Leiden University

Computer Science & Economics

Visualizing implicit associations between biomedical concepts

Name:	M.W.W. Ackermans
Studentnr:	1055267
Date:	August 31, 2015
1st supervisor:	E.A. Schultes
2nd supervisor:	K.J. Wolstencroft

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Visualizing implicit associations between biomedical concepts

M.W.W. Ackermans

Abstract

The growth rate of biomedical literature has now surpassed the ability of experts to keep up even in narrowly defined knowledge domains. New methods are being developed continuously to reduce this information overload. The Biosemantics Group at LUMC have text-mined abstracts from PubMed, creating a large semantic network of associated concepts and their co-occurrence frequencies. Although these data have been successfully used in automated knowledge discovery applications, it is still not understood how the network is structured and how this structure can lead to relevant novel associations. Here, we have developed an application called CPVisuals that visually represents concept profiles derived from the semantic network of concepts. We then systematically sample a rank-ordered list of all possible gene-disease associations, and use the CPVisuals application to show how structures within concept profiles contributes to high-ranking and low-ranking gene-disease pairs. We observe a potential transition separating certain knowledge from uncertain knowledge and conjecture that discovery tends to occur near this transition.

Acknowledgements

I am grateful to my supervisor Erik Schultes, whose expertise, drive and enthusiasm inspired me even more to work on a topic that was of great interest to me. It was a pleasure working with him.

I would like to express my gratitude towards my second reader Katy Wolstencroft, for taking the time to provide me with useful feedback in her busy schedule.

I would also like to acknowledge the BioSemantics Group, with a special thanks to Kristina Hettne for her enthusiastic response to the first version of CPVisuals and her suggestions on how to improve it.

Also from the BioSemantics Group, I would like to thank Mark Thompson, for providing the concept profiles and the gene-disease match scores.

I am grateful to Kees Burger from the BioSemantics Group, for giving CPVisuals an place online.

I would like to thank my mom, dad, brothers, roommates and my friends for their inspiration, patience and understanding.

Finally and most importantly, I would like to thank my girlfriend Vanessa, for putting up with me for the last couple of months, especially during the summer holidays.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

It has become apparent that the academic world cannot keep up with the huge amount of new research data and publications each day. In 2013, a total of 1,135,634 publications were added to the PubMed/MedLine database, averaging to roughly 3,111 new publications each day [3]. This has led to the situation in which it is impossible for scientists to discover all research relevant to their area of expertise. Next to new data flowing into the publication databases, there is the risk of losing relevant findings made in the past.

This issue has been noticed in the last few years and automated methods have been developed to cope with this. Text-mining literature is a viable option and was introduced by Don Swanson [13]. An application of his method is the discovery of associated biomedical concepts even if they do not appear together in the same document. The association would be discovered by looking at an intermediate concept, which is associated with both concepts. For example, concepts X and Y are mentioned and researched in an article. Concepts Y and Z co-occur in a different article. Then it could imply that concepts X and Z are associated through intermediate concept Y. The association between concepts X and Z could therefore be interesting to investigate if no prior research about this association exists.

The BioSemantics Group, a collaboration between LUMC and LIACS, implemented Swanson's idea. Abstracts of the PubMed database were text-mined, resulting in a large database of concepts which were disambiguated biomedical terms. These concepts represent genes, diseases, proteins, symptoms, biological processes and other biomedically relevant semantic types. Then, concept profiles were generated by making a list of all concepts for each concept they are associated with. These links between concepts are called *explicit*. For every associated concept the link will be given a weight. This weight represents the mutual information of co-occurrence frequency in the abstracts. Calculation of this weight is explained in Section 2.1.2. For all gene-disease concept profile pairs, a match score was computed representing the inner product of all weights of the overlapping concepts between two concept profiles. These links between concept profile pairs are called *implicit*. The strength of an implicit link is defined by the match score. The idea is that the concept pairs with the highest match scores would be the most promising pairs to research. [5] [19] [20]

In Figure 1.1, a concept profile pair can be seen. The concept profile of concept X consists of concepts Y1, Y2 and Y3, which are explicitly linked as represented by the black line. Concept Z is also associated with these concepts Y1, Y2 and Y3. Thus a match score can be calculated based on the inner products of the weights of these overlapping concepts Y1, Y2 and Y3. The implicit link between concepts X and Z is represented by the red dotted line.

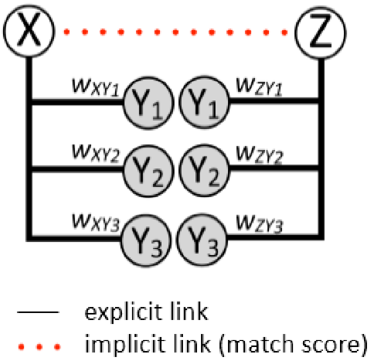


Figure 1.1: Implicit versus explicit links

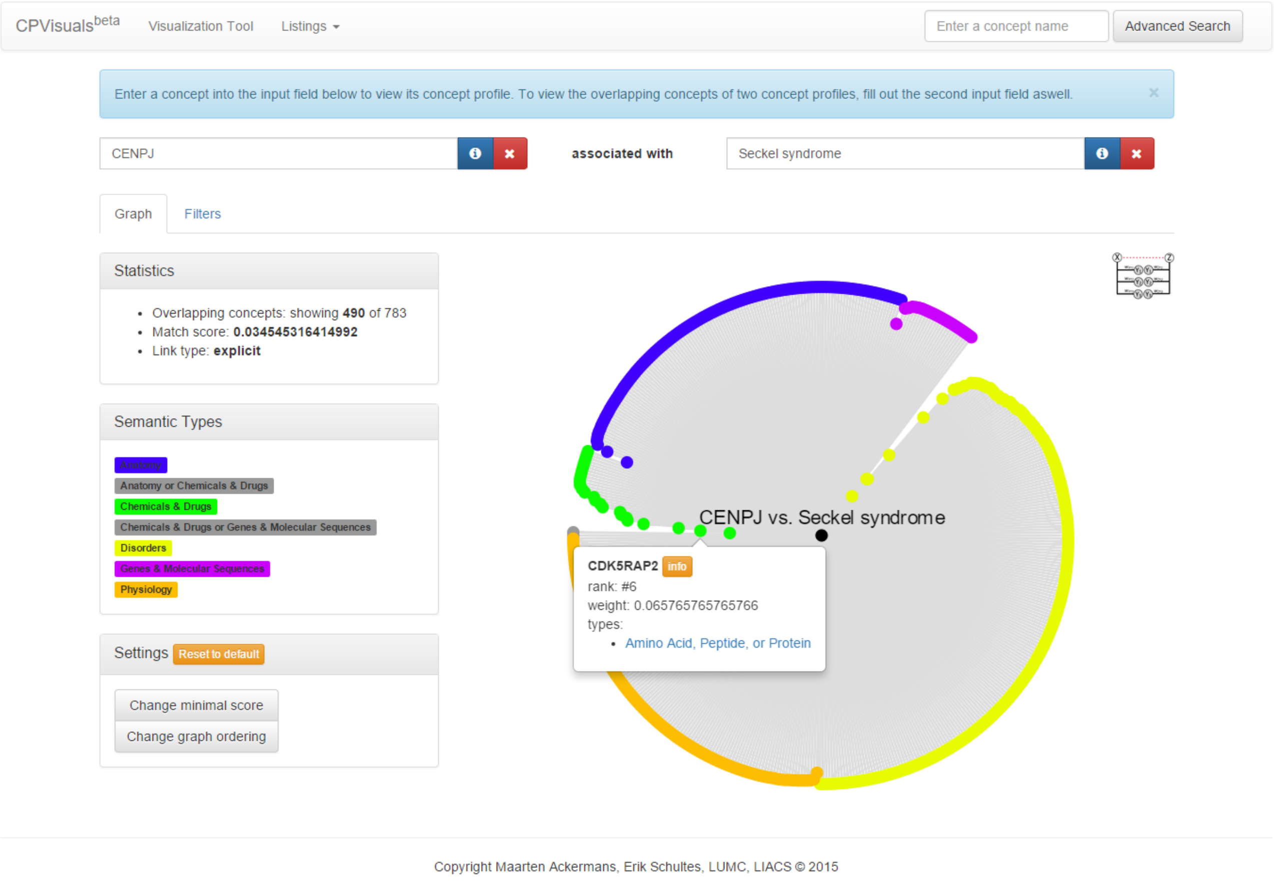
1.2 The problem

Presently, concept profiles and concept profile pairs are (for analytical purposes) treated as vectors. Although the information contained in concept profiles can be valuable to the biologist this information remains largely inaccessible to the human user. One approach to exposing this information in a user-friendly way is through graphical representation. Here, we develop a visualization method called CPVisuals.

1.3 Solution: CPVisuals

For this study, we built a web application tool CPVisuals to visualize concept profiles. The tool is able to visualize the structures of concept profiles individually. Another feature is the ability to visualize the overlapping concepts of concept profile pairs. In this project, the CPVisuals database consists of the concept profiles generated in the study by Hettne et al [5]. A demonstration of the tool can be seen in Figure 1.2 below. The figure shows a visualization of a concept profile pair: CENPJ associated with Seckel Syndrome.

We use this visualization tool to search for patterns in gene-disease concept profile pairs that will drive match scores.



Enter a concept into the input field below to view its concept profile. To view the overlapping concepts of two concept profiles, fill out the second input field aswell.

CENPJ associated with Seckel syndrome

Statistics

- Overlapping concepts: showing **490** of 783
- Match score: **0.034545316414992**
- Link type: **explicit**

Semantic Types

🟦 Anatomy
▨ Anatomy or Chemicals & Drugs
🟩 Chemicals & Drugs
▨ Chemicals & Drugs or Genes & Molecular Sequences
🟨 Disorders
🟪 Genes & Molecular Sequences
🟧 Physiology

CDK5RAP2 info
rank: #6
weight: 0.065765765765766
types:

- Amino Acid, Peptide, or Protein

Copyright Maarten Ackermans, Erik Schultes, LUMC, LIACS © 2015

Figure 1.2: Concept profile pair: CENPJ associated with Seckel Syndrome

Chapter 2

Methods

In this chapter we describe analytical methods and software that were used in this project.

2.1 Concept profiles

2.1.1 Data formats

All the data concerning concepts and concept profiles are described in the study by Hettne et al. [5].

The data related to biomedical *concepts* are located in the CPVisuals MySQL database [9]. In the *concepts* table, each row represents one concept. A row consists of a unique identifier integer *id*, a string *name* and a string *definition*. The CPVisuals database currently consists of 687,718 disambiguated concepts. These concepts can be related to 107 semantic types. Each semantic type is classified hierarchically into a category. See Appendix A. The size of the CPVisuals database is 463.9MB. Over 17 million PubMed abstracts as of January 1980 were text-mined to produce this data.

The data representing the *concept profiles* are saved in text files. The file name represents the unique concept identifier integer value. This identifier can be used to find the corresponding concept and its metadata in the CPVisuals database. The text file contains lines in the following format: *id,weight*. Each line represents an associated concept. *id* is the unique concept identifier as an integer and *weight* the mutual information between the co-occurrence frequency of concepts as a fraction. A total of 33,368 concept profiles were generated as text files, which amounts to 2.25GB on disk.

2.1.2 Calculation of concept weight

The means of calculating the weight of a concept is based on the method used in the study by Hettne et al. [5]. The weight w_{ij} of a concept j represents the strength of its association to concept i. For two concepts X and Y, four contingencies may occur in relation to their co-occurrence in an article: they both occur, only X occurs, only Y occurs, both X and Y do not occur. An association between X and Y is computed from this 2x2 contingency table by using a measure of mutual information, called the symmetric uncertainty coefficient: $U(X_i, Y_j)$, where H is entropy. See Equation 2.1.

$$w_{ij} = U(X_i, Y_j) = \frac{H(X_i) + H(Y_j) - H(X_i, Y_j)}{\frac{1}{2}(H(X_i + H(X_j)))} \tag{2.1}$$

2.2 CPVisuals visualization

CPVisuals provides two main features: (1) showing the structure of a single concept profile and (2) showing the structure of the overlapping concepts of a concept profile pair. In this section both will be explained and the methods used to generate each of them. A structure consists of a center node, multiple outer nodes and multiple edges connecting the center to the outer nodes. The outer nodes are evenly distributed in a circle around the center node. See Figure 1.2.

2.2.1 Concept profile

For a concept profile, the center node represents the main concept, the outer nodes represent the concepts explicitly linked to the main concept and the edges represent the co-occurrence of concepts with the main concept. The length of an edge represents the weight of the association (based on the co-occurrence frequency of both concepts).

2.2.2 Concept profile pair

For a concept profile pair, the center node represents a composition of both main concepts (the pair), the outer nodes represent the overlapping concepts of the main concept pair and the edge length represents the sum of the weights of the concept existing in both concept profiles. For example, if main concepts X and Z both have explicit links with concept Y, then the weight of Y associated with X plus the weight of Y associated with Z will be the value used to determine the length of the edge.

2.2.3 Graphic calculations

Outer node rotational position

Calculation of the rotational position of an outer node on an imaginary circle around the center node is performed by applying the Equation 2.2 below.

$$r = 360 * \frac{nodeCount}{totalNodes} + offset \tag{2.2}$$

where *r* is the rotation of the node in degrees compared to the center node, *nodeCount* is the amount of nodes who had their rotation calculated so far, *totalNodes* is the total of nodes that will be shown and *offset* is a configuration setting to start at a specific rotation.

Calculating node distance

The distance of the outer node compared to the center node is calculated through the percentage of the weight of the linked concept. It differs for concept profiles and concept profile pairs. Equation 2.3 will demonstrate how the percentage is calculated for an explicitly linked concept in a concept profile and Equation 2.4 will show how it is calculated for an overlapping concept between a concept profile pair. Then the actual length of the edge will be calculated in Equation 2.5.

Concept profile

$$p = \frac{weight - minWeight}{maxWeight - minWeight} \tag{2.3}$$

where *p* is a percentage which is inversely proportional to the maximal length of an edge, *weight* is the weight of the concept explicitly linked to the main concept, *minWeight* is the lowest weight found in the concept profile of the main concept and *maxWeight* is the highest weight found in the concept profile of the main concept.

Concept profile pair

$$p = \frac{(weight_{Y,X} + weight_{Y,Z}) - minWeight}{maxWeight - minWeight} \tag{2.4}$$

where *p* is a percentage which is inversely proportional to the maximal length of an edge, $weight_{Y,X}$ is the weight of the concept Y explicitly linked to the main concept X, $weight_{Y,Z}$ is the weight of the concept Y explicitly linked to the main concept Z, *minWeight* is the lowest sum of weights of two overlapping concepts found in the concept profiles of the main concepts and *maxWeight* is the highest sum of weights of two overlapping concepts found in the concept profiles of the main concepts.

Distance

$$d = ((1 - p) * (maxDistance - minDistance)) + minDistance \tag{2.5}$$

where d is the distance of the outer node compared to the center node (length of an edge), p is the percentage calculated in either Equation 2.3 or 2.4 and *maxDistance* and *minDistance* are configurable settings in pixels to set boundaries for the length of an edge.

Calculating position

The actual rendered *x,y* coordinates are then calculated by the following Equation 2.6.

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} centerX + d * \cos(r * \frac{\pi}{180}) \\ centerY + d * \sin(r * \frac{\pi}{180}) \end{pmatrix} \tag{2.6}$$

where *x,y* are the coordinates of the node, *centerX* and *centerY* are the coordinates of the center node, r is the rotation of the node calculated in Equation 2.2 and d is the distance calculated in Equation 2.5.

2.2.4 Filters

It is possible for the user to filter concepts based on semantic type. This is sometimes desired if not all semantic types are relevant and only clutter up the visualization. If a concept has at least one semantic type which is not filtered out, it will be processed for visualization.

2.2.5 Amount of visible nodes threshold

A concept profile is first processed fully to sort by weight. After that, a selection is made on which concepts to show. The amount of nodes shown is based on two criteria.

Dynamic threshold

The first criterion is a threshold based on the percentile in which the weight belongs. After all concepts have been processed, the highest weight and lowest weight are known, the lowest weight representing 0% and the highest weight representing 100%. Next, the concepts which are below a set percentage on that scale are left out of the selection.

Hard threshold

The second criterion is a hard cut-off of nodes, to prevent a user's browser from crashing if too much nodes are rendered simultaneously. The default setting is 500 nodes.

2.2.6 Semantic type mapping

Each concept belongs to one or more semantic types. This relationship was only available in the database, which proved too slow to process. Therefore, for each concept existing in the database, the corresponding semantic types were exported to a single text file in the following format:

```
conceptid|type1,type2,...,typeN
```

Furthermore, semantic types are classified hierarchically. Each semantic type belongs to a single category. See Appendix A.

Now, before processing the concepts, all semantic types are loaded into memory. For each loaded concept the corresponding semantic type identifier is attached. The next step is to unload the semantic types from memory which are not present in the concepts or that have a filter applied which excludes the semantic type specifically.

2.2.7 Color generation

For each semantic category, a color is generated and appointed to it. If a concept has semantic types spanning multiple categories, it is considered a new, unique semantic category. Composite categories are colored grey.

To ensure that each singular category has a distinct color, the golden ratio conjugate [21] was used. The golden ratio conjugate has a value of *0.618033988749895*. It enables us to use a sequential formula: for each new color we can generate the next distinct color. Each singular category is given an index based on its first appearance. The result from the formula is passed to a HSV color generation function as the *hue*, *saturation* is set at 0.99 and *value* is set at 0.99 to provide intense colors. The color is then converted to RGB format. See Equation 2.7 below.

$$distinctValue = (i * 0.618033988749895) \mod 1 \tag{2.7}$$

The variable i represents the index of the category. *distinctValue* is the value passed to the HSV color generation function.

For the purposes of this experiment, the following five semantic type categories were assigned the colors: Anatomy (blue), Chemicals & Drugs (green), Disorders (yellow), Genes & Molecular Sequences (purple) and Physiology (orange). As long as the semantic categories are not modified (because of updates in the UMLS vocabulary), the colors remain the same.

2.3 Software

CPVisuals is a web application and therefore makes use of various scripting languages, frameworks and libraries, which are explained in this section.

2.3.1 PHP

PHP [18] is the driver and the scripting language upon which the whole application has been built. It is easy to set up, lightweight, and above all widely used in web applications across the internet. Accessibility is a key point in deploying a web application; by using PHP no additional prerequisites exist for an arbitrary user.

2.3.2 Laravel Framework

Laravel [11] is the underlying PHP framework, which provides a Model-View-Controller architectural design. This aids in keeping the web application modular and organized. It also enables quick deployment and easy maintenance of web applications by providing various utilities.

2.3.3 MySQL database

MySQL [9] is the underlying database of the web application. It is capable of retrieving data quickly when queried. MySQL is most commonly used with web applications, has easy integration with PHP and Laravel and thus a logical addition to the application.

2.3.4 JavaScript

JavaScript is a scripting language which enables features that add to the user experience, making the web application more interactive. It is executed in the browser of the client and exists in every standard client browser.

2.3.5 jQuery

jQuery [17] is an extension of JavaScript and makes use of background calls to load the requested data directly into a part of the page, removing the need to reload the page completely.

2.3.6 Cytoscape.js

Cytoscape.js [15] is the interactive lightweight web version of the desktop application Cytoscape [16], which is used for displaying (large) networks and graphs. This addition provides the actual visual representation of concept profiles and concept profile pairs. It is easily integrated into the web application by using JavaScript and jQuery.

2.3.7 Bootstrap

Twitter's Bootstrap [10] provides the look-and-feel. It enhances the look of the web application by providing upgraded HTML components, making it more aesthetically pleasing.

2.3.8 Typeahead.js

Twitter's Typeahead.js [4] makes use of jQuery to provide a dropdown with instant search results sorted by match rating when entering a concept into the input field. It searches while the user types and highlights the matching part of the concept.

2.4 Analytics

For our experiment, we parsed the Gene-Disease database. This database consists of concept profile pairs of genes and diseases.

This database, from a study by Hettne et al. [5], was built pairing 19,113 gene concept profiles and 21,847 disease concept profiles, resulting in a total of (19,113 x 21,847 =) 417,561,711 possible gene-disease pairs. 213,489,335 gene-disease pairs, which is more than half of the total, lacked sufficient literature backing to build a concept profile for either one or both concepts, and were thus disregarded. The result is a total of 204,072,376 gene-disease pairs present in the database.

These gene-disease pairs are rank-ordered by match score. Concept profile pairs may have explicit or implicit associations. Here we focus on implicit associations only. The Gene-Disease database is represented by Figure

2.1. The small curve represents the explicitome, which are explicit gene-disease associations. The much larger curve represents the implicitome, which are indirect, implicit gene-disease associations.

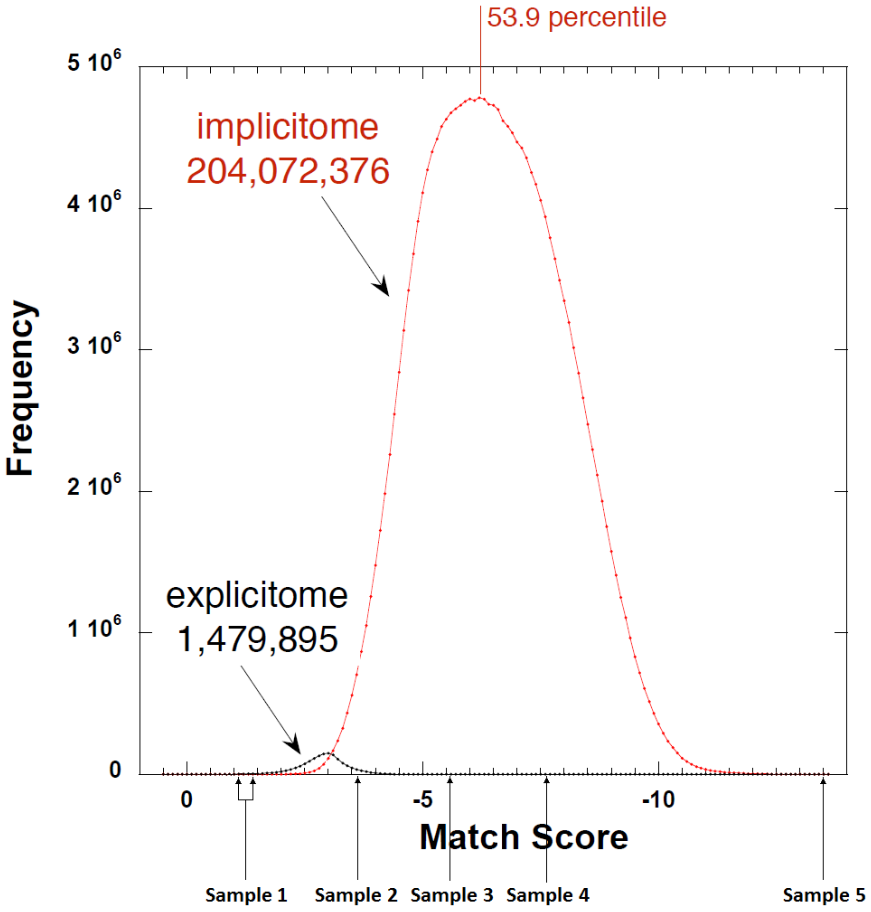


Figure 2.1: Implicitome (red) and explicitome (black): Amount of gene-disease pairs versus corresponding match score

We take five samples, referred to as Sample 1-5, of 10 gene-disease pairs from the Gene-Disease database, at equal intervals (rank-ordered by match score): Top 10, 75th percentile, 50th percentile, 25th percentile and lowest 10. The location of each sample in the database is marked in Figure 2.1. For example, the match scores of the pairs taken from the 75th percentile are higher than 75% of the match scores of all the pairs in the database. We take samples to see if the visual representations of gene-disease pairs from one sample differ significantly from gene-disease pairs from another sample.

Note that the highest match score for an implicit concept profile pair was found at rank 4657. The top 10 implicit concept profile pairs were found between rank 4657 and rank 6584, with a match score higher than 99.996% of all match scores of gene-disease pairs calculated. The other samples are taken from the database at the specified intervals.

Chapter 3

Results

In this section we will use the CPVisuals tool to provide visualizations for each of our concept profile pairs in five sample groups (Samples 1-5). We hope to find new insights that are driving the match scores.

3.1 Experiment

In this section, we analyze the results of our five samples. In each case we depict the CPVisuals visual representation of the gene-disease pair. We filter the semantic type categories to the following: Anatomy, Chemicals & Drugs, Disorders, Genes & Molecular Sequences and Physiology. Each of these categories has all the semantic types enabled (thus not filtered out). The corresponding acronyms and node colors can be found in Table 3.1.

Semantic Category	Acronym	Node color
Anatomy	ANAT	blue
Chemicals & Drugs	CHEM	green
Disorders	DISO	yellow
Genes & Molecular Sequences	GENE	purple
Physiology	PHYS	orange

Table 3.1: Semantic categories with corresponding acronyms and node colors

3.1.1 Sample 1

In Sample 1, we have obtained the 10 highest (implicit gene-disease pair) match scores from the Gene-Disease database. Sample 1 is depicted in Figure 3.1 and more detailed information can be found in Table 3.2.

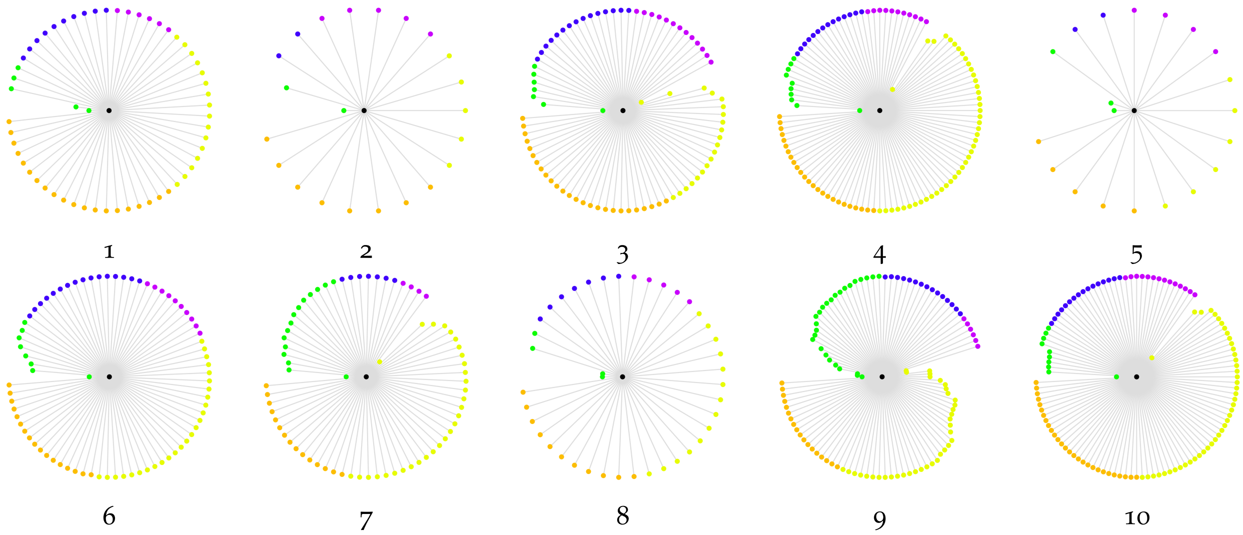


Figure 3.1: CPVisuals visualizations of overlapping concepts of the highest 10 rated gene-disease pair match scores

Index	Gene concept	Disease concept	Nodes	Match score	Rank
1	TTBK1	SCA11	57	0.0918	4657
2	CNNM3	cone-rod dystrophy and amelogenesis imperfecta	22	0.0856	5049
3	WDR62	MCPH4	81	0.0834	5208
4	SUMO4	insulin-dependent diabetes mellitus 8	104	0.0801	5448
5	CWH43	hyperphosphatasia with mental retardation	20	0.0796	5502
6	B9D1	MKS2	77	0.0751	5846
7	TUBB4	DYT2	71	0.0719	6148
8	AP4E1	SPG12	41	0.0717	6159
9	DDHD1	SPG21	102	0.0705	6276
10	SUMO4	insulin-dependent diabetes mellitus 4	112	0.0677	6584

Table 3.2: Overlapping concepts of the highest 10 rated gene-disease pair match scores

We see that the high match scores are always caused by one or two highly weighted overlapping concepts of the semantic type Amino Acid, Peptide or Protein (in the category Chemicals/Drugs (green)). In half of the cases it is also caused by highly weighted overlapping concepts of the semantic type Disease or Syndrome (in the category Disorders (yellow)).

3.1.2 Sample 2

In Sample 2, we have obtained 10 gene-disease pair match scores at the 75th percentile from the Gene-Disease database. Sample 2 is depicted in Figure 3.2 and more detailed information can be found in Table 3.3.

The visualizations from Sample 2 are usually more dense than the visualizations from Sample 1. This is most

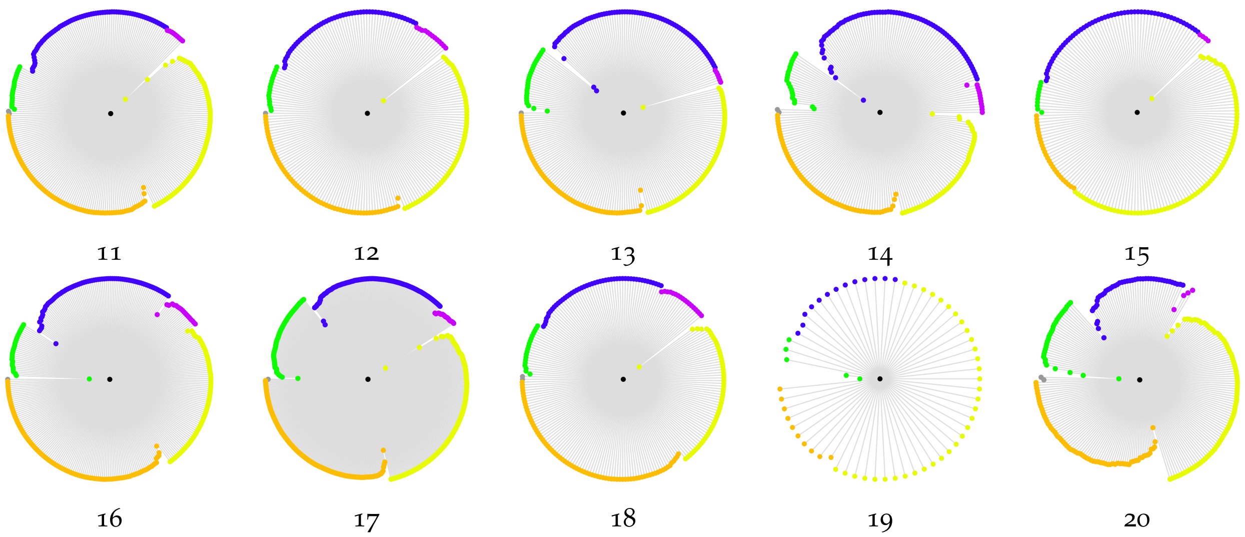


Figure 3.2: CPVisuals visualizations of overlapping concepts of 10 in the 75th percentile rated gene-disease pair match scores

Index	Gene concept	Disease concept	Nodes	Match score	Rank
11	AMPH	Premenstrual syndrome	278	4.6054E-6	51018094
12	KIAA1409	spondylocarpotarsal synostosis syndrome	213	4.6054E-6	51018095
13	CNTN1	hemodialysis-associated amyloidosis	246	4.6054E-6	51018096
14	TFAP2D	Laminitis	228	4.6054E-6	51018097
15	ODZ4	Onychogryposis	187	4.6054E-6	51018098
16	ADCY3	Fowlpox	311	4.6054E-6	51018099
17	DDX17	Secondary hypertension	500	4.6054E-6	51018100
18	MTHFD2	Hypotrichosis	225	4.6054E-6	51018101
19	NANOS3	Anal spasm	62	4.6054E-6	51018102
20	CDKN1C	anemia of renal disease	239	4.6054E-6	51018103

Table 3.3: Overlapping concepts of 10 in the 75th percentile rated gene-disease pair match scores

likely caused by the fact that individual weights of overlapping concepts lie more closely together. We can compare this to Sample 1, where a few outliers cause lots of overlapping concepts not to show due to the dynamic cutoff as described in Section 2.2.5.

In contrast to the results from Sample 1, the distribution of semantic types in Sample 2 are relatively uniform.

3.1.3 Sample 3

In Sample 3, we have obtained 10 gene-disease pair match scores at the 50th percentile from the Gene-Disease database. Sample 3 is depicted in Figure 3.3 and more detailed information can be found in Table 3.4.

Weights of overlapping concepts lie even more closely together compared to Sample 2, which is what causes the graphs to show more variation in node distances. For some graphs seashell-like structures seem to occur. This is caused by the ordering of overlapping concepts (first by semantic type, then by weight) and because the weight differences between the overlapping concepts are small.

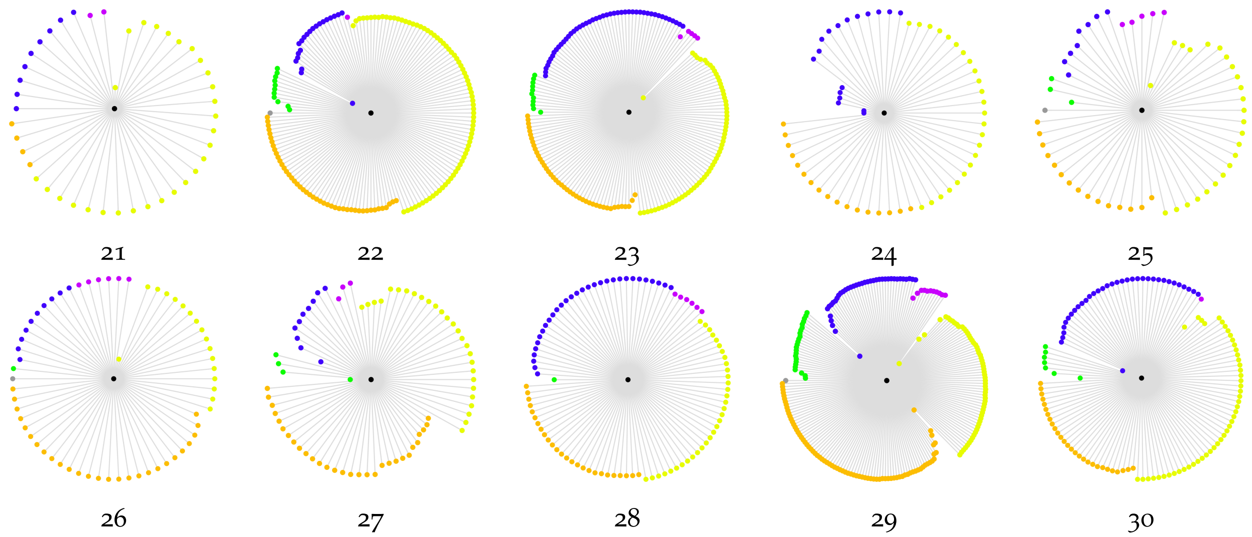


Figure 3.3: CPVisuals visualizations of overlapping concepts of 10 in the 50th percentile rated gene-disease pair match scores

Index	Gene concept	Disease concept	Nodes	Match score	Rank
21	TPMT	internal jugular vein stenosis	43	3.8132E-7	102036188
22	CNTN5	atrioventricular conduction disorder	156	3.8132E-7	102036189
23	FNBP1L	derangement of temporomandibular joint	168	3.8132E-7	102036190
24	NKX2-8	Bronchocentric granulomatosis	58	3.8132E-7	102036191
25	ERCC3	HAE III	56	3.8132E-7	102036192
26	SAGE1	autosomal recessive cutis laxa	62	3.8132E-7	102036193
27	MON2	cowden-like syndrome	74	3.8132E-7	102036194
28	PSAT1	Orbivirus Infection	97	3.8132E-7	102036195
29	CMPK1	Lens Diseases	214	3.8132E-7	102036196
30	PSME1	exudative otitis media	119	3.8132E-7	102036197

Table 3.4: Overlapping concepts of 10 in the 50th percentile rated gene-disease pair match scores

3.1.4 Sample 4

In Sample 4, we have obtained 10 gene-disease pair match scores at the 25th percentile from the Gene-Disease database. Sample 4 is depicted in Figure 3.4 and more detailed information can be found in Table 3.5.

The visualizations become more sparse.

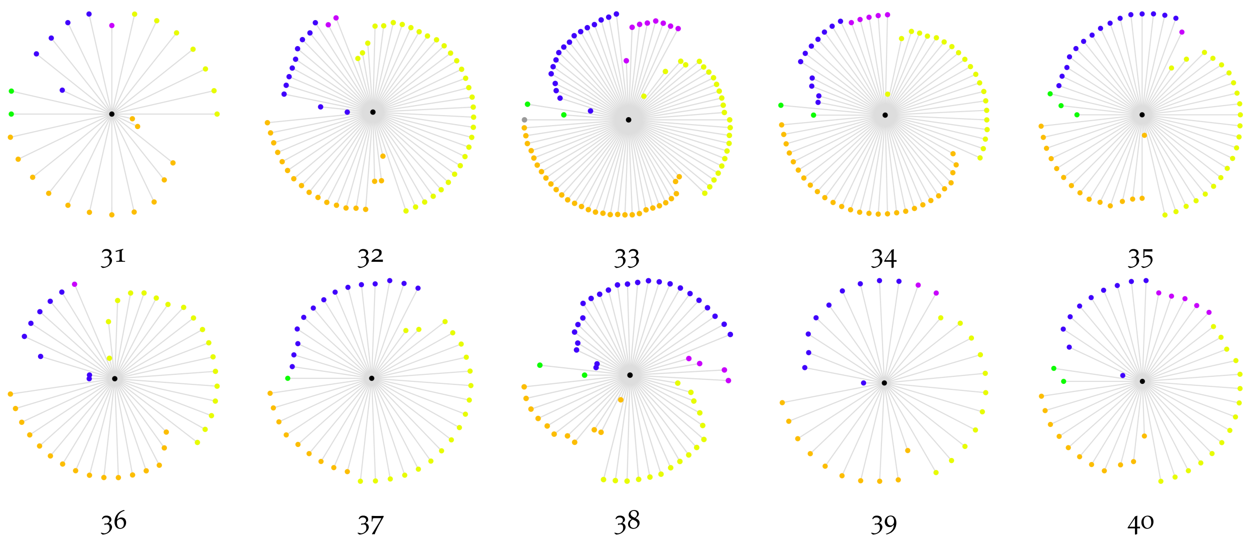


Figure 3.4: CPVisuals visualizations of overlapping concepts of 10 in the 75th percentile rated gene-disease pair match scores

Index	Gene concept	Disease concept	Nodes	Match score	Rank
31	CREBZF	Superficial folliculitis	28	2.7165E-8	153054282
32	CTNNA2	asymmetrical sensorineural hearing loss	63	2.7165E-8	153054283
33	PPFIBP2	disorder of small intestine	82	2.7165E-8	153054284
34	GALNT13	Eosinophilic meningoencephalitis	67	2.7165E-8	153054285
35	GPR97	autoimmune limbic encephalitis	56	2.7165E-8	153054286
36	MTA3	Suppurative parotitis	43	2.7165E-8	153054287
37	FFAR2	rheumatoid arthritis of temporomandibular joint	43	2.7165E-8	153054288
38	TMSB10	adult spinal muscular atrophy	57	2.7165E-8	153054289
39	VSTM1	Amalgam tattoo	33	2.7165E-8	153054290
40	RPL30	amyloid of vitreous	43	2.7165E-8	153054291

Table 3.5: Overlapping concepts of 10 in the 25th rated gene-disease pair match scores

3.1.5 Sample 5

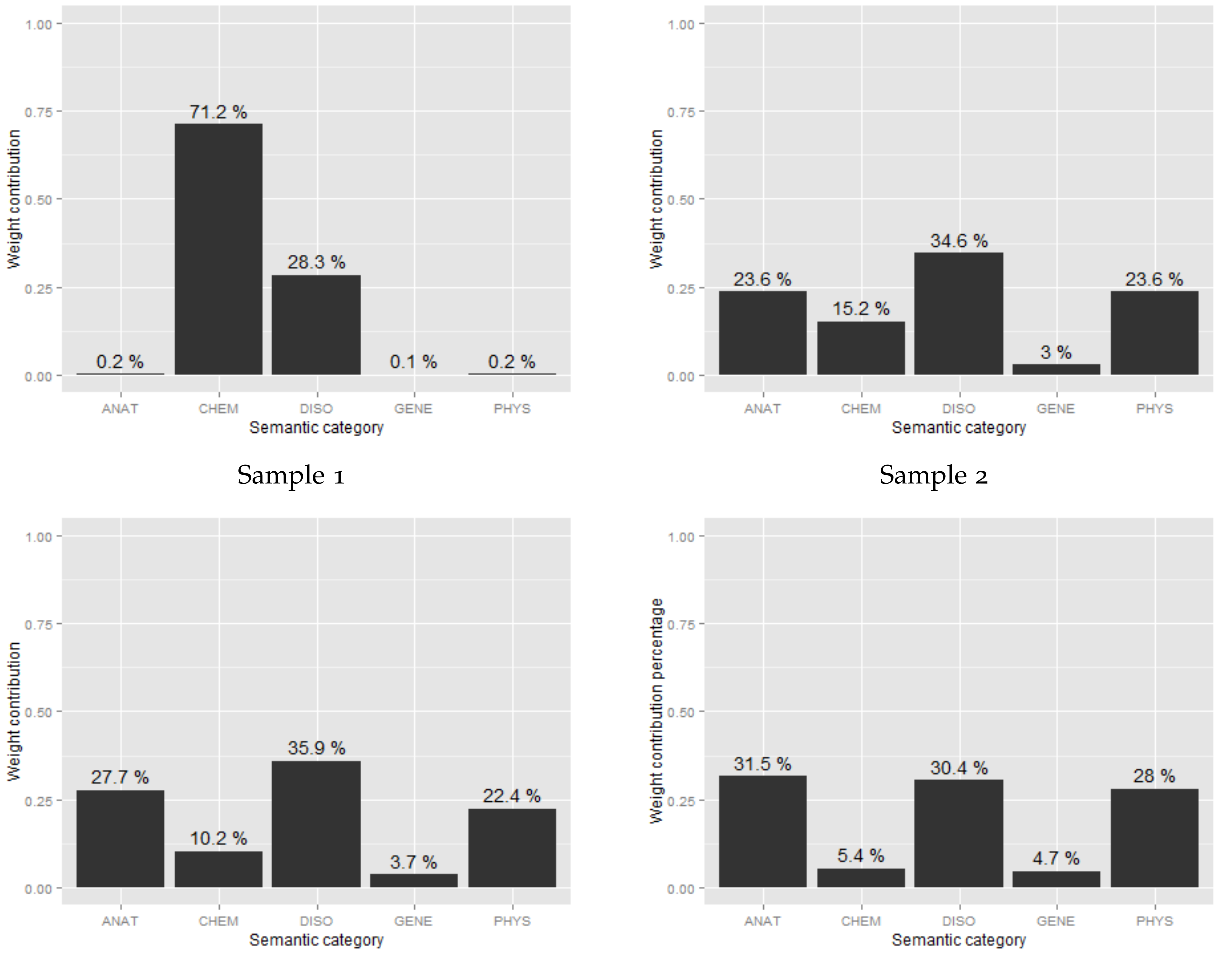
In Sample 5, we have obtained the bottom 10 gene-disease pair match scores from the Gene-Disease database. This sample represents the tail of the implicitome. There are no links with overlapping concepts. Therefore the resulting visualizations do not show any associations.

3.2 Summary

In this section we will perform three analyses on Samples 1-4. In the first analysis we will see the weight contribution of overlapping concepts to match scores per semantic category. In the second analysis we will see the amount of overlapping concepts (nodes) per index. The third analysis will show the weight variance of overlapping concepts per index. Note that index refers to the corresponding visualization in Section 3.1.

3.2.1 Analysis 1: Weight contribution to match score per semantic category

For each sample, we did an analysis of the overlapping concepts for each semantic category. First we summed the weights of these concepts per semantic category. We then converted this number into a percentage of contribution to the total sum of weights. The result can be seen in Figure 3.5.



Sample 1

Sample 2

Sample 3

Sample 4

Figure 3.5: Weight contribution to match score per semantic category per sample

In Sample 1, the only two categories to significantly contribute to the match scores were the categories Chemicals & Drugs (71.2%) and Disorders (28.3%). The other three categories Anatomy (0.2%), Genes & Molecular Sequences (0.1%) and Physiology (0.2%) are (almost) not visible in the plot, thus not significant contributors to the match scores. In Samples 2-4, a pattern of 3 large pillars (representing categories) is visible. Categories Anatomy (23.6%, 27.7%, 31.5%), Disorders (34.6%, 35.9%, 30.4%) and Physiology (23.6%, 22.4%, 28%) contribute 80% to 90% of the total. Category Genes & Molecular Sequences remains stable in Samples 2-4 around 3% to 5% contribution. For all samples, the category Disorders is the only solid contributing category, ranging between 28.3% and 35.9% contribution. Also, category Chemicals & Drugs degrades in contribution over all samples, from 71.2% in Sample 1, 15.2% in Sample 2, 10.2% in Sample 3 to 5.4% in

Sample 4.

3.2.2 Analysis 2: Amount of overlapping concepts (nodes) per index

We plotted the count of overlapping concepts (called nodes) per index. The result is depicted in Figure 3.6. We also calculated the mean, median and standard deviation of nodes per sample. These statistics can be found in Table 3.6.

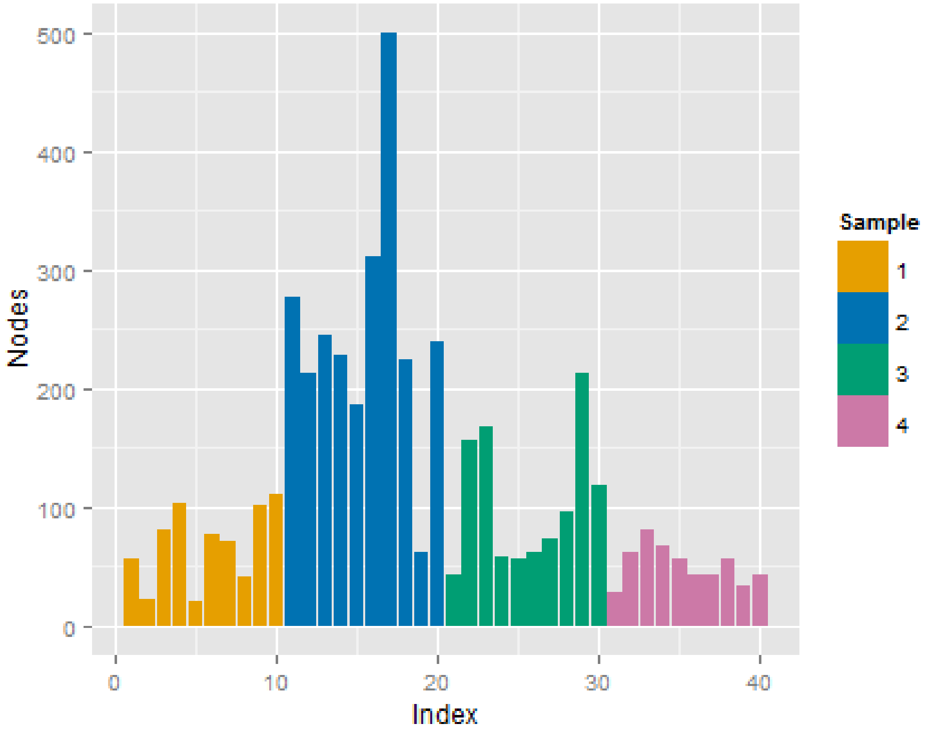


Figure 3.6: Amount of overlapping concepts (nodes) per index, coloured by sample

Sample	Mean	Median	St. Dev.
1	68.7	74.0	33.19
2	248.9	233.5	110.01
3	104.7	85.5	57.71
4	51.5	49.5	16.55

Table 3.6: Mean, median and standard deviation of amount of overlapping concepts per sample

From Figure 3.6 and Table 3.6, we can determine that Sample 2 has the most overlapping concepts with a mean of 248.9 nodes and a large standard deviation of 110.01. There is also one outlier reaching the hard threshold at 500 nodes (index 17). For Sample 1 we found a mean of 68.7 nodes with a standard deviation of 33.19. The comparison between Sample 1 and 2 is interesting - it suggests that the highest ranked concept pairs have a smaller amount of overlapping concepts contributing to the match score than lower ranked concept pairs. This must mean that each overlapping concept generally has a higher weight in the highest ranked concept pairs than in the lower ranked concept pairs. In Sample 3, we found a mean of 104.7 nodes and a standard deviation of 57.71 nodes. In Sample 4, we found a mean of 51.5 nodes and a standard deviation of 16.55 nodes. This means that in Samples 2-4, the amount of overlapping concepts degrades along with its variance.

3.2.3 Analysis 3: Weights of overlapping concepts per index

We plotted the weight of each overlapping concept per index. The result is depicted in Figure 3.7. Note that the y-scale (representing the weight of an overlapping concept) in this figure is logarithmic. We also calculated the mean, median and standard deviation of weights of overlapping concepts per sample. These statistics can be found in Table 3.7.

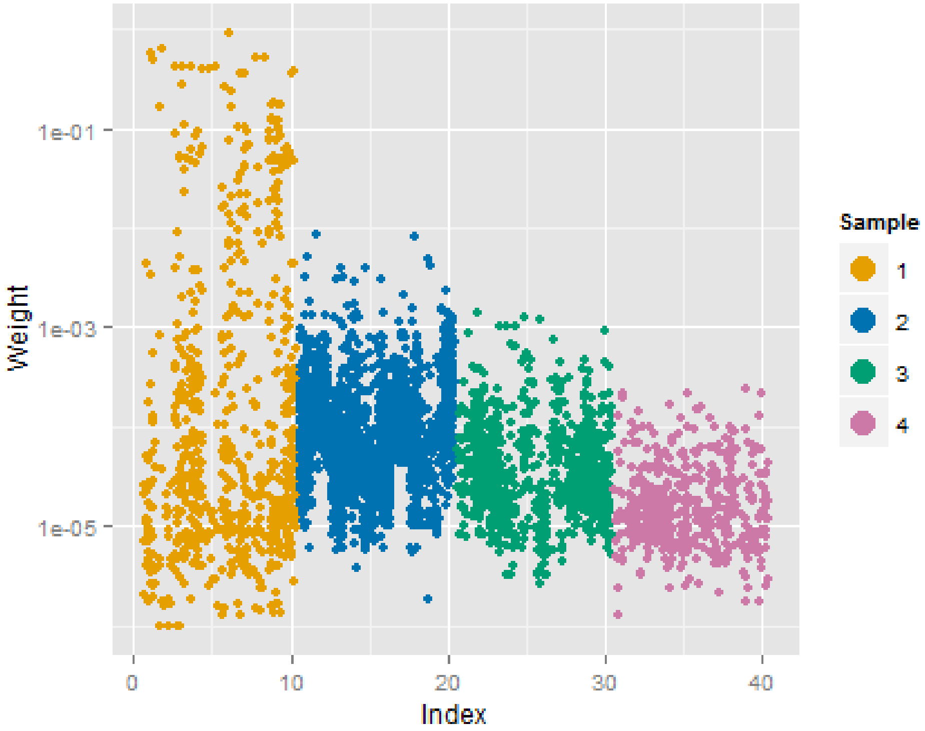


Figure 3.7: Weights of overlapping concepts per index, coloured by sample

Sample	Mean	Median	St. Dev.
1	0.0201816	0.0000361	0.0792601
2	0.0001668	0.0000806	0.0003871
3	0.0000675	0.0000342	0.0001216
4	0.0000252	0.0000141	0.0000319

Table 3.7: Mean, median and standard deviation of weights of overlapping concepts per sample

From Figure 3.7 and Table 3.7, we can clearly see that Sample 1 has the largest variance of weights. Also it is the only sample with weights greater than a value of 1.0E-2. Interesting to see are the medians of these samples. The median of Sample 1 is 3.61E-5, whereas Sample 2 has a greater median of 8.06E-5. Sample 3 follows with a smaller 3.42E-5. Lastly there is Sample 4 with the smallest median of 1.41E-5. The variance degrades from a standard deviation of 7.92601E-2 in Sample 1 to 3.19E-5 in Sample 4. Thus the weights of the lower ranked overlapping concepts are more concentrated.

Chapter 4

Conclusions

In this section, we will first provide an outline of this study. Then, we will discuss the effectiveness of CPVisuals. Lastly we will discuss the results of our experiment and the patterns that we have found.

In Section 1, we have provided a background of the issues we face today in terms of scientific literature abundance along with efforts made prior to this study to solve those issues. We introduced concept profiles and explained implicit and explicit associations between concepts. Also, we introduced CPVisuals, our concept profile visualization tool. We explained that the objective of CPVisuals is to aid in prioritizing research by pointing out promising concept profile pairs and to avoid prior research from getting lost in the large heap of literature. Then, we explained that our goal for this study is to use CPVisuals to discover patterns in gene-disease concept profile pairs that will drive match scores. In Section 2, we described analytical methods that CPVisuals uses to visualize concept profile pairs. We also discussed software used to produce CPVisuals. Lastly, we discussed the Gene-Disease database and the methods we used to extract five samples for our experiment. In Section 3, we discussed the results of our experiment. We depicted the visualizations along with statistical data corresponding to the samples in the experiment. We performed three analyses on these samples and discovered several patterns.

In the first analysis, we were looking for patterns in semantic categories. For simplicity, we identified five semantic categories that would seem most relevant to the biologist. In this analysis we were hoping to identify semantic categories that could be indicative of the match score value. In this regard, we were partially successful. For the highest ranking match scores, the two semantic categories Chemicals & Drugs and Disorders were dominating. The other three categories were almost non-existent. However, in Samples 2-4, the distributions are approximately the same. They seem to have no discrimination for match score. Also, the influence of the Chemicals & Drugs category has dramatically decreased in these last samples, while the Disorders category remained roughly the same. The lower ranking associations also demonstrate a balance of

Anatomy of Physiology categories, that remain constant over match score. Surprisingly, overlapping concepts in the Genes & Molecular Sequences category remain low throughout the samples and contribute little to the match score. From these data, it appears that high match scores are driven by overlapping concepts in the category Chemicals & Drugs. It is not possible to conclude from these data what role the other semantic categories are playing in determining match score. This suggests that we may need to expand this analysis to include all semantic categories.

In the second and third analyses, we observe in the Samples 1 and 2 (match scores in the 25th percentile or higher) a balance between number of nodes on one hand and the magnitude of weights on the other hand in relation to match score. Specifically, Sample 1 has on average a smaller number of nodes but those nodes have a higher average weight. In contrast, Sample 2 has on average a higher number of nodes but those nodes have a lower average weight. This means that in Sample 1, the match scores are driven by fewer, stronger links. In Sample 2, there are more, but weaker links driving the match scores.

For the lower ranking concept pairs in Samples 2-4, there is a trend of decreasing weights and decreasing amount of nodes. This degradation confirms our intuition that low match scores are associated with low information content associations, and in the limit are approaching meaningless, random associations. In the same way, the associations in Sample 1 confirms our intuition that high match scores are associated with high weights.

However, in Sample 2 we begin to see a tension between these two limiting cases, where concept pairs demonstrate a co-existence of both high and low information content associations and both sparsely connected and richly connected concepts. At this resolution of sampling, we can identify Sample 2 as a transition between certain knowledge and random associations. We conjecture that new associations (knowledge discovery) occur near this transition zone, possibly located around the explicitome, where highly weighted associations come into contact with low information content associations.

Having sampled match score with a small number of concept pairs (50), it is difficult to locate exactly where this transition occurs. Expanding this analysis to include all 204 million gene-disease concept pairs will help us to more precisely locate the position of this transition. Interestingly, when looking back at Figure 2.1, we see that the explicitome curve peaks in between Samples 1 and 2 . It would be instructive to determine the location of the the transition in relation to the peak of explicit knowledge.

Our CPVisuals concept profile visualization tool has proved to be a useful addition to further investigate structures of concept profiles and concept profile pairs. Based on the intuition a user obtains from the visualizations, it is now possible to prioritize analysis of concept profile pairs, leading to discover knowledge sooner.

Appendix A

Semantic types

Semantic types are grouped by semantic category. Based on the UMLS Semantic Groups [2] [8].

Activities & Behaviors	
Behavior	T53
Daily or Recreational Activity	T56
Event	T51
Governmental or Regulatory Activity	T64
Individual Behavior	T55
Machine Activity	T66
Occupational Activity	T57
Social Behavior	T54

Anatomy	
Anatomical Structure	T17
Body Location or Region	T29
Body Part, Organ, or Organ Component	T23
Body Space or Junction	T30
Body Substance	T31
Body System	T22
Cell	T25
Cell Component	T26
Embryonic Structure	T18
Fully Formed Anatomical Structure	T21
Tissue	T24

Chemicals & Drugs	
Amino Acid, Peptide, or Protein	T116
Antibiotic	T195
Biologically Active Substance	T123
Biomedical or Dental Material	T122
Carbohydrate	T118
Chemical	T103
Chemical Viewed Functionally	T120
Chemical Viewed Structurally	T104
Clinical Drug	T200
Eicosanoid	T111
Element, Ion, or Isotope	T196
Enzyme	T126
Hazardous or Poisonous Substance	T131
Hormone	T125
Immunologic Factor	T129
Indicator, Reagent, or Diagnostic Aid	T130
Inorganic Chemical	T197
Lipid	T119
Neuroreactive Substance or Biogenic Amine	T124
Nucleic Acid, Nucleoside, or Nucleotide	T114
Organic Chemical	T109
Organophosphorus Compound	T115
Pharmacologic Substance	T121
Receptor	T192
Steroid	T110
Vitamin	T127

Concepts & Ideas	
Classification	T185
Conceptual Entity	T77
Functional Concept	T169
Group Attribute	T102
Idea or Concept	T78
Intellectual Product	T170
Language	T171
Qualitative Concept	T80
Quantitative Concept	T81
Regulation or Law	T89
Spatial Concept	T82
Temporal Concept	T79

Devices	
Drug Delivery Device	T203
Medical Device	T74
Research Device	T75

Disorders	
Acquired Abnormality	T20
Anatomical Abnormality	T190
Cell or Molecular Dysfunction	T49
Congenital Abnormality	T19
Disease or Syndrome	T47
Experimental Model of Disease	T50
Finding	T33
Injury or Poisoning	T37
Mental or Behavioral Dysfunction	T48
Neoplastic Process	T191
Pathologic Function	T46
Sign or Symptom	T184

Genes & Molecular Sequences	
Amino Acid Sequence	T87
Carbohydrate Sequence	T88
Gene or Genome	T28
Molecular Sequence	T85
Nucleotide Sequence	T86

Geographic Areas	
Geographic Area	T83

Living Beings	
Age Group	T100
Amphibian	T11
Animal	T8
Archaeon	T194
Bacterium	T7
Bird	T12
Eukaryote	T204
Family Group	T99
Fish	T13
Fungus	T4
Group	T96
Human	T16
Mammal	T15
Organism	T1
Patient or Disabled Group	T101
Plant	T2
Population Group	T98
Professional or Occupational Group	T97
Reptile	T14
Vertebrate	T10
Virus	T5

Objects	
Entity	T71
Food	T168
Manufactured Object	T73
Physical Object	T72
Substance	T167

Occupations	
Biomedical Occupation or Discipline	T91
Occupation or Discipline	T90

Organizations	
Health Care Related Organization	T93
Organization	T92
Professional Society	T94
Self-help or Relief Organization	T95

Phenomena	
Biologic Function	T38
Environmental Effect of Humans	T69
Human-caused Phenomenon or Process	T68
Laboratory or Test Result	T34
Natural Phenomenon or Process	T70
Phenomenon or Process	T67

Physiology	
Cell Function	T43
Clinical Attribute	T201
Genetic Function	T45
Mental Process	T41
Molecular Function	T44
Organism Attribute	T32
Organism Function	T40
Organ or Tissue Function	T42
Physiologic Function	T39

Procedures	
Diagnostic Procedure	T60
Educational Activity	T65
Health Care Activity	T58
Laboratory Procedure	T59
Molecular Biology Research Technique	T63
Research Activity	T62
Therapeutic or Preventive Procedure	T61

Bibliography

[1] Z. Afzal, E. Pons, N. Kang, M. C. Sturkenboom, M. J. Schuemie, and J. Kors. ContextD: an algorithm to identify contextual properties of medical terms in a Dutch clinical corpus. *BMC bioinformatics*, 15(1):373, 2014.

[2] O. Bodenreider and A. T. McCray. Exploring semantic groups through visual approaches. *Journal of biomedical informatics*, 36(6):414–432, 2003.

[3] A. D. Corlan. Medline trend: automated yearly statistics of pubmed results for any query. `http://dan.corlan.net/medline-trend.html`, 2004. Accessed: 2012-02-14 (Archived by WebCite at `http://www.webcitation.org/65RkD48SV`).

[4] J. Harding and Twitter, Inc. Typeahead.js, a flexible javascript library that provides a strong foundation for building robust typeaheads. `http://twitter.github.io/typeahead.js/`. Accessed: 2015-07-04.

[5] K. M. Hettne, M. Thompson, H. van Haagen, E. van der Horst, R. Kaliyaperumal, E. Mina, Z. Tatum, J. F. Laros, E. M. van Mulligen, M. Schuemie, E. Aten, J. den Dunnen, G.-J. van Ommen, M. Roos, P. A. t Hoen, B. Mons, and E. A. Schultes. The implicitome: exposing gene-disease associations hidden in the literature. *submitted for publication*, 2015.

[6] D. Hristovski, B. Peterlin, J. A. Mitchell, and S. M. Humphrey. Using literature-based discovery to identify disease candidate genes. *International Journal of Medical Informatics*, 74(24):289 – 298, 2005. {MIE} 2003.

[7] R. Jelier, M. J. Schuemie, P.-J. Roes, E. M. van Mulligen, and J. A. Kors. Literature-based concept profiles for gene annotation: The issue of weighting. *International Journal of Medical Informatics*, 77(5):354 – 362, 2008.

[8] Lister Hill National Center for Biomedical Communications . The umls semantic groups. `http://semanticnetwork.nlm.nih.gov/SemGroups/`. Accessed: 2015-08-31.

[9] Oracle Corporation. MySQL, the world's most popular open source database. `https://www.mysql.com/`. Accessed: 2015-07-04.

[10] M. Otto, J. Thornton, and Bootstrap contributors. Bootstrap, the most popular html, css, and js framework for developing responsive, mobile first projects on the web. `http://www.getbootstrap.com`. Accessed: 2015-07-04.

[11] T. Otwell. Laravel php framework. `http://www.laravel.com`. Accessed: 2015-06-18.

[12] N. R. Smalheiser. Literature-based discovery: Beyond the abcs. *Journal of the American Society for Information Science and Technology*, 63(2):218–224, 2012.

[13] D. R. Swanson. Fish oil, raynaud's syndrome, and undiscovered public knowledge. *Perspectives in biology and medicine*, 30(1):7–18, 1986.

[14] D. R. Swanson, N. R. Smalheiser, and V. I. Torvik. Ranking indirect connections in literature-based discovery: The role of medical subject headings. *Journal of the American Society for Information Science and Technology*, 57(11):1427–1439, 2006.

[15] The Cytoscape Consoritum. Cytoscape.js, graph theory (a.k.a. network) library for analysis and visualisation. `https://cytoscape.github.io/cytoscape.js/`. Accessed: 2015-07-04.

[16] The Cytoscape Consortium. Cytoscape, network data integration, analysis, and visualization in a box. `http://www.cytoscape.org/`. Accessed: 2015-07-04.

[17] The jQuery Foundation. jQuery, write less, do more. `https://jquery.com/`. Accessed: 2015-07-04.

[18] The PHP Group. Php. `http://www.php.net`. Accessed: 2015-06-18.

[19] H. H. H. B. M. van Haagen, P. A. C. 't Hoen, A. Botelho Bovo, A. de Morre, E. M. van Mulligen, C. Chichester, J. A. Kors, J. T. den Dunnen, G.-J. B. van Ommen, S. M. van der Maarel, V. M. Kern, B. Mons, and M. J. Schuemie. Novel protein-protein interactions inferred from literature context. *PLoS ONE*, 4(11):e7894, 11 2009.

[20] H. H. H. B. M. van Haagen, P. A. C. 't Hoen, B. Mons, and E. A. Schultes. Generic information can retrieve known biological associations: Implications for biomedical knowledge discovery. *PLoS ONE*, 8(11):e78665, 11 2013.

[21] Wolfram Mathworld. Golden ratio conjugate. `http://mathworld.wolfram.com/GoldenRatioConjugate.html`. Accessed: 2015-08-17.