormat.TSV_ANNOT, ontology_annot);

GraphLoaderGeneric.populate(ontologyConf, graph);
GraphLoaderGeneric.populate(ontology_annotConf, graph);
```

Figure 9: SESAME's code lines defining the ontology and annotation formats and loading them to a graph.

4.2 Extending SML to handle multiple ontologies

By providing additional entities' relationship information the use of semantic references from more than one ontology represents an opportunity to strengthen the potential quality of clustering processes. Handling the integration of multiple ontologies into a single semantic similarity measure is a recognized challenge. Here we have used two simple approaches: Hyper-graph and Weighted Average.

Hyper-graph One implemented way to achieve this integration was to create a hyper-graph containing the chosen ontologies' graphs. To do this in SML a re-rooting process must be fulfilled where a virtual root is created to link the graphs loaded with each of the involved ontologies. Figure 10 shows the root and first layer of the hyper-graph resulting from this re-rooting process in the practical case used for testing in the present work where GO – Biological Process (BP), Molecular Function (MF) and Cellular Component (CC) – and ChEBI ontology were used.

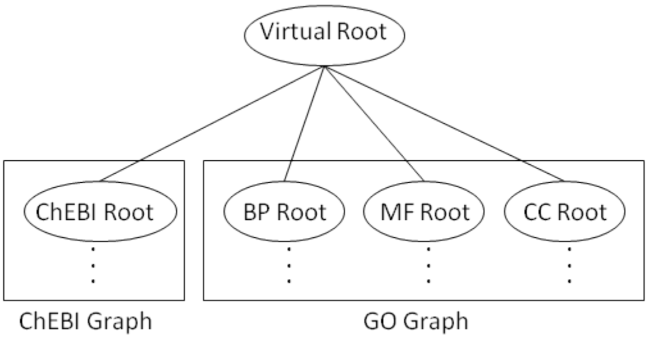


Figure 10: Root and first layer of hyper-graph containing GO and ChEBI graphs.

Weighted Average Another implemented way to benefit from having more than one ontological reference was to have separated graphs for each ontology and to calculate a final distance value based on weighted values obtained from individual distances calculated using each graph. The final distance between two instances is given by:

$$d(e_a, e_b) = \omega_1 d_{O_1}(e_a, e_b) + \omega_2 d_{O_2}(e_a, e_b) + ... + \omega_n d_{O_n}(e_a, e_b)$$

Where the final distance between entities (e_a and e_b) is a weighted average with ω_i weighting the contribution of each ontology O_i which is used to annotate the entities.

Chapter 5

Evaluation

Taking into consideration the relevance and growth of bioinformatics – in short, a management information system for molecular biology with many practical applications [69] – the software application development and testing were made for the specific case of protein clustering based on their annotations with concepts from GO and ChEBI.

The evaluation of each test application run was made with WEKA's classes to clusters method which compares how well the chosen clusters match pre-assigned classes in the data. Having clustered the data, WEKA determines the most represented class in each cluster and returns a confusion matrix showing discrepancies between clusters and true classes [15].

5.1 Chosen Data Sources

Annotations with concepts from GO and ChEBI were chosen to base protein clustering. Both these ontologies are available in OBO flat file format, an ontology representation language. This format attempts to achieve human readability, ease of parsing, extensibility and minimal redundancy. These characteristics conjugated with the fact that it is one of the formats accepted by SML are the reasons why GO and ChEBI in OBO format were used to test SESAME.

GO consortium members submit gene association files in GO annotation file format (GAF) which is not available for ChEBI annotations. Universal Protein Resource (UniProt), a comprehensive resource for protein sequence and annotation data [70], provides the means to obtain both GO and ChEBI annotations files. A simple query like *accession:* AND organism:"Homo sapiens (Human) [9606]"* in UniProt Knowledgebase (UniProtKB) and the selection of "Entry" and "Gene ontology IDs" as the sole results' columns provides a list of all human proteins and corresponding GO annotations (if existent). UniProtKB cofactor annotations are based on ChEBI so, a list of all human proteins and corresponding ChEBI annotations can be obtained using the same simple query but the selection of "Entry" and "Cofactor" as results' columns. Lists like these may then be downloaded and converted to TSV formatted files, also accepted by SML as annotations files. Unlike the GO annotations list file the ChEBI annotations list file demanded extensive additional work since each cofactor attribute includes more information, as shown in table 5, than just ChEBI identifiers separated by semicolons as required by SML's TSV formatted annotations files.

UniProt Cofactor Annotations File's Lines

Entry	Cofactor
P04637	COFACTOR: Name=Zn(2+); Xref=ChEBI:CHEBI:29105; ; Note=Binds 1 zinc ion per subunit.;
P00441	COFACTOR: Name=Cu cation; Xref=ChEBI:CHEBI:23378; Evidence=ECO:0000269|PubMed:17888947; ; Note=Binds 1 copper ion per subunit. ECO:0000269|PubMed:17888947;; COFACTOR: Name=Zn(2+); Xref=ChEBI:CHEBI:29105; Evidence=ECO:0000269|PubMed:17888947; ; Note=Binds 1 zinc ion per subunit. ECO:0000269|PubMed:17888947;

Corresponding ChEBI Annotations File's Lines

Entry	ChEBI
P04637	CHEBI:29105
P00441	CHEBI:23378;CHEBI:29105

Table 5: Example of ChEBI protein annotations obtained from UniProt provided protein's Cofactor annotations.

Another important component of data used to test SESAME were lists of classified proteins. Clustering algorithms based on semantic measures were applied to classified proteins and the results evaluated taking into consideration the degree of resemblance between the generated protein clusters and the original protein classes. Reactome, an open-source, open access, manually curated and peer-reviewed pathway database [71], provides the necessary tools to obtain a list of involved proteins for each metabolic pathway. For each test clustering task, a number of those lists has been used according to the chosen number of target clusters. They were extracted from Reactome in TSV formated files, each saved with a name corresponding to the respective metabolic pathway later used as the class of all the proteins in the file.

5.1.1 Gene Ontology

GO results from the compromise of GO Consortium to the goal of producing a structured, precisely defined, common, controlled vocabulary for describing the roles of genes and gene products in any organism. Their effort derives from having identified important opportunities and challenges presented by experimentally confirmed high degree of sequence and functional conservation between gene products from distinct organisms [7]:

- The main opportunity is the possibility of automated transfer of biological annotations from the experimentally tractable model organisms to the less tractable organisms based on gene and protein sequence similarity. Such information can be used to improve human health or agriculture;

- The main challenge is to meet the requirements for a largely or entirely computational system for comparing or transferring annotation among different species.

Each node in GO is linked to other kinds of information, including the many gene and protein keyword databases such as SwissPROT [70] and Gen-Bank [72]. One reason for this is that the state of biological knowledge of what genes and proteins do is very incomplete and changing rapidly. Discoveries that change the understanding of the roles of gene products in cells are published on a daily basis. To illustrate this, consider annotating two different proteins, one which knowledge about is substantial and another in which it is minimal. Being able to organize, describe, query and visualize biological knowledge at vastly different stages of completeness is mandatory. Any system must be flexible and tolerant of this constantly changing level of knowledge and allow updates on a continuing basis.

The GO Consortium found that a static hierarchical system, although computationally tractable, was also likely to be inadequate to describe the role of a gene or a protein in biology in a manner that would be either intuitive or helpful for biologists. Also, the vagueness of the term "function" when applied to genes or proteins emerged as a particular problem, as this term was colloquially used to describe biochemical activities, biological goals and cellular structure. All these reasons led to the construction of three independent ontologies.

Biological Process Refers to a biological objective to which the gene or gene product contributes. A process is accomplished via one or more ordered assemblies of molecular functions. Processes often involve a chemical or physical transformation, in the sense that something goes into a process and something different comes out of it.

Molecular Function Is defined as the biochemical activity (including specific binding to ligands or structures) of a gene product. This definition also applies to the capability that a gene product (or gene product complex) carries as a potential. It describes only what is done without specifying where or when the event actually occurs.

Cellular Component Refers to the place in the cell where a gene product is active. These terms reflect our understanding of eukaryotic cell structure. As is true for the other ontologies, not all terms are applicable to all organisms; the set of terms is meant to be inclusive. Cellular component includes such terms as "ribosome" or "proteasome", specifying where multiple gene products would be found.

GO terms are connected into nodes of a network, thus the connections between its parents and children are known and form what are technically described as directed acyclic graphs, i.e. any child term may have one or more parent terms. The ontologies are dynamic, in the sense that they exist as a network that is changed as more information accumulates, but have sufficient uniqueness and precision so that databases based on the ontologies can automatically be updated as the ontologies mature. The ontologies are flexible in another way, so that they can reflect the many differences in the biology of the diverse organisms. In this way the GO Consortium has built up a system that supports a common language with specific, agreed-on terms with definitions and supporting documentation that can be understood and used by a wide biological community. Figure 11 shows part of GO graph including protein *Q6A162* annotations [73].

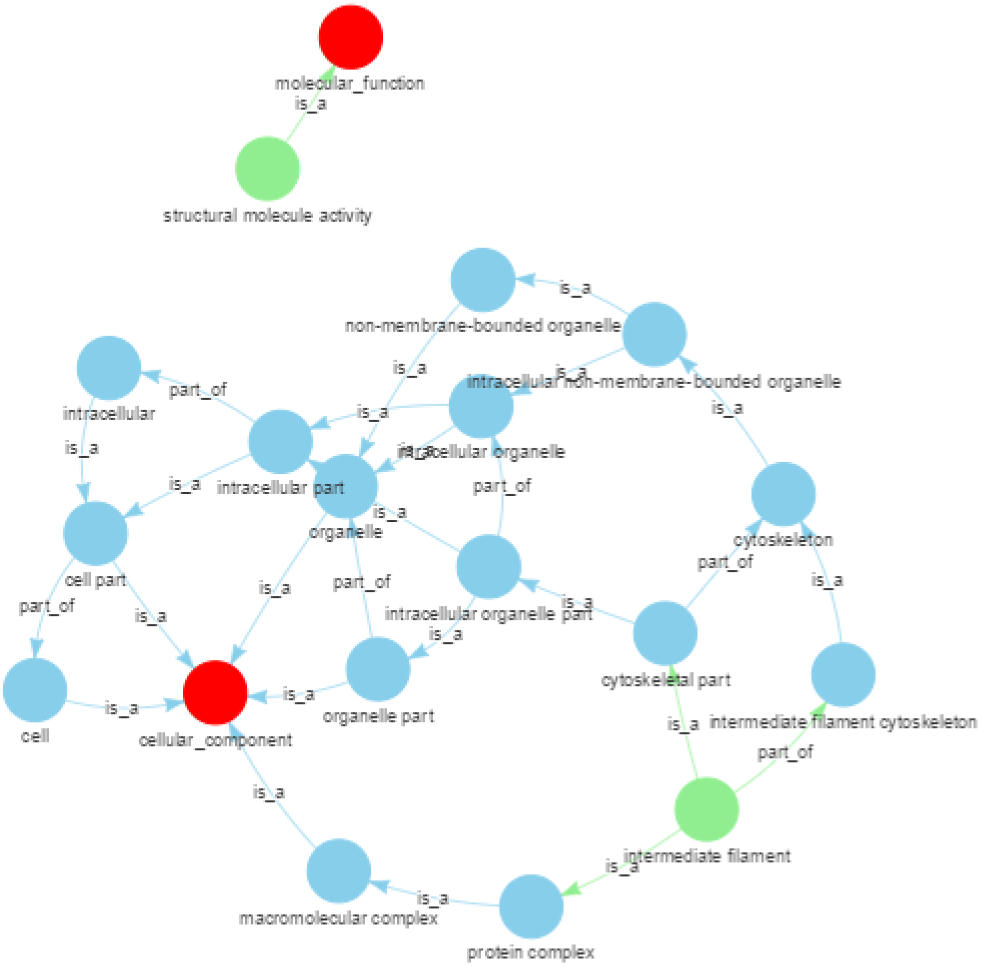


Figure 11: Part of GO graph including protein Q6A162 annotations, in green circles. Red circles refer to the root nodes of Molecular Function and Cellular Component ontologies.

5.1.2 Chemical Entities of Biological Interest Ontology

ChEBI was another bioinformatics (and biochemistry) ontology used during SESAME's testing phase.

It is an unfortunate fact that chemical data has for a long time been neglected by the computational biology/bioinformatics community. In order to address this issue, in 2002 a project was initiated at the European Bioinformatics Institute (EBI) to create a definitive, freely available dictionary of Chemical Entities of Biological Interest. The main principles involved were [8]:

- The terminology used in ChEBI should be "definitive" in the sense that it should be explicitly endorsed, where applicable, by international bodies;
- Nothing held in the database should be proprietary or derived from a proprietary source that would limit its free distribution/availability to anyone;
- Every data item in the database should be fully traceable and explicitly referenced to the original source;
- The entirety of the data should be available to all without constraint as, for example, OBO format flat files.

ChEBI ontology is one of the results of the aforementioned project, it consists of four sub-ontologies:

- Molecular Structure – in which molecular entities or parts thereof are classified according to structure;
- Biological Role – which classifies entities on the basis of their role within a biological context (e.g. antibiotic, co-enzyme, hormone);
- Application – which classifies entities, where appropriate, on the basis of their intended use by humans (e.g. pesticide, drug, fuel);
- Subatomic Particle, which classifies particles smaller than atoms.

Two of the relationships used in ChEBI ontology are the quite common *is a* (relationship between more specific and more general concepts) and *is part of* (relationship between part and whole), but others are new and specifically required by ChEBI like *is tautomer of* (cyclic relationship used to show the interrelationship between two tautomers) and *has parent hybrid* (relationship between an entity and its parent hybrid). Another significant difference from a "classic" OBO such as GO is that some of the ChEBI ontology's relationships are necessarily cyclic. The members of these cyclic relationships are placed at the same hierarchical level of the ontology. The relationships were introduced out of a need to formalize the differences between terms that are often (incorrectly) interchangeably used, especially in the biochemical literature. Figure 12 shows a fragment of ChEBII Ontology [8].

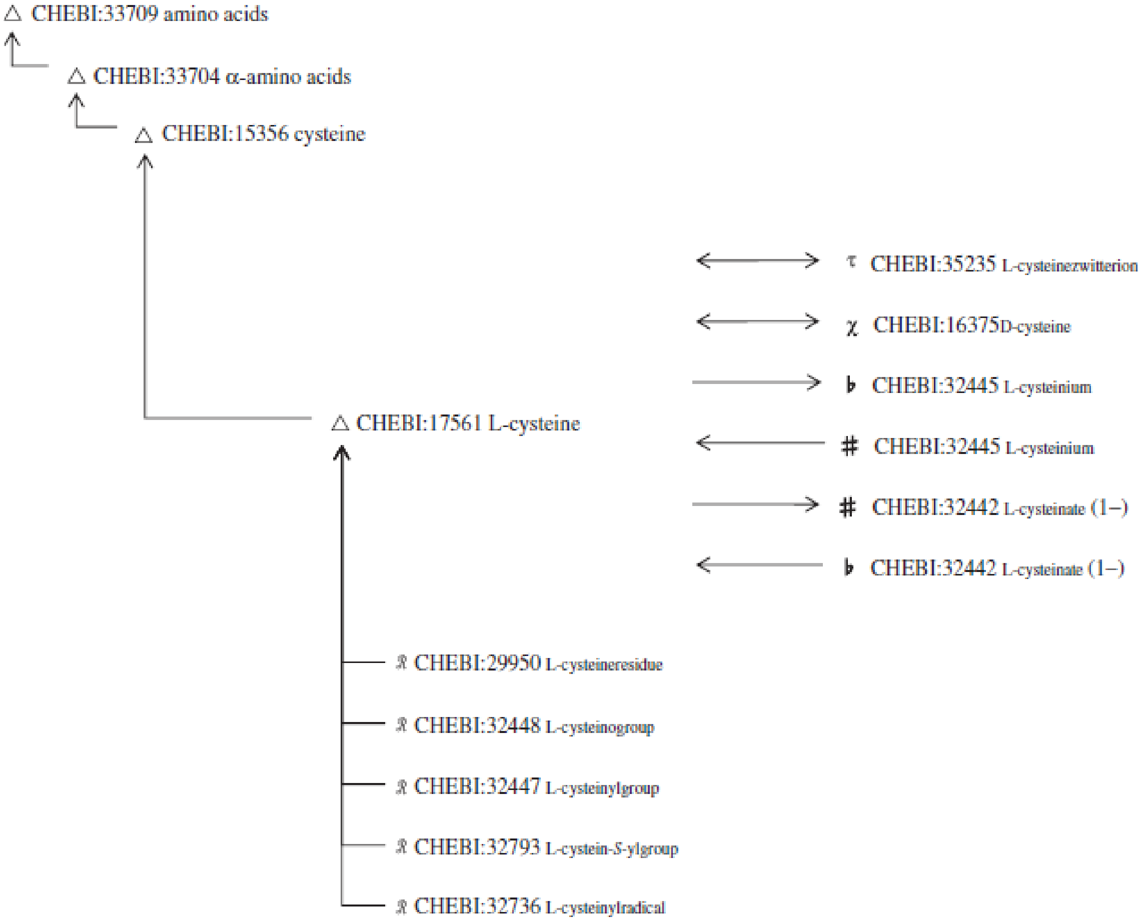

CHEBI:33709 amino acids
CHEBI:33704 α-amino acids
CHEBI:15356 cysteine
CHEBI:35235 L-cysteinezwitterion
CHEBI:16375D-cysteine
CHEBI:32445 L-cysteinium
CHEBI:17561 L-cysteine
CHEBI:32445 L-cysteinium
CHEBI:32442 L-cysteinate (1−)
CHEBI:32442 L-cysteinate (1−)
CHEBI:29950 L-cysteineresidue
CHEBI:32448 L-cysteinogroup
CHEBI:32447 L-cysteinylgroup
CHEBI:32793 L-cystein-S-ylgroup
CHEBI:32736 L-cysteinylradical

Figure 12: Fragment of ChEBI Ontology.

5.2 Baseline

In order to being able to evaluate clustering results obtained using the proposed software solution, reference results were required. The idea was to have, for the same data, a way to determine clustering performance variations caused by the introduction of semantic measures. A baseline – or reference clustering – was defined using WEKA's clustering algorithms without any kind of semantic similarity calculation. It was necessary to find an alternative to using input files with the protein instances to cluster plus files containing those instances' GO and ChEBI annotations and the ontologies themselves, to load a graph for semantic analysis and calculations. So, to capture the influence of using semantic distance versus conventionally used distances in clustering, we established a baseline where annotations were treated as words, using the *StringToWordVector* unsupervised filter provided by WEKA which converts a string attribute to a vector representing word occurrence frequencies [15].

This filter was applied to CSV data files containing the same instances but having each protein's accession number replaced by that protein annotations as shown in table 6. These CSV files must be converted to WEKA's ARFF format so they must also have a header line identifying each of the columns of data. Clustering is made based on the calculation of distances between instances which take into consideration the presence or absence of each ontology concept in each pair of compared proteins

GOChEBI_annots	Class

CHEBI:29105;GO:0005524;GO:0038095;GO:0000165;
GO:0007265;GO:0000186;GO:0007411;GO:0006464;
GO:0005829;GO:0007173;GO:0008543;GO:0045087;
GO:0008286;GO:0046872;GO:0043066;GO:0048011;
GO:0033138;GO:0004672;GO:0004674;GO:0005057;
GO:0032006;GO:0032434;GO:0007264;GO:0048010
	Axonguidance

CHEBI:29105;GO:0070062;GO:0005615;GO:0016021;
GO:0016020;GO:0004181;GO:0005634;GO:0006518;
GO:0016485;GO:0004185;GO:0005802;GO:0008270
	Membrane Trafficking

CHEBI:29108;GO:0008449;GO:0005975;GO:0070062;
GO:0006027;GO:0030203;GO:0042340;GO:0042339;
GO:0043202;GO:0046872;GO:0044281;GO:0008484
	Membrane Trafficking

Table 6: Example of a baseline input file's content.

5.3 Setup

As explained before in this chapter, the classes used in the evaluation were defined using metabolic pathways. The evaluation focused on three types of sets of human metabolic pathways:

- *No Link*, distant groups of pathways, without any link between them;
- *Link*, different groups of pathways that share a link between them;
- *Same Group*, pathways in the same group.

Table 7 presents the number of classes and proteins for each of the eight tests made for each mentioned pathways set. It also shows the pathways involved in the eight tests of each of the three pathways sets.

Set	Not linked pathways							
Test	1	2	3	4	5	6	7	8
Classes (#)	8	4	4	3	5	4	2	3
Proteins (#)	2300	671	1734	443	1669	36	1242	944
Involved Pathways	Cell Cell Communication, Cell Cycle, Cellular Responses to Stress, Chromatin Modifying Enzymes, Chromatin Organization, Circadian Clock, Detoxification of Reactive Oxygen Species, Developmental Biology, Diseases of Signal Transduction, DNA Replication, DNA Repair, Extra-cellular Matrix Organization, Metabolism of Proteins, Muscle Contraction, Organelle Bio-genesis and Maintenance, Programmed Cell Death, Synthesis of DNA							
Set	Pathways with only one link							
Test	9	10	11	12	13	14	15	16
Classes (#)	8	3	4	2	5	3	2	3
Proteins (#)	4512	1310	2495	1105	2412	30	2838	2264
Involved Pathways	Aquaporin Mediated Transport, Axon Guidance, Developmental Biology, Disease, Gene Expression, Hemostasis, Immune System, Membrane Trafficking, Neuronal System, Trans-membrane Transport of Small Molecules, Vesicle Mediated Transport							
Set	Pathways in the same group							
Test	17	18	19	20	21	22	23	24
Classes (#)	6	3	5	7	4	4	2	3
Proteins (#)	1522	600	581	1040	644	33	1400	1678
Involved Pathways	ABC Family Proteins Mediated Transport, Aquaporin Mediated Transport, Biological Oxidations, Diseases of Metabolism, Diseases of Signal Transduction, Epigenetic Regulation of Gene Expression, Generic Transcription Pathway, Infectious Disease, Ion Channel Transport, Iron Uptake and Transport, Metabolism of Amino Acids and Derivatives, Metabolism of Carbohydrates, Metabolism of Lipids and Lipoproteins, Metabolism of Nucleotides, Metabolism of Vitamins and Co-factors, Pep-tide Hormone Metabolism, Post Translational Protein Modification, Protein Repair, Signaling by EGFR, Signaling by GPCR, Signaling by NGF, Signaling by PDGF, Signaling by Retinoic Acid, Signaling by Rho GTPases, Signaling by VEGF, Signaling by Wnt, Surfactant Metabolism, SLC Mediated Trans-membrane Transport, tRNA Aminoacylation, tRNA Processing							

Table 7: Tests characteristics.

5.4 Results and Discussion

Table 8 presents a summary of the average correctly clustered instances percentage organized by pathways set groups, semantic similarity measuring configuration and clustering algorithm, when using both ontologies, GO and ChEBI. It also provides the results for the baseline.

Pathways Set	SSM	Clustering Algorithm	Avg. Corr. Clust. - SML	Avg. Corr. Clust. - Baseline
No Link	SimGIC-WAvg	FF	46.15% (± 10.12%)	42.87% (± 8.52%)
	SimGIC-HGraph		46.81% (± 11.01%)	
	Lin-WAvg		45.56% (± 9.23%)	
	Lin-HGraph		46.60% (± 10.21%)	
	SimGIC-WAvg	SKM	49.43% (± 7.62%)	43.64% (± 7.27%)
	SimGIC-HGraph		48.84% (± 8.62%)	
	Lin-WAvg		49.08% (± 8.55%)	
	Lin-HGraph		48.05% (± 8.63%)	
Link	SimGIC-WAvg	FF	52.68% (± 11.65%)	42.03% (± 10.70%)
	SimGIC-HGraph		53.17% (± 11.67%)	
	Lin-WAvg		52.56% (± 12.84%)	
	Lin-HGraph		52.21% (± 13.06%)	
	SimGIC-WAvg	SKM	47.38% (± 12.60%)	42.47% (± 10.19%)
	SimGIC-HGraph		47.08% (± 12.27%)	
	Lin-WAvg		47.47% (± 12.29%)	
	Lin-HGraph		48.74% (± 12.87%)	
Same Group	SimGIC-WAvg	FF	50.73% (± 12.60%)	52.83% (± 17.27%)
	SimGIC-HGraph		51.96% (± 11.82%)	
	Lin-WAvg		48.98% (± 11.71%)	
	Lin-HGraph		49.97% (± 11.42%)	
	SimGIC-WAvg	SKM	48.16% (± 18.49%)	51.53% (± 15.29%)
	SimGIC-HGraph		48.90% (± 18.75%)	
	Lin-WAvg		51.52% (± 17.07%)	
	Lin-HGraph		52.23% (± 16.63%)	

Table 8: Overview of clustering results using the two ontologies (with standard deviation). SSM (Semantic Similarity Measure); WAvg (Weighted Average); HGraph (Hyper-graph); FF (Farthest First); SKM (SimplekMeans).

Several interesting facts can be observed. Regarding the type of pathways set, the use of semantic similarity is clearly beneficial for both the *No Link* and *Link* sets, with increases in performance ranging from +3% (No Link-Lin-WAvg-FF) to +11% (Link-SimGIC-HGraph-FF). In the *Same Group* set, the baseline performs on average better than the semantic similarity based approaches, with differences in performance ranging from -4%(Lin-WAvg-FF) to +1%(Lin-HGraph-SKM). Looking at the baseline alone, the easiest pathway set to cluster is the *Same Group*. This is probably due to the fact that more similar pathways have a higher probability of sharing some of the same annotations, making their similarity easier to determine using non-semantic approaches. However, this observation is not as straightforward when using semantic similarity, since the best results were found for the *Link* set when using the Farthest First algorithm. One possible explanation is that using semantic similarity improves the ability to detect similarities between more distantly related proteins, making them easier to cluster, but for more closely related proteins, the discriminating power of semantic similarity is lower, hindering clustering.

However, in all cases the standard deviation is high, (7-19%), stressing the sensitivity of the method to the characteristics of each clustering task. For instance, the target number of clusters has a clear impact on results (see figure 13). A clear tendency to a decrease in clustering correction for an increased number of target clusters is visible, regardless of the employed approach. There is however an outlier with seven clusters. This is probably caused by the fact that only one out of the total of twenty four tests had seven target clusters (as shown in table 7), and also belonged to the *Same Group* set, which skewed the results. The target number of clusters also appears to have a stronger impact on the approaches using semantic similarity, with smaller clusters numbers corresponding to better performances than the baseline, while larger (above 5), correspond to poorer performance than the baseline in most cases.

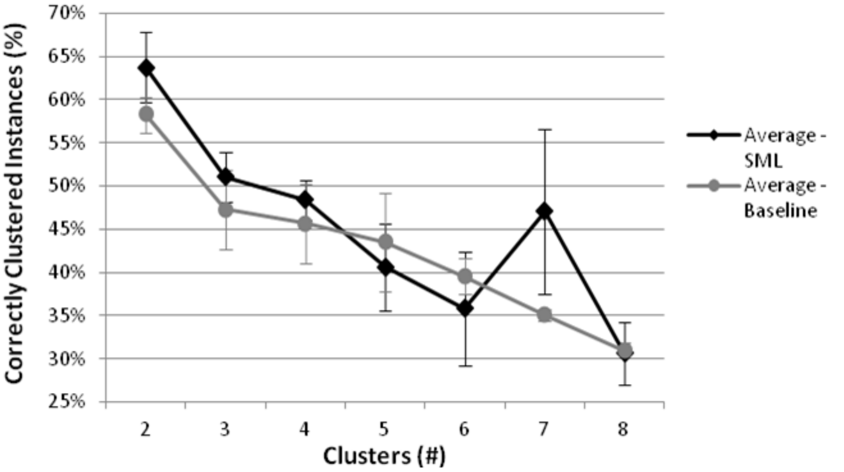


Figure 13: Correctly clustered instances per number of clusters.

On the other hand, the number of instances to cluster by itself doesn't seem to have a direct relation with clustering results, either using semantic similarity or not. Figure 14 illustrates the variation of average percentage of correctly clustered instances (including standard deviation) as a function of the number of proteins used in each of the twenty four clustering tests. Results are shown for both semantic similarity and baseline approaches including (just for the former since it is the same for the latter), for each point a data label with the corresponding number of target clusters. In this figure, it is not possible to identify a tendency relating the variation of the number of proteins with the variation of clustering results. However, it is unquestionable that the worst average percentages of correctly clustered instances obtained using semantic similarity occur for tests where a high number of proteins to cluster is combined with the highest value of tested number of target clusters, eight:

- Test with 2300 proteins to eight target clusters achieves a average percentage of correctly clustered instances of just 31,92% (with 7,46% standard deviation);
- Test with 4512 proteins to eight target clusters achieves a average percentage of correctly clustered instances of just 29,28% (with 2,35% standard deviation).

This observation suggests that future efforts to test and/or improve clustering with semantic similarity based on multiple ontologies must always pay special attention to results concerning high number of instances combined with high target number of clusters.

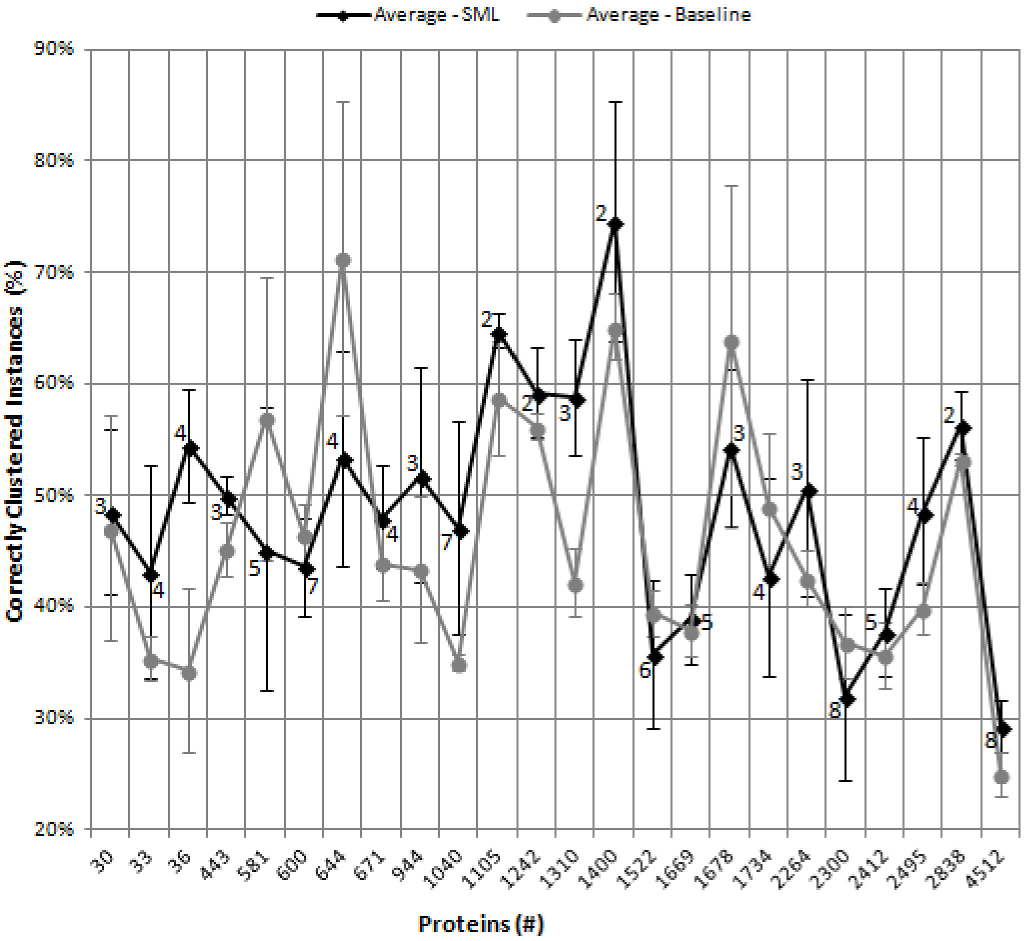


Figure 14: Correctly clustered instances per number of proteins.

Figure 15 illustrates the impact of using each ontology separately and their two modes of combination. It provides an overview of results for the three pathways set types, using the direct groupwise semantic measure SimGIC. Using both ontologies or just GO results reveal very similar performance for the same clustering and semantic similarity approaches, whereas this is not the case when using just ChEBI. A small proportion of proteins in the data-sets are annotated to ChEBI (roughly just 5 to 10%). This means that for these tests, the number of proteins in each task is smaller. It is interesting to note however, that while for the *Same Group* and *Link* sets, using only ChEBI results in a performance equivalent or somewhat lower than the baseline, in the *No Link* set, when using the Farthest First clustering algorithm, using semantic similarity improves performance by 3%. This is probably due to the fact that for more closely related proteins (with more similar annotations) semantic similarity using ChEBI is not able to provide extra information on the similarity of the proteins, both because:

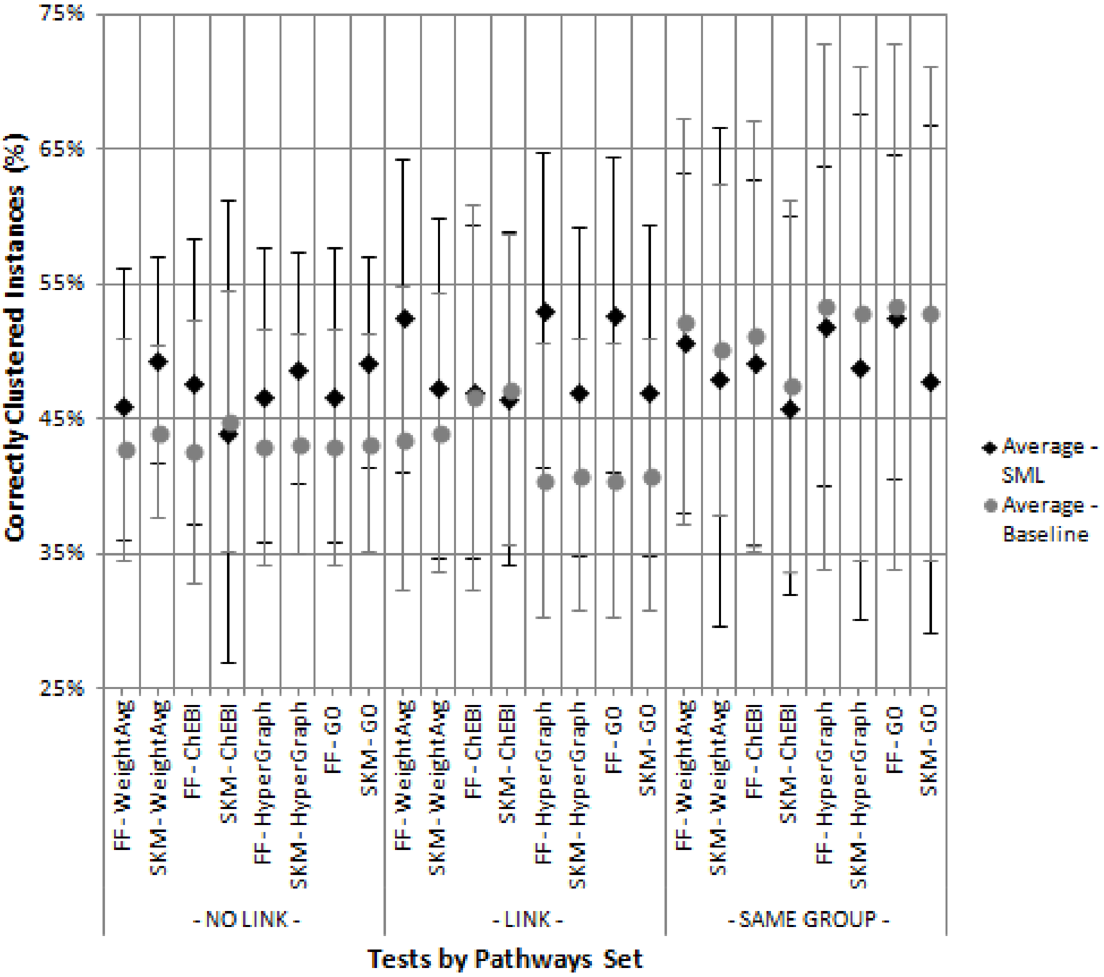


Figure 15: Correctly clustered instances per pathways set tests.

1. ChEBI has a very large proportion of leaf nodes – i.e., without descendants – (51493 leaves out of 65413 concepts to be exact) which makes their semantic information contribution poorer;

2. The number of annotations to ChEBI is significantly lower than to GO.

When evaluating a software solution, the time taken to run each task is a valuable indicator. In the case of SESAME, the clustering time was recorded for each of the total three hundred and eighty four test tasks (sixteen different clusterer plus semantic measuring configuration combinations for each of the twenty four tests). These recorded time values provided the data for figure 16 where average clustering time (in seconds) is shown for the four possible clusterer plus semantic measure configurations separated by different tested numbers of target clusters. Here, the different used options for graph loading were not distinguished since the two semantic measures used are always combined in the tests with each and every one of those options. This way it is possible to determine if:

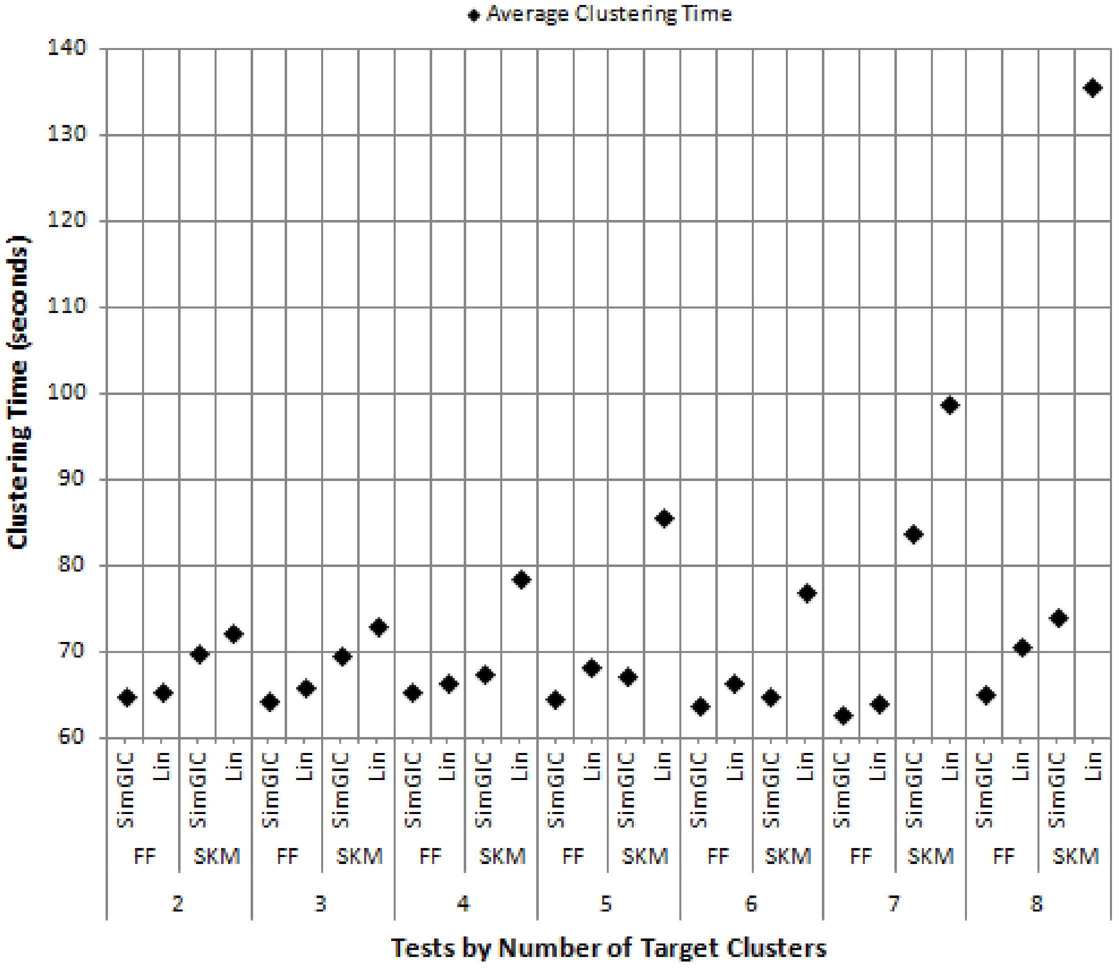


Figure 16: Clustering time per clusterer plus semantic measure configurations separated by tested numbers of target clusters.

- Is there a faster clusterer?
- Is there a faster semantic measure?
- Is there a faster clusterer plus semantic measure combination?

- The number of clusters affects clustering time?

An immediate consequence of average clustering times including all the possible graph loading options is very high standard deviation values. Tasks using graphs exclusively loaded with ChEBI ontology always involve much inferior number of proteins and annotations than those using graphs loaded with GO or a combination of both the ontologies. For this reason, these former tasks obviously always take much less clustering time causing the very high standard deviation values which, for practical reasons, could not be included in figure 16.

In the figure, SimGIC stands for the direct groupwise semantic measure with that name using ontology based IC computation and Lin stands for indirect groupwise measure based on Lin's pairwise measure with Best Match Average aggregation strategy also using ontology-based IC computation.

Analyzing the figure it is possible to conclude that:

- In the same testing conditions, Farthest First is always faster than SimplekMeans being that difference strengthened as the number of target clusters increases, particularly in the case of the combination of SimplekMeans with Lin's based groupwise semantic measure;

- In the same testing conditions, SimGIC groupwise semantic measure is always faster than Lin's based groupwise semantic measure;

- For the same number of target clusters, the combination of Farthest First with SimGIC is always the fastest;

- Only for the combination of SimplekMeans with Lin there is a clear tendency for clustering time increase related to target number of clusters increase (outlier for six target clusters), the other three combinations' clustering times are independent of the target number of clusters even preserving interesting stable results between around sixty five and seventy five seconds (outlier for seven target clusters);

- The clustering time for the combination of SimplekMeans with Lin, beyond target number of clusters, taking into consideration the disparity of the result obtained for eight target clusters, is probably also severely affected by the number of proteins to cluster since, as shown in table 7, for eight target clusters there are two tests both with a very high number of proteins to cluster (2300 and 4512).

A question arises: is there a relation between speed and correction in clustering results using the identified faster options? By analyzing table 8, it becomes clear that there is not such a relation due to the introduction of the pathways sets as a conditioning factor. Pathways set *Link* reveals better average correctly clustering results using semantic similarity for Farthest First and also for the combination of Farthest First with SimGIC, therefore establishing a relation with the aforementioned conclusions about rapidity. However, this relation is contradicted by the results observable, for instance, for pathways set *No Link*.

5.4.1 Annotations Completeness

For all the input data files used to test SESAME the number of annotations to GO and ChEBI of every protein was counted with the goal to try to evaluate its importance to achieve successful clustering results. Since average numbers of protein annotations were calculated for each test, attention must be paid – particularly in the case of GO which has incomparably higher numbers of annotations – to the fact that high numbers of proteins, with quite different numbers of annotations each, are involved implying very high standard deviation values.

Exclusively for clustering tasks with semantic similarity based on GO, figure 17 illustrates the relation between each of the twenty four tests' (eight for each of the three used pathways sets) average percentage of correctly clustered instances and average number of protein annotations to GO.

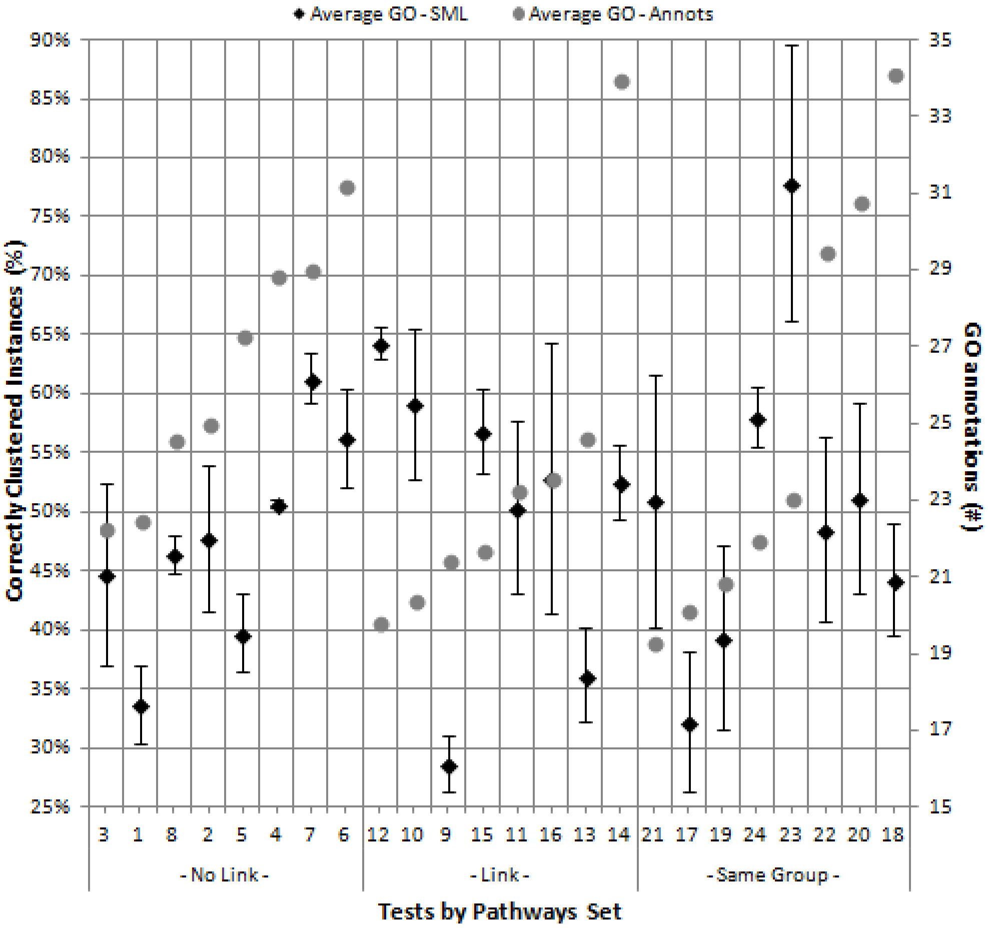


Figure 17: Average percentage of correctly clustered instances and average number of protein annotations to GO per tests (numbered according to table 7) by pathways set.

For a clearer analysis the eight tests of each pathways set were sorted by the number of annotations to GO. The figure allows to conclude that, generally, improved clustering results can not be associated to increased number of annotations to GO. There is only a, somehow, close relation for the *No Link* pathways set from test one to seven with an exception for test five, no relation at all for the *No Link* pathways set and again a partial relation from test seventeen to twenty three of the *Same Group* pathways set.

Figure 18 illustrates the relation between each of the twenty four tests' average percentage of correctly clustered instances and average number of protein annotations to ChEBI, exclusively for clustering tasks with semantic similarity based on ChEBI. Once again, for clarity purposes, the eight tests of each pathways set were sorted by the number of annotations to ChEBI.

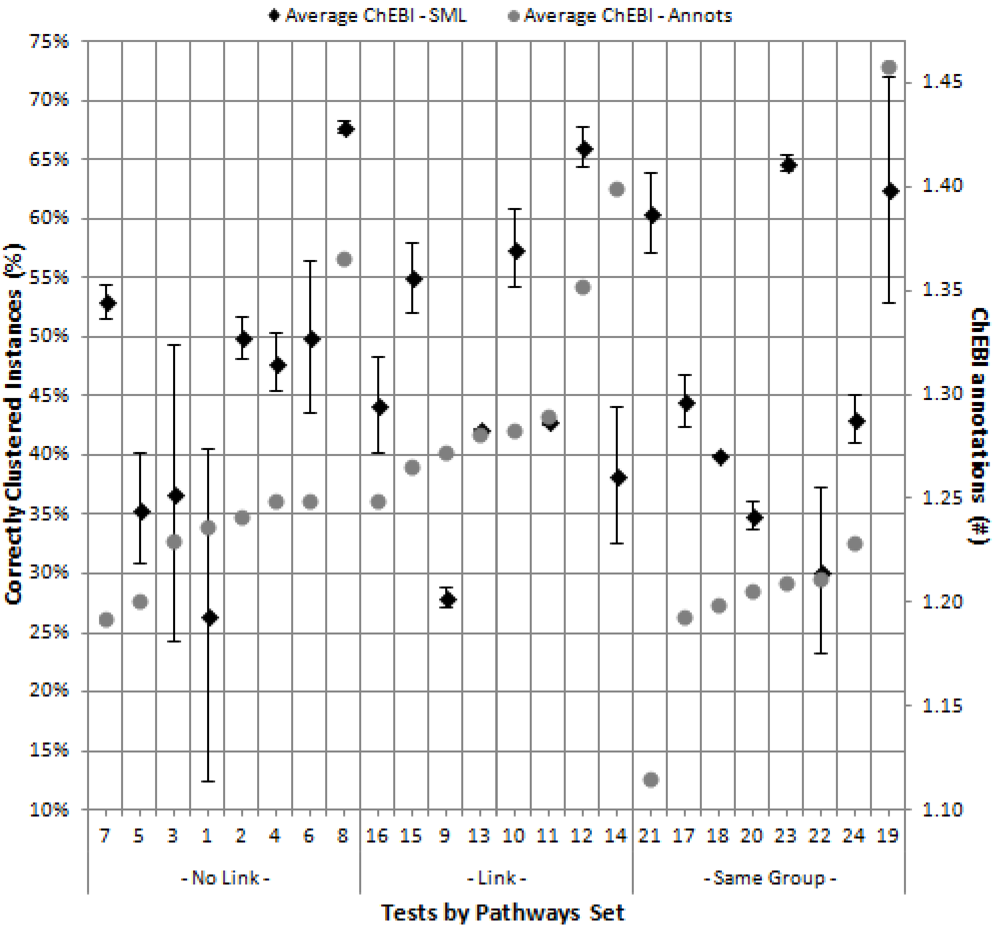


Figure 18: Average percentage of correctly clustered instances and average number of protein annotations to ChEBI per tests (numbered according to table 7) by pathways set.

The global variation of the average number of protein annotations to ChEBI is just forty four tenths. Still, the figure allows to conclude that higher average number of annotations to ChEBI are not a contribute to better average clustering results. Only for the three last tests of the *No Link* pathways set a slight correspondence can be identified.

A straightforward calculation of *Pearson Product-Moment Correlation Coefficient* gives a too low value of 0.28 between the average number of protein annotations to ChEBI and the average percentage of correctly clustered instances for clustering tasks with semantic similarity based on ChEBI and an even lower value of 0.06 between the average number of protein annotations to GO and the average percentage of correctly clustered instances for clustering tasks with semantic similarity based on GO thus preventing any conclusive observation.

To complement this annotations completeness evaluation the present work's multiple ontologies options should also be analyzed but, due to both the much reduced number of proteins annotated to ChEBI than to GO and the much reduced number of ChEBI than GO annotations per annotated protein (see figure 19):

- Hyper-graph and Weighted Average approaches' average percentage of correctly clustered instances values and average numbers of protein annotations to GO and ChEBI are all identical to those obtained with the single ontology approach using GO;
- The test's average numbers of protein annotations to ChEBI is almost irrelevant in comparison with those to GO.

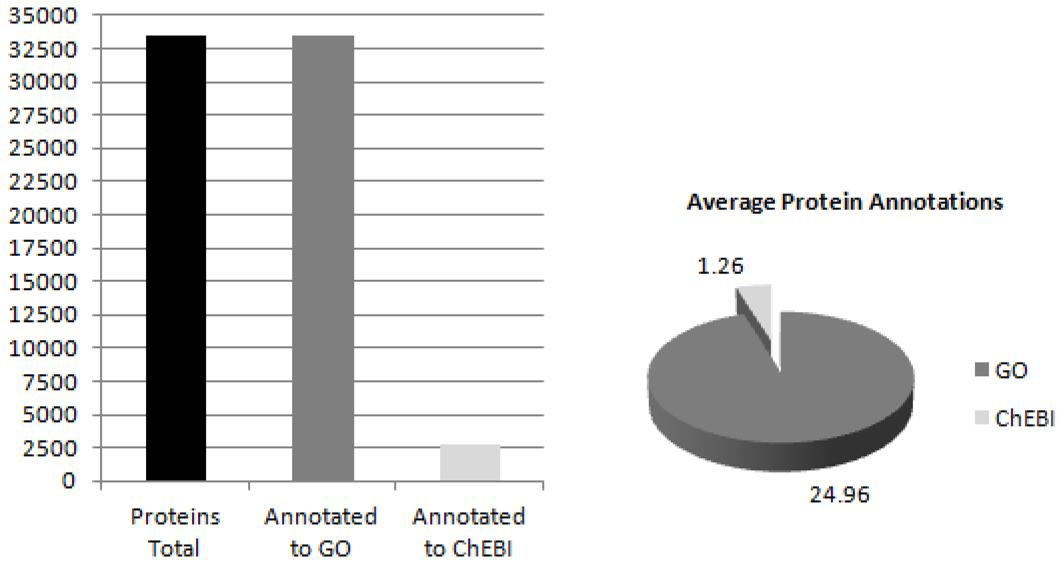


Figure 19: GO and ChEBI contributions.

Therefore, the results obtained for single ontology approach using GO are sufficiently representative of those obtained for the two multiple ontologies approaches using combinations of GO and ChEBI.

The number of annotations of an entity with concepts from a certain ontology is a broad measure of how complete is the information about that entity in that ontology.

It is then advised to pose the hypothesis that if the information about two entities in an ontology is quite substantial, the semantic similarity between those two entities calculated based on that ontology should be more accurate than that calculated between two scarcely annotated entities. The high variability in annotation number certainly plays a role in the results observed, which do not sustain this hypothesis. However, a positive relation can be seen for the *No Link* group, pointing that for more diverse entities, annotation completeness can have a positive impact on the performance of semantic distance based clustering.

5.4.2 Case Study

Two of the twenty four tests stood out for their substantially (over 15%) different average percentages of correctly clustered instances despite the fact of having similar number of proteins and the same number of target clusters. Table 9 summarizes these tests' characteristics including the average percentage of correctly clustered instances using SML and the average number of proteins annotations to GO and ChEBI.

Test	Proteins (#)	Target Clusters (#)	Pathways Set	Avg. GO Proteins Annots. (#)	Avg. ChEBI Proteins Annots. (#)	Avg. Correctly Clustered - SML (%)
23	1400	2	Same Group	23.04	1.21	74.60% (± 10.76%)
7	1242	2	No Link	28.99	1.19	59.22% (± 4.11%)

Table 9: Characteristics of two tests with substantially different average percentage of correctly clustered instances results for similar number of proteins and the same number of target clusters.

Figure 20 shows the percentage of correctly clustered instances, using semantic similarity and the baseline, for all of these two tests' tasks. For each of the two used clusterers, Farthest First and SimplekMeans, the two multiple ontologies approaches, Weighted Average and Hyper-graph, and the two single ontology approaches, GO and ChEBI, combined with both SimGic and Lin groupwise semantic measures.

First of all, the figure makes the advantage of using semantic similarity clear in both tests. The majority of the results of the sixteen clustering tasks for each test are better when using semantic similarity. More, the number of tasks with better results when using semantic similarity is higher in the case of the test with metabolic pathways from the *No Link* pathways set. This points in the expected direction that more closely related proteins (with more similar annotations), like those involved in metabolic pathways from the *Same Group* pathways set, are more likely to cause higher number of unsuccessful clustering tasks when using semantic similarity. In the case of test 7, *No Link* pathways set, only clustering tests based on ChEBI ontology contradict the general best clustering

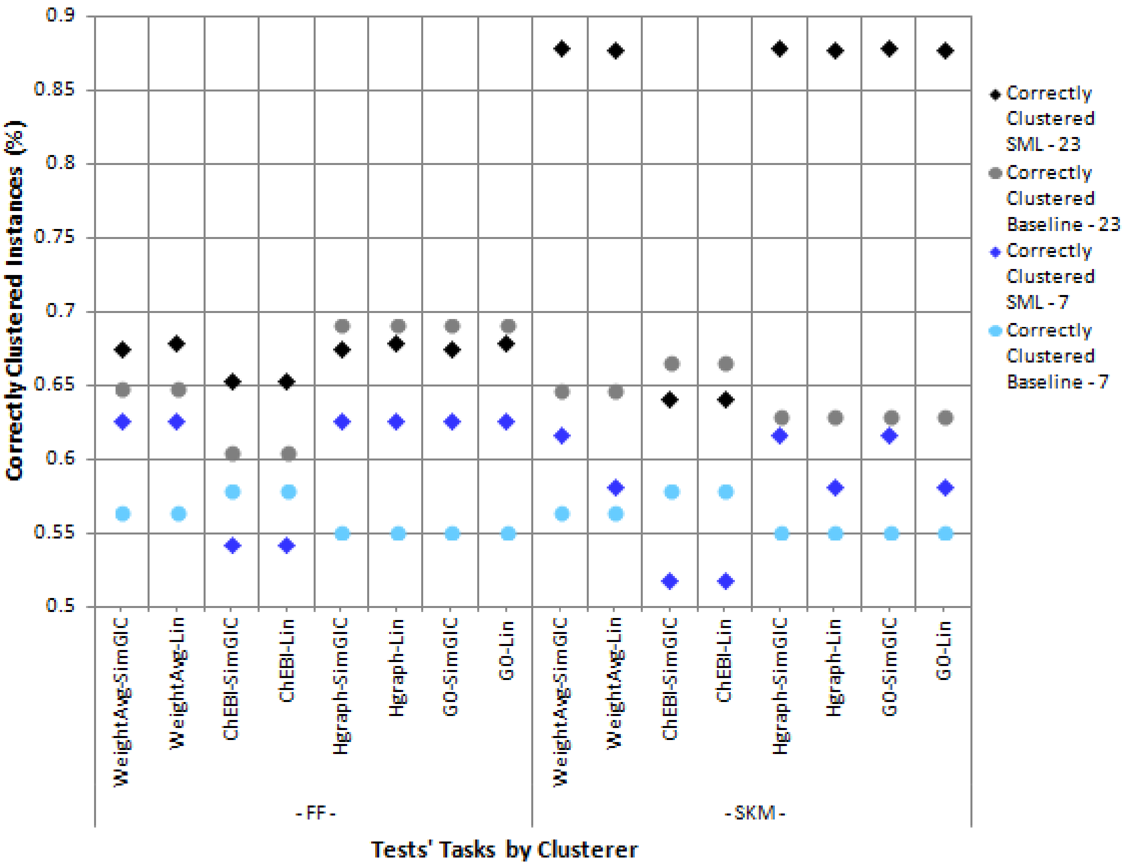


Figure 20: Percentage of correctly clustered instances for all the tasks of the two SESAME's tests characterized in table 9.

results when using semantic similarity which is understandable taking in consideration the already mentioned characteristics of that ontology. This is not the case for test 23, *Same Group* pathways set, with two clustering tasks based on multiple ontologies and two other based only on GO, although close, achieving better results for baseline.

That said, the reason why this two tests stood out, i.e. test's average percentages of correctly clustered instances using semantic similarity with over 15% difference despite the fact of having similar number of proteins and the same number of target clusters, is also relevant. Relevance reinforced by two facts:

- The test with better average percentage of correctly clustered instances using semantic similarity is the one with metabolic pathways from the *Same Group* pathways set;
- The test with worst average percentage of correctly clustered instances using semantic similarity is the one with higher average number of proteins annotated to GO.

All in all, the usefulness of semantic similarity as a contribute to better clustering results proves generally positive. As with any clustering work, the specificities of the data can always impact the performance, which is the case for the two tests of this case study. Nevertheless, it reveals the importance of, beyond average global values, whenever possible, analyze the parts that produce that average. For instance, clustering task's data instances may have a very high average number of annotations to an ontology, but if this is mainly caused by an exceptionally high number of annotations of just a few instances this will probably not be a great contribution to achieve better clustering results.

Chapter 6

Conclusions

The purpose of developing a software solution for clustering based on semantic similarity as an extension of an extremely popular machine learning platform like WEKA reflects the identified high potential of such solution's algorithms and methods to adequately integrate with others of renowned efficiency. WEKA's self proclaimed [66] plugin nature with very easy ways to extend its existing algorithms through automatic discovery of classes on the *classpath* has been fully confirmed in the present work. Hopefully, WEKA's popularity will constitute an additional pathway to promote the dissemination of the present work's results thus contributing to a much welcome feedback regarded as a valuable help to future improvements.

SML, which combines integrability characteristics with the needed software tools for semantic measures implementation, proved to be a good choice concerning semantic similarity. The availability in the library of diversified semantic measures based on graph analysis options, has been of utmost importance for achieving the goal of using ontologies and annotations as semantic support resources. Several alternative accepted formats demanded, on one hand, a time intensive thorough study but eventually allowed, on the other, to reach an effective compromise in order to implement the intended strategies for semantic space exploration.

Rooted in these two reference software libraries' integration, single and multiple ontologies based semantic similarity computation approaches were implemented as planned further enriching the expectations to present a solid first contribution towards making semantics-based clustering more accessible to the community, particularly through this work's contribution of releasing the implemented software on *github*. In fact, the integration via SML of semantic similarity measures into WEKA and the extension of SML to handle multiple ontologies are other two of the present work's contributions and meet its goals of developing new clustering strategies based on the exploration of the semantic space making use of semantic similarity measures and implementing those strategies in the WEKA 3 library.

The integration was subject to a preliminary evaluation for clustering proteins according to their GO and ChEBI annotations, which was shown to improve the performance comparatively to non-semantic approaches, particularly for datasets where there was more semantic diversity. Furthermore, it was shown that the usefulness of employing semantic similarity depends not only on the diversity of the datasets, but also on the structure of the ontologies employed, and the degree to which they are able to impart useful information to identify similar instances. The known tendency to reduced clustering performance related to increased number of target clusters was confirmed particularly in those cases in which high number of target clusters is concomitant with high number of instances to cluster. As expected, this concomitance also proved slow obtaining the higher clustering times. Providing these preliminary test's results is another of this work's contributions and meets its goal of assessment of the newly developed clustering strategies using real data.

6.1 Future Work

Although the present work provides a first step, future endeavors need to be undertaken in several fronts:

- Further evaluate this work's approaches capabilities by making additional tests with different combinations and number of used ontologies and different data instances' classes;
- Explore alternative clustering algorithms (e.g., hierarchical or spectral clustering) including alternative centroid initialization methods;
- Consider other more complex semantic similarity measures;
- Examine the impact of annotation quality;
- Investigate computational efficiency issues in using multiple ontologies.

The latter question is of particular relevance for the hyper-graph approach, since all ontologies need to be in memory to support SML computations. This can represent a serious challenge in the biomedical domain where many ontologies can be used to describe the data, and the ontologies themselves can be quite large (with hundreds of thousands of concepts).

Ultimately, the proposed approach can be used to analyse diverse datasets composed of both semantic annotations and numerical values, by combining it with the conventional approaches already available in WEKA.

Acronyms

ACA all common ancestors 12

ARFF Attribute-Relation File Format 20, 23, 39

ChEBI Chemical Entities of Biological Interest 2, 31, 33, 34, 37, 38, 41, 44, 46–50

CSV Comma-Separated Values 19, 20, 23, 39

EBI European Bioinformatics Institute 37

GAF GO Annotation File Format 30

GO Gene Ontology 2, 15, 31, 33–38, 41, 43, 44, 46–51

GPC *GenericPropertiesCreator.props* 22, 23

GUI Graphical User Interface 20–23

IC information content 12, 14, 25, 26, 46

IDE Integrated Development Environment 18, 21

IRI Internationalized Resource Identifier 11, 12

JAR Java Archive 18, 22

KDD Knowledge Discovery from Data 5, 6

KNIME Konstanz Information Miner 11

MICA maximum informative common ancestor 12

OBO Open Biomedical Ontologies 15, 20, 30, 33, 37

OWL Web Ontology Language 15, 30

RDF Resource Description Framework 15, 30

SML Semantic Measures Library 15, 18, 22, 25, 30, 31, 33, 34, 50, 53

TSV Tab-Separated Values 20, 21, 26, 30, 34

UniProt Universal Protein Resource 33

UniProtKB UniProt Knowledgebase 33

URI Uniform Resource Identifier 20, 26

VSM vector space models 12

WEKA Waikato Environment for Knowledge Analysis 10, 11, 18, 20–25, 33, 38, 39, 53

Bibliography

[1] Ian Horrocks. Ontologies and the semantic web. *Communications of the ACM*, 51(12):58, 2008.

[2] Brijendra Singh and Hemant Kumar Singh. Web Data Mining research: A survey. *IEEE International Conference on Computational Intelligence and Computing Research*, pages 1–10, 2010.

[3] Jason Bell. *Machine Learning, Hands-On for Developers and Technical Professionals*, volume 53. Wiley, 2013.

[4] Jürgen Schmidhuber. Deep Learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[5] Nada Lavrač, Anže Vavpetič, Larisa Soldatova, Igor Trajkovski, and Petra Kralj Novak. Using ontologies in semantic data mining with SEGS and g-SEGS. *Proceedings of the 14th international conference on Discovery science (DS'11)*, pages 165–178, 2011.

[6] Luc. de Raedt. *Logical and Relational Learning: From ILP to MRDM*. 2008.

[7] The Gene Ontology Consortium. Gene ontology: Tool for the identification of biology. *Natural Genetics*, 25(may):25–29, 2000.

[8] Kirill Degtyarenko, Paula De matos, Marcus Ennis, Janna Hastings, Martin Zbinden, Alan Mcnaught, Rafael Alcántara, Michael Darsow, Mickaël Guedj, and Michael Ashburner. ChEBI: A database and ontology for chemical entities of biological interest. *Nucleic Acids Research*, 36(SUPPL. 1):344–350, 2008.

[9] Peter N. Robinson, Sebastian Köhler, Sebastian Bauer, Dominik Seelow, Denise Horn, and Stefan Mundlos. The Human Phenotype Ontology: A Tool for Annotating and Analyzing Human Hereditary Disease. *American Journal of Human Genetics*, pages 610–615, 2008.

[10] David Sánchez, Montserrat Batet, David Isern, and Aida Valls. Ontology-based semantic similarity: A new feature-based approach. *Expert Systems with Applications*, 39(9):7718–7728, 2012.

[11] João Diogo Silva Ferreira. *Semantic Similarity Across Biomedical Ontologies*. PhD thesis, University of Lisbon, 2016.

[12] J. Han and M. Kamber. *Data Mining: Concepts and Techniques (3rd ed)*. 2012.

[13] Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[14] Tom M Mitchell. The Discipline of Machine Learning. *Machine Learning*, 17(July):1–7, 2006.

[15] Ian H. Witten, Eibe Frank, and Mark a. Hall. *Data Mining*, volume 277. 2011.

[16] A. Jovic, K. Brkic, and N. Bogunovic. An overview of free software tools for general data mining. *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2014 - Proceedings*, pages 1112–1117, 2014.

[17] Remco R. Bouckaert, Eibe Frank, Mark a. Hall, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. WEKA—Experiences with a Java Open-Source Project. *The Journal of Machine Learning Research*, 11:2533–2541, 2010.

[18] Brett Lantz. *Machine Learning with R*. 2013.

[19] KNIME. https://www.knime.org/.

[20] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. ... *of Machine Learning* ..., 12:2825–2830, 2012.

[21] Gayo Diallo, Michel Simonet, and Ana Simonet. An approach to automatic ontology-based annotation of biomedical texts. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4031 LNAI:1024–1033, 2006.

[22] Peter Geibel, Martin Trautwein, Hebun Erdur, Lothar Zimmermann, Stefan Krüger, Josef Schepers, Kati Jegzentis, Frank Müller, Christian Hans Nolte, Anne Becker, Markus Frick, Jochen Setz, Jan Friedrich Scheitz, Serdar Tütüncü, Tatiana Usnich, Alfred Holzgreve, Thorsten Schaaf, and Thomas Tolxdorff. *On the Move to Meaningful Internet Systems: OTM 2013 Conferences*, volume 8185. 2013.

[23] Alexander Maedche and Valentin Zacharias. Clustering Ontology-Based Metadata in the Semantic Web. *Principles of Data Mining and Knowledge Discovery*, 2431:383–408, 2002.

[24] Catia Luisa Santana Calisto Pesquita. *Automated Extension of Biomedical Ontologies*. PhD thesis, 2012.

[25] Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. Semantic Measures for the Comparison of Units of Language, Concepts or Entities from Text and Knowledge Base Analysis. *arXiv preprint arXiv: ...*, pages 1–102, 2013.

[26] Waqar Ali and Charlotte M. Deane. Functionally guided alignment of protein interaction networks for module detection. *Bioinformatics*, 25(23):3166–3173, 2009.

[27] Young-Rae Cho, Woochang Hwang, Murali Ramanathan, and Aidong Zhang. Semantic integration to identify overlapping functional modules in protein interaction networks. *BMC bioinformatics*, 8:265, 2007.

[28] Mihail Popescu, James M Keller, and Joyce a Mitchell. Fuzzy measures on the Gene Ontology for gene product similarity. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 3(3):263–74, 2006.

[29] David Martin, David Martin, Christine Brun, Christine Brun, Elisabeth Remy, Elisabeth Remy, Pierre Mouren, Pierre Mouren, Denis Thieffry, Denis Thieffry, Bernard Jacq, and Bernard Jacq. GOToolBox: functional analysis of gene datasets based on Gene Ontology. *Genome biology*, 5(12):R101, 2004.

[30] Hongwei Wu, Zhengchang Su, Fenglou Mao, Victor Olman, and Ying Xu. Prediction of functional modules based on comparative genome analysis and Gene Ontology application. *Nucleic Acids Research*, 33(9):2822–2837, 2005.

[31] Sidahmed Benabderrahmane, Malika Smail-Tabbone, Olivier Poch, Amedeo Napoli, and Marie-Dominique Devignes. IntelliGO: a new vector-based semantic similarity measure including annotation origin. *BMC Bioinformatics*, 11(1):588, 2010.

[32] James Z Wang, Zhidian Du, Rapeeporn Payattakool, Philip S Yu, and Chin-Fu Chen. A new method to measure the semantic similarity of GO terms. *Bioinformatics (Oxford, England)*, 23(10):1274–81, 5 2007.

[33] Meeta Mistry and Paul Pavlidis. Gene Ontology term overlap as a measure of gene functional similarity. *BMC bioinformatics*, 9(1):327, 2008.

[34] Hisham Al-Mubaid and Anurag Nagar. Comparison of four similarity measures based on GO annotations for Gene Clustering. In *2008 IEEE Symposium on Computers and Communications*, pages 531–536. IEEE, 7 2008.

[35] Catia Pesquita, Daniel Faria, Hugo Bastos, António E N Ferreira, André O Falcão, and Francisco M Couto. Metrics for GO based protein semantic similarity: a systematic evaluation. *BMC bioinformatics*, 9 Suppl 5:S4, 2008.

[36] R Gentleman. Visualizing and distances using GO. *URL http://www. bioconductor. org/docs/vignettes. ...*, pages 1–5, 2005.

[37] Ping Ye, Brian D Peyser, Xuewen Pan, Jef D Boeke, Forrest a Spencer, and Joel S Bader. Gene function prediction from congruent synthetic lethal interactions in yeast. *Molecular systems biology*, 1:2005.0026, 2005.

[38] Brendan Sheehan, Aaron Quigley, Benoit Gaudin, and Simon Dobson. A relation based measure of semantic similarity for Gene Ontology annotations. *BMC bioinformatics*, 9:468, 2008.

[39] Homin K Lee, Amy K Hsu, Jon Sajdak, Jie Qin, and Paul Pavlidis. Coexpression Analysis of Human Genes Across Many Microarray Data Sets. pages 1085–1094, 2004.

[40] Haiyuan Yu, Ronald Jansen, and Mark Gerstein. Developing a similarity measure in biological function space. *Bioinformatics*, 2007.

[41] Julie Chabalier, Jean Mosser, and Anita Burgun. A transversal approach to predict gene product networks from ontology-based similarity. *BMC bioinformatics*, 8(1):235, 2007.

[42] Olivier Bodenreider, Marc Aubry, and Anita Burgun. Non-lexical approaches to identifying associative relations in the gene ontology. *Pacific Symposium on Biocomputing.*, pages 91–102, 1 2005.

[43] Francisco M. Couto, Mário J. Silva, and Pedro M. Coutinho. Measuring semantic similarity between Gene Ontology terms. *Data & Knowledge Engineering*, 61(1):137–152, 4 2007.

[44] Jay J. Jiang and David W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. page 15, 9 1997.

[45] Dekang Lin. An Information-Theoretic Definition of Similarity. *Proceedings of ICML*, pages 296–304, 1998.

[46] Razib M Othman, Safaai Deris, and Rosli M Illias. A genetic similarity algorithm for searching the Gene Ontology terms and annotating anonymous protein sequences. *Journal of biomedical informatics*, 41(1):65–81, 2 2008.

[47] Viktor Pekar and Steffen Staab. Taxonomy learning. In *Proceedings of the 19th international conference on Computational linguistics -*, volume 1, pages 1–7, Morristown, NJ, USA, 8 2002. Association for Computational Linguistics.

[48] Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. page 6, 11 1995.

[49] Xiaomei Wu, Lei Zhu, Jie Guo, Da-Yong Zhang, and Kui Lin. Prediction of yeast protein-protein interaction network: insights from the Gene Ontology and annotations. *Nucleic acids research*, 34(7):2137–50, 2006.

[50] Hui Yu, Lei Gao, Kang Tu, and Zheng Guo. Broadly predicting specific gene functions with expression similarity and taxonomy similarity. *Gene*, 352:75–81, 6 2005.

[51] Bo Li, James Z. Wang, F. Alex Feltus, Jizhong Zhou, and Feng Luo. Effectively integrating information content and structural relationship to improve the GO-based similarity measure between proteins. 1 2010.

[52] Andreas Schlicker, Francisco S Domingues, Jörg Rahnenführer, and Thomas Lengauer. A new measure for functional similarity of gene products based on Gene Ontology. *BMC bioinformatics*, 7(1):302, 1 2006.

[53] Francisco M Couto, M Silva, and P Coutinho. Implementation of a Functional Semantic Similarity Measure between Gene-Products. *Recherche*, (DI/FCUL TR 03–29), 2003.

[54] Shobhit Jain and Gary D Bader. An improved method for scoring protein-protein interactions using semantic similarity within the gene ontology. *BMC bioinformatics*, 11(1):562, 1 2010.

[55] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* -, pages 133–138, Morristown, NJ, USA, 6 1994. Association for Computational Linguistics.

[56] Antonio Sanfilippo, Christian Posse, Banu Gopalan, Rick Riensche, Nathaniel Beagley, Bob Baddeley, Stephen Tratz, and Michelle Gregory. Combining Hierarchical and Associative Gene Ontology Relations With Textual Evidence in Estimating Gene and Gene Product Similarity. *IEEE Transactions on Nanobioscience*, 6(1):51–59, 3 2007.

[57] Pietro H Guzzi, Marco Mina, Concettina Guerra, and Mario Cannataro. Semantic similarity analysis of protein data: assessment with biological features and issues. *Brief Bioinform*, 13(5):569–585, 2012.

[58] David Sánchez, Montserrat Batet, and David Isern. Ontology-based information content computation. *Knowledge-Based Systems*, 24(2):297–303, 3 2011.

[59] Catia Pesquita, Daniel Faria, André O. Falcão, Phillip Lord, and Francisco M. Couto. Semantic Similarity in Biomedical Ontologies. *PLoS Computational Biology*, 5(7):e1000443, 2009.

[60] David Sánchez and Montserrat Batet. A semantic similarity method based on information content exploiting multiple ontologies. *Expert Systems with Applications*, 40(4):1393–1399, 2013.

[61] H Al-Mubaid and H A Nguyen. Measuring Semantic Similarity Between Biomedical Concepts Within Multiple Ontologies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 39(4):389–398, 2009.

[62] Abraham Bernstein, Esther Kaufmann, Christoph Kiefer, and Christoph Bürki. SimPack: A Generic Java Library for Similarity Measures in Ontologies. *University of Zurich*, page 20, 2005.

[63] Mingxin Gan, Xue Dou, and Rui Jiang. Review Article From Ontology to Semantic Similarity : Calculation of Ontology-Based Semantic Similarity. 2013, 2013.

[64] T.D. Breaux and J.W. Reed. Using Ontology in Hierarchical Information Clustering. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, 00(C):1–7, 2005.

[65] Karina Gibert, Aïda Valls, and Montserrat Batet. Introducing semantic variables in mixed distance measures: Impact on hierarchical clustering. *Knowledge and Information Systems*, 40:559–593, 2014.

[66] Remco R Bouckaert, Eibe Frank, Mark Hall, Richard Kirkby, Peter Reutemann, Alex Seewald, and David Scuse. Weka Manual-3-6-13. pages 1–327, 2013.

[67] Narendra Sharma, Aman Bajpai, and Ratnesh Litoriya. Comparison the various clustering algorithms of weka tools. *International Journal of Emerging Technology and Advanced Engineering*, 2(5):73–80, 2012.

[68] Y Xu, M Guo, W Shi, X Liu, and C Wang. A novel insight into Gene Ontology semantic similarity. *Genomics*, 101(6):368–375, 2013.

[69] N M Luscombe, D Greenbaum, and M Gerstein. What is bioinformatics? A proposed definition and overview of the field. *Methods of information in medicine*, 40(4):346–58, 1 2001.

[70] UniProt - The Universal Protein Resource. http://www.uniprot.org/.

[71] Reactome | Pathway Browser. http://www.reactome.org/PathwayBrowser/.

[72] NCBI GenBank. http://www.ncbi.nlm.nih.gov/genbank/.

[73] GOGraphViewer. http://activeomics.org:3838/GOGraphViewer/.