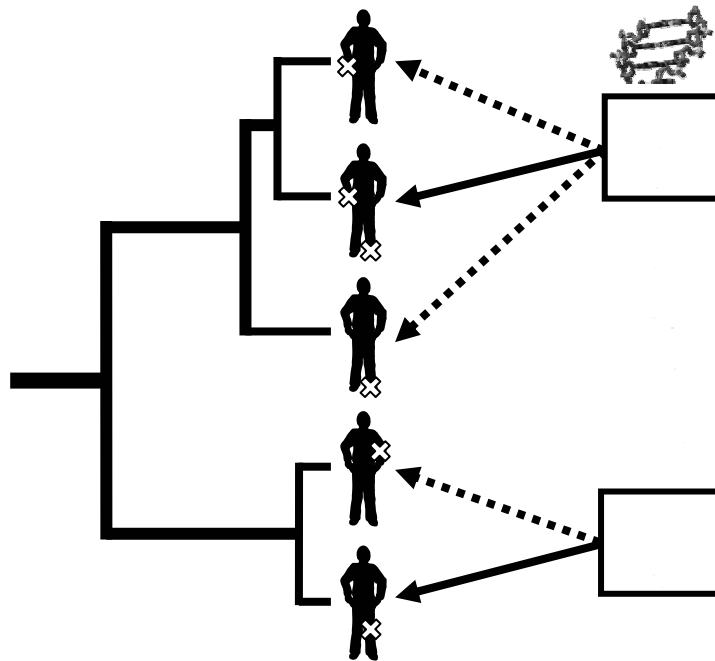


# Mining MIM

Phenotype clustering as source of candidate genes

Jorn Bruggeman  
Genome Informatics, Wageningen University  
November 2002 – April 2003



# Mining MIM

Phenotype clustering as source of candidate genes

Jorn Bruggeman  
Genome Informatics, Wageningen University  
November 2002 – April 2003

Supervision: Jack Leunissen



WAGENINGEN UNIVERSITY  
*University for Life Sciences*

# Table of contents

<b>Table of contents.....</b>	<b>3</b>
<b>Abstract.....</b>	<b>4</b>
<b>1 Introduction.....</b>	<b>5</b>
1.1 Why research phenotypic correlations? .....	5
1.2 Extracting phenotype features.....	6
1.3 Phenotype correlation.....	8
1.4 Phenotype clustering.....	8
1.5 Result evaluation.....	9
<b>2 Methods.....</b>	<b>11</b>
2.1 Feature extraction.....	11
2.2 Comparing phenotypes.....	17
2.3 Clustering phenotypes.....	19
2.4 Result evaluation.....	22
<b>3 Results.....</b>	<b>24</b>
3.1 Comparing feature matrices.....	24
3.2 Clustering tendency.....	26
3.3 Phenotype similarity vs. genotype similarity.....	28
3.4 Examples of ranking and trees.....	28
<b>4 Conclusions.....</b>	<b>34</b>
4.1 Phenotype sources.....	34
4.2 Hyperonym addition.....	35
4.3 Identification of phenotype similarity to find genes.....	35
<b>5 Discussion.....</b>	<b>36</b>
5.1 Consequences of dictionary-based feature extraction.....	36
5.2 Suggestions for improvement.....	37
5.3 General applicability.....	40
5.4 Concluding.....	41
<b>6 References.....</b>	<b>42</b>
<b>7 Acknowledgements.....</b>	<b>44</b>

## Abstract

Given a description of an anomalous phenotype, one may well be able find the gene(s) responsible for the anomaly in a list of genes associated with very similar phenotypes. Thus, a technique that identifies similar phenotypes could be exploited as a resource of candidate genes. In this project, we aim to develop such a technique.

As test case, we take the *Online Mendelian Inheritance in Man* (OMIM) database, which contains 14,000+ articles describing human inheritable traits and diseases. To determine the similarity between OMIM articles, we first characterize each article by the occurrence of a select set of (bio)medical terms. This set comprises the ‘anatomy’ and ‘disease’ categories of the *Medical Subject Headings* (MeSH) thesaurus. All MeSH entries are regarded as potential features of an OMIM article; per article, the value of any such feature is taken equal to the number of occurrences of the corresponding entry. In addition, we add for each matched entry its MeSH ancestors (i.e. the entries describing a superset of the entry) to the feature vector. This allows for closely related terms (sharing ancestors) to contribute to between-article similarity rather than diminishing it; in effect, the system has become term-relation-aware. Finally, feature vectors are corpus-size normalized, and feature values are weighed according to a scheme common in the field of information retrieval.

To determine similarity between phenotype feature vectors, we calculate their length-normalized correlation (i.e. the cosine of the angle between the vectors). With this similarity measure, we can list phenotypes similar to a given reference, ordered by proximity (i.e. ranking). In addition, we perform hierarchical clustering (UPGMA) on the full set of phenotypes.

To establish whether phenotype similarities indicate genotype similarities, we derive a genotype feature matrix for a set of  $\pm 1,000$  OMIM articles, selecting as features the *Gene Ontology* (GO) terms indirectly associated with the articles. Subsequently, we calculate the correlation between phenotype- and genotype proximity matrices, both pre- and post-clustering.

Result evaluation is difficult, as the set of phenotypes is large, and calculation of the significance of phenotype-genotype correlation coefficients is not feasible. However, both random sampling of article neighbors and qualitative evaluation of correlation coefficients indicate our system could prove a valuable resource of candidate genes. This is specifically the case for human phenotypes (i.e. OMIM); current application for other species may be troublesome due to low-quality phenotype descriptions, and lack of a dictionary with phenotype-describing terms.

# 1 Introduction

## 1.1 Why research phenotypic correlations?

At present, the human genome has been fully sequenced, and that of numerous other species, including cattle and crops, will rapidly follow. Yet, the greatest challenge awaits: to achieve understanding of how millions and millions of base pairs ultimately translate into a living, breathing individual. This requires complete knowledge of genes, but also of gene products (RNA, proteins), chemical pathways, and interactions on all levels, from the chemical level up to that of cells and organs. This knowledge will be crucial for numerous areas of research. It will aid industries through allowing for sophisticated genetic modification (based on the phenotype desired), and also medicine, particularly in the mapping, tracking and potentially curing of genetically inherited diseases. The potential uses of genome-to-phenotype data are numerous.

However, the many factors involved make genotype-to-phenotype mapping a difficult and laborious process. As a result, large parts of the genome will remain unlinked to phenotype characteristics for an indefinite amount of time. Vice versa, it will take a long time to tie the multitude of documented inheritable phenotype characteristics to their genetic underpinnings. This is for instance the case for the *Mendelian Inheritance in Man* (MIM) database, which consist of 14,000+ articles on inheritable traits: only about 8,000 are currently associated with a particular protein – let alone a gene. Many years will pass before all inheritable traits are associated with one or more corresponding genes.

Yet, the need to understand the genetic base of such phenotypes remains urgent. Patients with genetically inherited diseases are discovered every day. Any information on the cause of their disease (and potentially its cure) may prove vital to them. And – though more a matter of commerce than of life and death – indications of the (genetic) source of differences between breeding lines of cattle and crops could allow for more sophisticated and targeted genetic modification. In short, any means to link phenotypes to candidate genes will be most valuable.

But even if we do not know the genetic underpinnings of a particular phenotype, this does not imply we know nothing at all about its cause. Similar phenotype characteristics are likely to result from related genes. Such a relation may be close (e.g. when the gene products are different subunits of the same protein) or more distant (when the products play a role in the same chemical pathway, or fulfill similar functions), but it is likely to be there. This fact could be used as a valuable resource in the quest for phenotype-to-genotype mappings: one may be able to derive the genotypic root of an unlinked phenotype from the genes associated with closely correlated phenotypes. However, to apply this resource, we need a way to determine the correlation between phenotypes.

The aim of this project is to develop a technique to uncover phenotypic correlations, and to study their viability as a source of candidate genes. Note that we do not intend to produce a fully functional system to serve a particular need; rather, we aim to deliver a proof-of-concept that shows the possibilities of correlating phenotypes. As a test case, we use the MIM database (Hamosh et al., 2000): a catalog of 14,848 human genes and genetic disorders (as of April 2003). For numerous entries, this database provides the phenotypic description in a variety of formats: a set of keywords, an abstract, and a complete description. Thus, we can evaluate the performance of our system for different levels of description detail.

## 1.2 Extracting phenotype features

### 1.2.1 About features

Comparing items, whether phenotypes or others, requires knowledge of the features the items share (the intersection), and those in which they differ (the difference). Primarily, this requires the user to select the features he is interested in. At this point the following should be kept in mind: (1) Automated comparisons can only deal with quantitative or qualitative feature variables (e.g. ordinal, binary, Boolean). Features of other type (e.g. nominal) must be converted into a suitable one. (2) The resulting set of features must describe variation between phenotypes adequately and completely. Sufficient features should be included, yet one should try to exclude superfluous features. These add to the time required for feature comparison, but do not contribute any additional information. With these two requirements in hand, one can start to select features.

For phenotype comparisons, the feature set observable depends fully on the format of the phenotype descriptions. If these descriptions come as simple sets of keywords, phenotypes are defined solely by the occurrence of individual terms. This provides a natural starting point for feature definition. However, if descriptions are provided as ‘full-text’ (articles, or abstracts thereof), feature definition is more complex. ‘Full-text’ is composed of sentences, which contain phenotype-keywords, but in addition convey a meaning. Sentences may describe the relevance of terms (e.g. ‘this disease is clearly/likely/potentially/not related to *term*’), and relations between terms (‘*term1* only/sometimes/never occurs in combination with *term2*’). In essence, ‘full-text’ descriptions contain a superset of the information present in a keyword set. The design of a phenotype-processing system depends a great deal on whether the inclusion of such contextual information is desired.

### 1.2.2 Dictionary-based feature extraction

For sets of keywords, an intuitive and straightforward feature definition is ‘the presence/absence of a term’. In effect, every unique term can become a feature. The feature value equals 1 (*true*) if the term is present in the description and 0 (*false*) if it is not. Alternatively, one could set the feature value equal to the number of occurrences of the term. Focusing on term presence/occurrences has been termed a dictionary-based approach: one maintains a dictionary with  $x$  relevant terms, and determines per description whether (or how many times) each term matches. Thus, you obtain a feature vector of length  $x$  for every description. We can distinguish two different dictionary-based approaches: one that maintains an external dictionary (a predefined set of terms, independent from the contents of the descriptions), and one that maintains and updates an internal dictionary (every unique term found in the descriptions becomes a dictionary term). Like the dictionary-based approach in general, each comes with its advantages and disadvantages.

Clearly, any dictionary-based approach has a number of important drawbacks: (1) Different conjugations of a word stem (e.g. single vs. plural for nouns, present vs. past tense for verbs) are mapped to different features. (2) Different terms with the same meaning (synonyms) are mapped to different features. (3) ‘All terms are equal’; i.e. terms similar to the casual observer (e.g. hand/foot) will be deemed as different as complete unrelated terms (e.g. hand/diabetes). In addition, different dictionary-based approaches have their specific drawbacks: an internal dictionary requires the user to define term boundaries. This is troublesome because the most natural and usable term definition (a set of word-characters, e.g. {a-z, A-Z, 0-9, -, \_}, preceded and followed by a non-word character) causes the system to miss compound terms (a combination of two or more words). On the other hand, an

external dictionary can recognize compound terms if they are present in the dictionary, but fails to find unknown terms (describing the phenotype, but not in the dictionary).

### 1.2.3 Natural Language Processing

Dictionary-based systems for feature extraction are not limited to keyword sets. In fact, most systems that process full-text resources apply dictionaries. However, a dictionary-based system clearly cannot incorporate contextual information present in the text. Interpretation of context is the field of natural language processing (NLP), where a grammar parser or part-of-speech (POS) tagger is used to identify the role of sentence components. Given proper – and complex – post-processing filters and techniques, information obtained with NLP could identify key terms (both single-word and compound terms, without requiring dictionaries), as well as between-term relations, and term relevance.

However, the use of NLP techniques in biology and medicine is thus far fairly limited (for an overview see Blaschke et al., 2002). NLP has been used to identify gene names by context (Tanabe and Wilbur, 2002). However, this system defined typical gene context through training, but did not attempt to interpret contextual information (functions of genes, relations between genes).

The limited use of NLP should not come as a surprise. Aside from being relatively novel (and thus, unknown), NLP techniques require a lot of computer time. Identification of sentence components, word stems, etc. are laborious tasks, and far slower than simple dictionary-based term matching. This explains why most NLP applications either do not interpret context, or restrict themselves to small corpora. Ultimately, NLP techniques will a viable and effective technique in topic- and context interpretation of large corpora. Currently, however, designing an effective NLP system that both handles large amounts of text, and interprets context in detail is not feasible.

### 1.2.4 Our approach: a thesaurus-based dictionary

Because of the unfeasibility of NLP techniques, and our intention of handling both keyword- and full-text phenotype descriptions, our system applies a dictionary-based approach for feature extraction. We use an external dictionary, based on a subset of the *Medical Subject Headings* (MeSH) thesaurus (U.S. National Library of Medicine, 2003). This thesaurus has been developed by the editors of MedLine to provide a standardized vocabulary for annotation of (bio)medical articles.

The MeSH thesaurus is well suited to serve as the basis of a dictionary. MeSH entries list not only synonyms, but also conjugations (typically plural). By mapping these terms to one single feature, we eliminate both the synonym- and conjugation problems typical for dictionary-based methods. Thus, every phenotype feature represents a MeSH entry; feature values we take equal to the number of occurrences of the corresponding entry (or its synonyms/conjugations).

In addition, we use MeSH references to incorporate information on relationships between entries. The MeSH thesaurus contains many cross references, both flat ('see also ...') and hierarchical ('is a superset/subset of ...'). We use the latter type of references to deal with relationships between terms: if an article matches a particular MeSH entry, we assume all terms describing supersets of that entry ('ancestors') to bear relevance too<sup>1</sup>. The degree of 'inherited' relevance is strictly defined, and added to the feature vector of the article. Thus,

---

<sup>1</sup> 'Ancestors' include an entry's 'parents', as well as its 'grand-parents', great-grandparents', etc. Thus, most MeSH entries have multiple ancestors. In addition, terms with multiple parents are quite common. E.g. 'blood' is a child of both 'body fluids' and 'hemic and immune systems'.

agreement between phenotypes no longer requires explicit MeSH term matches; if phenotypes match closely related terms (i.e. with a common ancestor), this will increase their similarity.

### 1.2.5 Feature refinement

Obviously, any dictionary-based approach that counts term occurrences (as does ours) produces feature vectors with values that depend on corpus-size: larger descriptions match more terms. Yet, phenotypes should not be defined by the number of words an author decided to spend on their behalf; if that were the case, future description addendums would drastically change the phenotype as represented by our feature vector. To avoid this, we normalize the feature values by dividing by some measure of description size.

In addition, counting term occurrences instead of merely detecting their presence (*true* or *false*), has more subtle consequences. Every term match increments the corresponding value in the feature vector with 1, independent of the current vector state. This implies that every match is of equal importance: the 100<sup>th</sup> occurrence of a term is rated as significant as the 1<sup>st</sup>. This is counter-intuitive: one would prefer to rate the one-time-only occurrence of term *a* higher than the (>1)<sup>th</sup> occurrence of term *b* (while preserving *b*'s advantage over *a*, naturally). We obtain this effect by weighting feature values according to a strictly defined function (placing more weight on low, and less on high feature values).

## 1.3 Phenotype correlation

Given a method for feature extraction, we can obtain one feature vector per phenotype description. The feature vectors describe the importance of MeSH terms (and ancestors), with values ranging between 0 and 1 (this maximum is due to corpus-size normalization). Given the steps taken for feature refinement, we can treat the feature values as ratio-scaled variables, that is: the ratio between two such variables is meaningful. This qualification provides a starting point for selection of a measure to describe similarity between phenotype vectors.

In literature, numerous similarity measures – and the opposite: distance measures – have been described that deal with ratio-scaled variable vectors (e.g. Theodoridis and Koutroumbas, 1999). There are little theoretical grounds on which to select a measure. Some measures have geometrical interpretations, such as the metric distance between vector tips in multidimensional space (Euclidian distance), or (a function of) the angle between vectors (e.g. the in-product of length-normalized vectors). Others are mere ad-hoc constructions designed to deal with a particular problem. Ultimately, however, the choice of a similarity- or distance measure is a subjective one.

We have tested various similarity- and distance measures, and all rendered qualitatively similar results. Hence, we simply settle on a measure that is commonly used (e.g. Wilbur and Yang, 1996) and – consequently – well-described: the in-product of the length-normalized feature vectors (or: the cosine of the angle between the vectors).

## 1.4 Phenotype clustering

### 1.4.1 Clustering vs. ranking

With feature vectors and a similarity measure in hand, one can begin to compare phenotypes. For numerous goals, this suffices: for any given phenotype, one can find and rank similar phenotypes. However, when dealing with large groups of phenotypes (e.g MIM), it may pay to also perform cluster analysis. In this process, all phenotypes are assigned to classes, based on their between-phenotype similarities.

Compared to ranking, clustering offers the advantage of taking all phenotypes into account simultaneously. If phenotype A is somewhat similar to B, but B is far more similar to



a group very different from A, B will cluster with the latter group, not with A. In practice, this is for instance the case with a variant of Parkinson's disease. The description of this variant is somewhat similar to that of Alzheimer's disease (and appears high in the Alzheimer's disease ranking), but ultimately clusters around other Parkinson variants. In effect, its rated similarity to Alzheimer's decreases.

### **1.4.2 Clustering tendency**

Before actually clustering the phenotype feature vectors, we establish whether the vectors show a tendency to cluster, i.e. whether the phenotypes indeed form distinct groups. To ascertain this, we visualize the phenotype feature vectors in 2-dimensional space. This involves remapping the phenotypes from  $\pm 5,000$ -dimensional space (the number of features per phenotype) to 2-dimensional space. We tested a variety of common multidimensional scaling techniques (including Principal Component Analysis, classic multidimensional scaling, Independent Component Analysis), and all rendered qualitatively similar results: no evident clusters are discernable. This does not imply clustering is impossible; however, it does reduce the number of suitable clustering technique.

### **1.4.3 Clustering technique**

Many popular clustering techniques, including most scalable ones, require the user to specify upfront the number of clusters in the data. As our phenotypes show no clear clustering tendency, the choice of the number of clusters would be an extremely subjective one. Instead, we resorted to hierarchical clustering, which produces a number of different clusterings, each cluster part of a cluster at a higher level. This is similar to the evolutionary trees commonly found in biology.

Different methods of hierarchical clustering exist. One can distinguish between agglomerative methods (where initially, every item is a cluster, and clusters are merged), and divisive methods (where initially, one cluster contains all items, and cluster are split up). However, all have in common that they require significant computer time, as they must evaluate all items simultaneously; unlike non-hierarchical methods, one cannot start of with a random subset of items, cluster those, and then appoint remaining items to the clusters formed. This poses problems in our case, because the number of items is very large (14,000+ phenotypes). To reduce the required amount of computer time to a minimum, we settled for an uncomplicated – and hence, fast – method: the Unweighted Pair Group Method with Averages algorithm (Sneath and Sokal, 1973). This is an agglomerative algorithm, which accepts a matrix describing between-phenotype distances (computed from between-phenotype similarities), and produces a hierarchy of clusterings.

## **1.5 Result evaluation**

### **1.5.1 Linkage between phenotype and genotype**

Our system defines similarities between (groups of) phenotypes, both through ranking and clustering. However, for most practical applications, we are not interested in phenotype similarities as such, but rather in their viability as source of candidate genes. Therefore, the true performance of the system is characterized best by a measure specifying how well the calculated phenotype similarities translate into genotype similarities. To obtain such a measure, we repeat the ranking and clustering of OMIM articles with features of the genotype rather than the phenotype. The resulting similarity matrices and clusterings are then compared with those obtained with phenotype clustering.

### **1.5.2 Ranking and clustering genotypes**

To obtain genotype<sup>1</sup> features, we select only the OMIM articles that are mapped to one of more proteins (in the SWISS-PROT/TrEMBL databases), and extract the properties of these proteins from the Gene Ontology (GO) database (Gene Ontology Consortium, 2000). This database was designed to provide a controlled vocabulary for describing functions and characteristics of genes and proteins across species. Using GO, we obtain a set of genotype features for every mapped OMIM article (every feature represents a GO entry).

Similar to the MeSH thesaurus, GO incorporates all entries in a hierarchical structure. As with the phenotype feature vectors, we use this information on entry relations to supplement the feature vectors: ancestors of a relevant GO entry are added to the feature vector. This is most needed, as the average number of ‘direct’ genotype features per OMIM article is low (2.8). Such low numbers of features would result in very little overlap between genotypes, and – consequently – crude ranking/clustering.

Given the genotype feature vectors, ranking and clustering is done like before: similarities equal the in-product of length-normalized vectors, and clustering is done with the UPGMA algorithm.

### **1.5.3 Comparing genotype- and phenotype results**

To compare phenotype-based rankings (i.e. similarity matrices) and clusterings (i.e. trees) with genotype-based ones, we use the CoPhenetic Correlation Coefficient (CPCC), a statistic that originated in phylogeny research (Rolph, 1970). The CPCC is a measure of correlation between two similarity- or distance matrices, and can therefore directly be applied to the ranking results. To apply it to hierarchical clusterings, the trees are first described in cophenetic matrices. These are distance matrices, the distance between nodes equal to their distance (summed branch lengths) in the tree.

---

<sup>1</sup> Note that we use the term genotype loosely; our genotype features are derived from associated proteins, rather than genes.

## 2 Methods

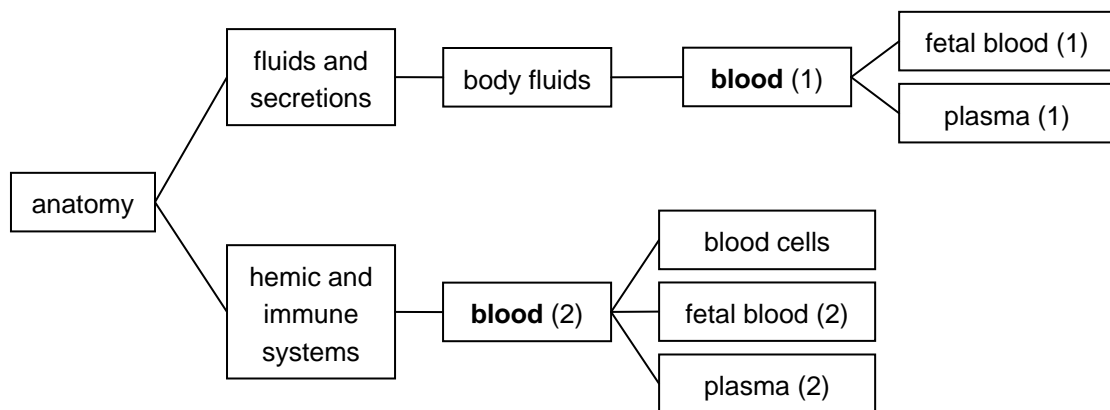
### 2.1 Feature extraction

#### 2.1.1 Building the dictionary

We extract phenotype features using an external dictionary that is based on the Medical Subject Headings (MeSH) thesaurus. This thesaurus has been developed by the editors of MedLine to provide a standardized vocabulary for annotation of (bio)medical articles. It currently contains 21,973 entries, 5,391 of which pertain to anatomy and disease (and hence are relevant to phenotypes described in OMIM articles).

To understand the MeSH' structure, one must know the vocabulary used to refer to the properties of entries. A MeSH entry is termed a descriptor. Every descriptor has a unique identifier (UI) appointed to it (a 'D' followed by six digits). Descriptors describe one or more concepts, which in turn are described by one or more terms. Terms include synonyms and conjugations (typically plural). Both descriptors and concepts are abstract concepts, i.e. they do not have a direct association with a particular description as such; rather, every descriptor has a 'preferred' concept, and every concept has a 'preferred' term.

The hierarchical links provided by MeSH are between descriptors only. These links are incorporated by tree numbers, describing the descriptor's position in the global MeSH tree. Part of this tree is shown in figure 2.1. Tree numbers start with a topic identifier (e.g. 'A' for anatomy, 'B' for organisms, 'C' for diseases), followed by 2 digits specifying the node below the topic, and, for each deeper level (if any), a dot followed by 3 digits. For instance, if a descriptor has tree number A12.231.123, it tells us the descriptor has A12.231 as a parent, A12 as grandparent, and the anatomy (A) root as great-grandparent. Descriptors can – and often do – have multiple tree numbers associated with them, and can therefore occupy multiple positions in the tree (figure 2.1, descriptors 'blood', 'fetal blood' and 'plasma'). Note that the tree number format permits such descriptors to have different (sets of) children at each tree position (figure 2.1, descriptor 'blood').



**Figure 2.1.** Descriptor 'blood' in the MeSH tree. Only nodes directly related to 'blood' (children, parents, grandparents, etc.) are shown; other children of higher level nodes are omitted.

The MeSH database can be obtained from <http://www.nlm.nih.gov/mesh/> as one single file in XML format (222 MB). In the XML, every MeSH entry is available as a 'DescriptorRecord' node below the 'DescriptorRecordSet' root node. Using a SAX-based XML parser (Perl package XML::Parser), we extract the following properties from each DescriptorRecord node:

- The descriptor's unique identifier (the contents of XML node /DescriptorUI)
- All terms associated with any of the descriptor's concepts (the contents of every XML node /ConceptList/Concept/TermList/Term/String)
- All tree numbers associated with the descriptor (the contents of every XML node /TreeNumberList/TreeNumber).

Descriptors are included in the dictionary only if one or more of the tree numbers start with 'A' (anatomy) or 'C' (disease). A low number of descriptors (<100) have no tree number association; these are excluded. Relevant descriptors are stored as a {descriptor UI, associated terms} combinations (to build the dictionary), and separately as {descriptor UI, associated tree numbers} combinations (to retain hierarchical relations between MeSH terms). For the dictionary, a subset of non-informative A/C MeSH terms (e.g. syndrome, disease, cells) is excluded.

### **2.1.2 Processing OMIM**

The OMIM database can be obtained from <http://www.ncbi.nlm.nih.gov/omim/> as one single file in plain text format (81 MB). This file is in essence a flat database: records are listed consecutively, as are the fields within each record. Empty fields are not included. The start of a record is indicated by a line containing only the string '\*RECORD\*'. Fields are preceded by a '\*FIELD xx\*' header, where 'xx' indicates the type of field. A listing of fields used in OMIM records is available in table 2.1. We selected the following fields as relevant to the phenotype: the title (TI), the full text (TX), the abstract (mini-MIM, MN) and the keyword list (Clinical Synopsis, CS).

**Table 2.1.** Fields used in OMIM records. The percentage of OMIM records (total: 14,484) containing the field type is given in the ‘presence’ column.

Abbreviation	Description	Presence (%)
NO	Record identifier (six digits), preceded by ‘^’ if the record has been moved to (incorporated in) another record. The first digit identifies the category of inheritance (autosomal vs. X-linked vs. Y-linked vs. mitochondrial and dominant vs. recessive).	100.0
TI	Record number and title (i.e. trait name), often preceded by a disease status identifier (‘*’/’#’). The trait name may be followed by trait synonyms, separated by ‘;’.	100.0
TX	Full text trait description, structured with headers, sub-headers, etc.	100.0
MN	Abstract of full text trait description; updated less often than the full text field.	1.3
CS	Clinical Synopsis, i.e. a listing of relevant keywords; usually structured (keywords are grouped using headers and – sometimes – sub-headers).	30.2
RF	References to relevant literature.	95.8
SA	List of references that describe the trait in detail (pointers to RF references).	14.2
CD	Record creation date/creator name.	100.0
CN	Record contributors (lists contributor name/date of last update combinations).	47.8
ED	Edit history of the record (lists editor name/date combinations).	100.0

To extract phenotype features, we parse OMIM on a per-record basis. Each of the 14,848 records is first parsed to identify fields, and for each relevant field (i.e. TI/TX/MN/CS) we match against all dictionary entries. Per field, we perform global matching to count term occurrences: the match count per dictionary entry is increased every time one of the entry terms (thus incl. synonyms/conjugations) is encountered. Matching is case-insensitive to prevent problems with capitalized headers (typically in the Clinical Synopsis).

We store results in sparse matrix format, one row per OMIM record (identified by the record’s NO field), one column per dictionary entry (identified by MeSH descriptor unique identifier). Matrix values denote the number of entry occurrences. We maintain different feature matrices for each type of field. Thus, we end up with 4 matrices (for the TI, TX, MN and CS fields). Separate field feature matrices are used to allow for future merging with different weighing factors per field type; one could for instance decide to attribute more weight to title/keyword/abstract matches than to full text (TX) matches.

We ultimately evaluate the performance of our system with two feature matrices: the keyword (CS) feature matrix and an ‘ultimate’ feature matrix that sums all 4 source matrices (weighted equal: 1). This allows us to determine the system’s sensitivity to description size; the keyword (CS) field contains far less information on the phenotype (matches far less dictionary entries) than the 4 fields combined. The abstract (MN) feature matrix would be a suitable intermediate category, but is not used because of the low presence of the MN field (1.3 %).

### 2.1.3 Incorporation of MeSH hierarchy

To incorporate information on term relationships, we assume an article matching a particular MeSH term  $A$  also bears relevance to the hyperonyms<sup>1</sup> of  $A$ . Hyperonyms are terms describing supersets of  $A$ , ‘parents’, if you like. We take the degree of relevance  $r_p$  of any such parent terms to follow from (1) the relevance of  $A$  ( $r_A$ ), and (2) the number of children of the parent ( $n_{c,p}$ ):

$$r_p = \frac{1}{n_{c,p}} r_A$$

In other words: if a parent term describes a class of many specialized terms (i.e. a broad category), the link between parent and child is weak; if the parent term describes a class with few specialized terms, the link is strong. Note that the above equation implies that if all children of a parent have relevance  $r$ , the parent also automatically receives relevance  $r$ .

The above relationship extends through the entire MeSH hierarchy upwards: for any relevant (i.e. matched) term, all ancestors (parent, grandparents, great-grandparents, etc.) bear relevance too. This is a logical consequence of the above assumption:

$$\begin{aligned} \text{base term relevance:} & \quad r_A \\ \text{parent relevance:} & \quad r_{p1} = r_A \cdot \frac{1}{n_{c,p1}} \\ \text{grandparent relevance:} & \quad r_{p2} = r_{p1} \cdot \frac{1}{n_{c,p2}} = r_A \cdot \frac{1}{n_{c,p1}} \cdot \frac{1}{n_{c,p2}} \end{aligned}$$

Thus, given the MeSH hierarchical links and the relevance of a base term, we can calculate the relevance of the ancestors, and use these to extend the original feature vector.

For our system, we let every matched term  $A$  contribute ancestors, taking the relevance of  $A$  equal to the number of occurrences of  $A$  (as defined in the feature matrix). An exception to this rule is made for matched terms that occupy multiple positions in the tree (e.g. the term ‘blood’ appears as a child of both ‘body fluids’ and ‘hemic and immune systems’). Terms with multiple positions are quite common. Intuitively, different tree positions of a single term could be qualified as different contexts in which the term is usable. If such terms are matched, we cannot determine their context (i.e. select the relevant tree position). Therefore, we divide the term’s base relevance (number of occurrences) by the number of tree positions (‘contexts’), then add the ancestors of each position they occupy.

To clarify this rule, let’s take a few MeSH nodes from figure 2.1. If ‘blood’ is matched, half of its relevance is transferred to parent ‘body fluids’, and half to ‘hemic and immune systems’ (because blood can occur in these two ‘contexts’). Similarly, if ‘plasma’ is matched, half of its relevance is transferred to ‘blood (1)’, and half to ‘blood (2)’; these subsequently transfer their relevance to ‘body fluids’ and ‘hemic and immune systems’, respectively. On the other hand, if ‘blood cells’ is matched, its full relevance is transferred to ‘blood (2)’, and thus, indirectly to ‘hemic and immune systems’ alone. Note that in each of these examples,

---

<sup>1</sup> Parent MeSH terms (one level up the hierarchy) of any child term are not always hyperonyms in the true sense of the word; they can also describe a structure of which the children are part. To clarify this: the MeSH term ‘hand’ has child ‘finger’. Yet, ‘hand’ is not a hyperonym of ‘finger’, because a finger is not some sort of hand. On the other hand, the term ‘diabetes mellitus’ is a true hyperonym of ‘diabetes mellitus, insulin-dependent’, as the latter is indeed some sort of the ‘diabetes mellitus’. For the sake of simplicity, however, we will refer to any MeSH ancestor (parent, grand-parent, etc.) of term  $a$  as hyperonym of  $a$ .

the transferred relevance  $r_A$  is still divided by the number of children of the parent ( $n_{c,p}$ ) to arrive at  $r_p$ , the parent relevance.

Ancestor relevance values inferred from true term matches are used to increment the original feature vectors. Hence, an entry's feature value now becomes the sum of its original number of matches, and the ancestral relevance inferred from matched offspring entries.

To test our method of hierarchy incorporation, we evaluate the system's performance both with and without inclusion of ancestors in the feature matrix. This is done for both the keyword-based (CS) feature matrix and the full (TI/CS/MN/TX) feature matrix, thus bringing the total number of matrices evaluated to 4.

#### 2.1.4 Corpus size normalization

The feature vectors characterize the phenotype of OMIM articles. These characterizations should not depend on the article's size, only on its contents. Yet, the current feature vectors contain (linear combinations of) the number of term occurrences. As that number depends strongly on article size (larger articles match more terms), so will the feature vectors.

To remove this dependency, we normalize the feature vectors to corpus size (i.e. article size). This is done by dividing the feature values by a measure of article size, on a per-article basis. Different size measures may be used, e.g. the total number of words of the article, the total number of matches of the article or the maximum number of matches between MeSH entries. Likely, all will render qualitatively similar results. We follow the approach of Wilbur (Wilbur and Yang, 1996) in dividing by the maximum number of matches between entries. This has the advantage of restricting all feature values between 0 and 1; knowledge of this range is helpful for further feature value refinement.

#### 2.1.5 Weighing term relevance per article

With dictionary-based feature extraction, one could rationalize both a focus on the presence/absence of entries (with binary feature variables), and a focus on the number of term occurrences. Binary feature variables pose a problem because they attribute as much weight to frequently mentioned entries (in one article) as to entries that occur only once. This clearly not conforms to the nature of the phenotype description. On the other hand, the focus on number of entry occurrences will neglect little mentioned terms if one or more other terms appear very frequently. In those circumstances, the full spectrum of entries becomes irrelevant, and common entries dominate.

One would prefer to strike a balance between a feature variable based on presence/absence, and one based on the number of occurrences. Taking the number of occurrences as base feature variable, one can obtain this effect by relatively increasing low feature values, and/or relatively decreasing high ones. Thus, the feature variable is no longer proportional to the number of occurrences.

Different functions may be used to change the balance between common and rare features; the main requirements being that the function  $f(x)$  increases as  $x$  (the feature value) increases, while the ratio  $f(x)/x$  decreases. In addition, easy interpretation of the modified feature values calls for  $f(x)$  to equal 0 at  $x=0$ . Formalized:

$$\begin{aligned}
 x = 0 : & \quad f(x) = 0 \\
 0 \leq x \leq 1 : & \quad \frac{df(x)}{dx} > 0, \quad \frac{d(f(x)/x)}{dx} < 0
 \end{aligned}$$

Note that any feature value  $x$  cannot exceed 1 due to the corpus size normalization technique. Hence, the function requirements are relevant only for the  $x$  range  $[0,1]$ . Many different functions fulfill these requirements. One likely candidate is the hyperbole<sup>1</sup>:

$$f(x) = a \frac{x}{b+x}$$

This function is continuous for  $x$  range  $(-b, \infty)$ , and saturates to  $a$  as  $x$  increases. The value of  $b$  determines the rate of saturation. This rate corresponds to the advantage common features have over rare ones: low values of  $b$  result in fast saturation (and a small advantage for common features), high values of  $b$  result in slow saturation (and a large advantage). We have experimented with the functional response with  $a=2$  and  $b=1$ . For these values,  $f(x)$  is plotted in figure 2.2; its relative effect on feature values is plotted in figure 2.3. With these values,  $f(1)=1$ ; all feature values below 1 increase. Low feature values experience a higher increase than high ones.

Wilbur (Wilbur and Yang, 1996) weights feature values with a more simple function:

$$\begin{aligned} x = 0: & \quad f(x) = 0 \\ x > 0: & \quad f(x) = 0.5 + 0.5x \end{aligned}$$

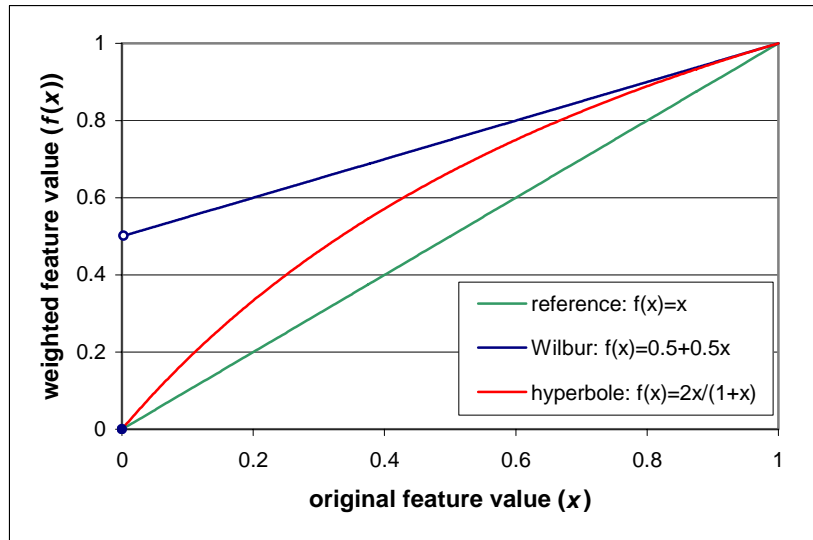
This function is plotted in figure 2.2; its relative effect on feature values if plotted in figure 2.3. Recall that corpus size normalization caused feature values to range between 0 and 1; the result of  $f(x)$  will therefore also range between 0 and 1 (or, more accurately:  $\{0\} \cup [0.5,1]$ ). A drawback of this function is the abrupt change in weighted feature value as  $x$  exceeds 0. Thus, features with even the most minute relevance (e.g. indirect matches through ancestral MeSH relations, with direct matched offspring much deeper in the tree) will be rated 0.5; already half of the maximum relevance possible!

Even though the linear weighing function has its drawback, we ultimately opt for this approach of feature weighing. We do this simply because the article by Wilbur offers us a frame of reference. Unlike the hyperbole weighing function, this approach has already been tested and found to perform adequately.

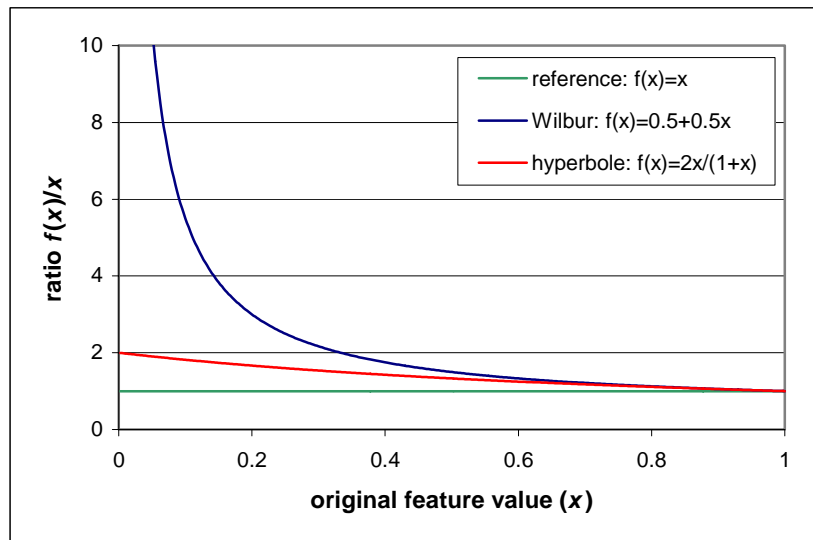
---

<sup>1</sup> for biologists: the functional response for the modeling of predation and food consumption; for chemists: the Michaelis-Menten equation for single-substrate enzyme kinetics





**Figure 2.2.** Functions suitable for re-evaluation of feature values so that more weight is placed on low-valued features. The green line represents unweighed feature values. Note that the function used by Wilbur (blue line) is discontinuous at  $x=0$ .



**Figure 2.3.** Relative effect of feature weighing with two different weighing functions (also see figure 2.1). Graphs plotted show the weighed feature value relative to the original feature value; for the reference non-weighing function (green line) these values coincide, causing it to equal 1 for each  $x$ .

## 2.2 Comparing phenotypes

### 2.2.1 Proximity measures

By normalizing and weighting the feature values, we obtain feature variables that can be viewed as ratio-scaled. (This means that the ratio between feature values is meaningful; one may say a feature with value 1 is twice as important as a feature with value 0.5). To calculate the similarity between two feature vectors, we can thus use any of the literally hundreds of proximity measures suited for ratio-scaled variables (for an overview, see for instance Theodoridis and Koutroumbas, 1999).

We have experimented with different proximity measures, described in detail below. Note that  $x_i$  corresponds to the  $i^{\text{th}}$  element of vector  $\mathbf{x}$ , and  $l$  to the number of feature variables (i.e. the number of elements in vectors  $\mathbf{x}$  and  $\mathbf{y}$ ).  $C_A$  refers to the number of elements in set  $A$ , i.e. its cardinality.

We tested the following distance measures:

- the Euclidian distance:

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^l (x_i - y_i)^2$$

The Euclidian distance corresponds to the geometrical distance between the vector tips in multidimensional space. Range:  $[0, \infty)$ .

- the Manhattan distance:

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^l |x_i - y_i|$$

The Manhattan distance is quite similar to the Euclidian, but is less sensitive to large feature differences between vectors. A feature's contribution is proportional to its difference between vectors ( $\Delta$ ), whereas this contribution is proportional to  $\Delta^2$  for the Euclidian. Range:  $[0, \infty)$ .

And the following similarity measures:

- the Tanimoto measure (Tanimoto, 1958):

$$s(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 - \mathbf{x}^T \mathbf{y}} = \frac{\sum_{i=1}^l x_i y_i}{\sum_{i=1}^l (x_i^2 + y_i^2 - x_i y_i)}$$

For binary feature vectors (valued 0 or 1) the measure corresponds to  $C_{x \cap y} / C_{x \cup y}$ . Range:  $[0, 1]$  given  $x_i \geq 0$ .

- A custom measure that is comparable with the ratio between intersection and union of  $x$  and  $y$  as used with binary feature variables:

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^l \min(x_i, y_i)}{\sum_{i=1}^l \max(x_i, y_i)}$$

For binary feature vectors, this measure corresponds to  $C_{x \cap y} / C_{x \cup y}$ . Range:  $[0, 1]$  given  $x_i \geq 0$ .

- length-normalized correlation:

$$s(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^T \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} = \frac{\sum_{i=1}^l x_i y_i}{\sqrt{\sum_{i=1}^l x_i^2} \sqrt{\sum_{i=1}^l y_i^2}}$$

This measure equals the cosine of the angle between  $\mathbf{x}$  and  $\mathbf{y}$ , and depends therefore exclusively on vector direction, not on vector length. Range:  $[0,1]$  given  $x_i \geq 0$  (if not for this restriction on  $x_i$  range, the measure would – like any cosine – range between -1 and 1).

Though all above proximity measures naturally render quantitatively different results, they perform quite similar qualitatively. We performed ranking (i.e. given a reference vector, listing all other phenotype vectors in order of proximity) with two OMIM reference articles (108300: Stickler syndrome I and 104300: Alzheimer disease). The rankings produced were different, but a large percentage of the top 50 articles overlapped. It is impossible to make a truly objective choice in proximity measures; ultimately, it is more or less a matter of taste. We have opted for the length-normalized correlation as proximity measure; this measure is commonly used (e.g. Wilbur and Yang, 1996), and therefore well-documented.

## 2.3 Clustering phenotypes

### 2.3.1 Clustering tendency: multidimensional scaling

Before blindly stumbling into the domain of clustering, one would be well advised to first ascertain whether the data (in this case the phenotype vectors) show a tendency to cluster. This clustering tendency characteristic can then be used to decide on the usefulness of clustering or on the clustering technique to apply.

One of the most intuitive ways of establishing clustering tendency is to visualize the multidimensional data (in this case the number of dimensions equals the number of features: 5,000+) in two- or three-dimensional space. This provides the user with a clear – though subjective – means of judging the tendency of the data to group: one might see the data scattered apparently random in low-dimensional space (no obvious clustering tendency), or grouped, with large space between groups (clear clustering tendency).

Many different techniques exist that map high-dimensional data to low-dimensional space; such techniques are referred to as multi-dimensional scaling techniques. We perform multidimensional scaling with various libraries available for the open source package R (R Foundation for Statistical Computing, 2002). We have tested the following methods:

- Principal Component Analysis (PCA) (R package: mva, function: prcomp)

PCA finds a new coordinate system for multivariate data such that the first coordinate has maximal variance; the second coordinate has maximal variance subject to being orthogonal to the first, etc.

PCA has been described as a technique suited mainly for normally distributed data (i.e. values of a particular feature follow a normal distribution). As clusterable data is typically not normally distributed, one might argue PCA analysis will often provide little insight in clustering tendency.

- projection pursuit (R package: fastICA, function: fastICA)

Projection pursuit is a technique for finding ‘interesting’ directions in multi-dimensional datasets. ‘Interesting’ directions here means the directions which show the least random distribution. For our purposes, we use the R package fastICA, which finds the directions with the least Gaussian distribution. More information can be found in Hyvärinen and Oja, 2000. It is interesting to note that this method of projection pursuit initializes by performing PCA analysis; this is done simply to normalize (‘whiten’) the data.

- Sammon mapping (R package: multiv, function: sammon)

Sammon mapping finds a new, reduced-dimensionality, coordinate system for multivariate data such that an error criterion between distances in the given space, and distances in the result space, is minimized.

- classic multidimensional scaling (R package: mva, function: cmdscale)

Classic multidimensional scaling takes a set of dissimilarities and returns a set of points such that the distances between the points are approximately equal to the dissimilarities.

Both PCA and projection pursuit are linear scaling methods: new (lower-dimensional) coordinates for any data point are linear combinations of its original coordinates. Sammon mapping and classic multidimensional scaling are non-linear methods: these first calculate the Euclidian distance between data points (i.e. build a distance matrix), then find those lower-dimensional coordinates that minimize the difference between original and new (Euclidian) distances, according to a certain error criterion.

All scaling methods described above have a significant drawback in addition to being computationally intensive: they require a lot of memory. Memory consumption is mainly an effect of algorithm implementation. We used ready-to-go R routines, rather than hand-crafted optimized code. This has the advantage of speed: no coding or debugging required. However, all packages require R to have the entire feature matrix (14,000+ × 5,000+ items) loaded in main memory. With memory-optimized programming, this would require appr. 280 MB (with float variables), but R easily takes 10 times as much, likely due to inefficient variable storage. As a result, lack of memory (Windows 2000/XP have a theoretical memory limit of 2 or 3 GB for applications, ±1.5 GB in practice) prevented inclusion of all OMIM articles in our multidimensional scaling analysis. Instead, we settled for a sample of articles: those with a clinical synopsis (CS) field present. Note that this sample is not necessarily representative: the articles with CS are those that are best documented, and therefore contain most information on the phenotype.

### **2.3.2 Clustering: Unweighted Pair Group Method with Averages (UPGMA)**

The various methods for establishing clustering tendency produced quantitatively similar results: no evident clusters are discernable. This does not imply clustering is impossible, yet it does exclude any clustering method that requires a specified number of clusters upfront; as we cannot discern true clusters in two-dimensional space, the choice of the number of clusters would be extremely subjective.

Instead, we resort to a hierarchical clustering method, which produces a set of nested clusterings rather than one single clustering. To limit computing time and memory consumption, we opt for the one of the simplest techniques: the UPGMA agglomerative clustering algorithm. This algorithm starts by taking each node (OMIM article) as a single cluster, and then proceeds by merging clusters until one ‘root’ cluster is left. Its output is a so-called dendrogram: a tree of nested clusters. In runtime, the UPGMA algorithm typically maintains the following variables:

- a distance matrix  $\mathbf{D}$  specifying distances between clusters. Note that  $\mathbf{D}$  is symmetric:  $\mathbf{D}(i,j)=\mathbf{D}(j,i)$ . Thus, one can suffice with only the upper- or lower triangular.
- a vector  $\mathbf{n}$  with the number of end nodes per cluster (i.e. the cardinality of the cluster).
- a vector  $\mathbf{p}$  with the distance to the tree tips (or end nodes) per cluster.

Initially, every node (OMIM article) is a single cluster. Therefore  $\mathbf{D}(i,j)$  must initially contain the distances between articles. Though we define article similarity (length-normalized correlation)  $s(i,j)$ , rather than distance  $d(i,j)$ , distances are easily calculated given  $s(i,j)$  ranges between 0 and 1:  $d(i,j)=1-s(i,j)$ . Logically, for each cluster  $C_i$ ,  $\mathbf{n}(C_i)=1$  and  $\mathbf{p}(C_i)=0$  initially.

Given the initial  $\mathbf{D}$ ,  $\mathbf{n}$  and  $\mathbf{p}$ , the UPGMA algorithm typically proceeds along the following steps:

1. Find the lowest  $\mathbf{D}(C_i,C_j)$ , i.e. the two clusters  $C_i$  and  $C_j$  in the distance matrix with the shortest distance (or highest correlation). These clusters are to be merged into one new cluster  $C_q$ .
2. Append a new row and column to the distance matrix  $\mathbf{D}$  for the merged cluster  $C_q$ . Distances from  $C_q$  to every other existing cluster  $C_s$  are calculated according to the following formula:

$$\mathbf{D}(C_q, C_s) = \frac{\mathbf{n}(C_i)}{\mathbf{n}(C_i) + \mathbf{n}(C_j)} \mathbf{D}(C_i, C_s) + \frac{\mathbf{n}(C_j)}{\mathbf{n}(C_i) + \mathbf{n}(C_j)} \mathbf{D}(C_j, C_s)$$

3. Add entries to vectors  $\mathbf{p}$  and  $\mathbf{n}$  for the merged cluster  $C_q$ :

$$\mathbf{p}(C_q) = \frac{\mathbf{D}(C_i, C_j)}{2}$$

$$\mathbf{n}(C_q) = \mathbf{n}(C_i) + \mathbf{n}(C_j)$$

Note that given  $\mathbf{p}$ , the length  $b$  of the branches between  $C_i/C_j$  and  $C_q$  is automatically defined:

$$b(C_i, C_q) = \mathbf{p}(C_q) - \mathbf{p}(C_i)$$

$$b(C_j, C_q) = \mathbf{p}(C_q) - \mathbf{p}(C_j)$$

4. Register the merge event so that we later can recover (1) the clusters involved ( $C_i$ ,  $C_j$ ,  $C_q$ ), and (2) the distances between merged clusters and the combining cluster.
5. Remove  $C_i$  and  $C_j$  from  $\mathbf{D}$ ,  $\mathbf{n}$  and  $\mathbf{p}$ .
6. If the number of remaining clusters exceeds 1, restart at 1.

We used a modified version of the UPGMA algorithm of PHYLIP (Felsenstein, 1993). PHYLIP is a phylogeny software package of which the C source code is freely available. To

reduce memory consumption and increase speed, we implemented several changes in the source code, the most important being:

- We store distances as float (4 bytes) rather than as double (8 bytes).
- We disable in-memory storage of the number of replica's (integer, 4 bytes) per distance (PHYLIP reads and stores distance matrices with replica's, but ignores replica values while iterating).
- We keep the lower triangular of the distance matrix in memory, rather than the full matrix (PHYLIP supports lower triangular matrices, but still stores the full matrix in memory).
- We disable support for random input order of nodes (with PHYLIP's support for random input order, every distance lookup first requires a `index-to-distance_index` lookup).

With these changes, the memory footprint was reduced by approximately 83 %, while computing time was reduced with 50 % at minimum.

PHYLIP stores UPGMA results in Newick tree format, which can be viewed with various tools, incl. PHYLIP's DrawGram.

## 2.4 Result evaluation

One the most important goals of this project is to develop a system capable of finding candidate genes for a given phenotype. For this to succeed, phenotypes that end up together in a clustering or ranking should be related on the genetic level. To test whether this genetic relationship exists, we repeat the ranking and clustering with genotype features associated with OMIM articles, and then compare the result with the phenotype clustering.

### 2.4.1 Extracting genotype features

As of April 2003, more than 50% of the OMIM articles are linked to one or more proteins in the Swiss-Prot and TrEMBL databases (Boeckmann et al., 2003). Swiss-Prot is a curated protein sequence database with a high level of annotations, a minimal level of redundancy and high level of integration with other databases. TrEMBL is a computer-annotated supplement of Swiss-Prot that contains all the translations of EMBL nucleotide sequence entries not yet integrated in Swiss-Prot. In both databases (but especially Swiss-Prot), proteins are commonly linked to entries in the Gene Ontology Annotation database (European Bioinformatics Institute, 2002). In turn, each GOA entry points to an entry in the Gene Ontology (GO) database (Gene Ontology Consortium, 2000). The GO database provides a standardized vocabulary for describing the function of genes and proteins.

Using SRS, we select all OMIM entries that have one or more indirect links to a GO entry (SRS query: `'go>goa>swall>omim'`). Subsequently, we query for GO entries associated with each selected OMIM article (SRS query `'[omim-id:xxxxxx]>swall>goa>go'` for OMIM article id xxxxxx). This method renders a set of 7,803 articles with each on average 7.4 GO links. We then select only a subset of 6,937 GO entries which we deem relevant to the genotype. This subset comprises the category 'biological process', but excludes categories 'molecular function' and 'cellular component'. This restriction on GO terms renders a genotype feature matrix containing 7,633 OMIM articles with each on average 2.8 GO links (genotype features).

Like the MeSH thesaurus, GO provides hierarchical links between entries. We incorporate this hierarchical information as we did for the phenotype feature matrices: parent relevance equals the relevance of the child, divided by the number of children of the parent.

With the augmented genotype feature matrix, we perform ranking and clustering as before. Note that we do not apply corpus size normalization or term weighting.

### 2.4.2 Comparing distance matrices and trees

Using both MeSH (phenotype) and GO (genotype) based feature matrices, we have obtained proximity matrices and trees that define similarities between OMIM articles. To compare genotype- and phenotype results, we use the CoPhenetic Correlation Coefficient (CPCC). This coefficient of matrix correlation originated in phylogeny research (Rolph, 1970), and has become the de facto standard for comparison of hierarchical clusterings (Halkidi et al., 2001; Halkidi et al., 2002). The CPCC can directly be applied to proximity matrices; in order to apply it to trees, one must first construct the CoPhenetic matrix. This is essentially a distance matrix for which the distances  $d(i,j)$  equal the tree level where clusters  $i$  and  $j$  are merged for the first time (see the UPGMA algorithm,  $\mathbf{p}$  vector).

The CoPhenetic Correlation Coefficient takes only one triangular (excl. the diagonal) of distance matrices  $\mathbf{D}_1$  and  $\mathbf{D}_2$  into account, and is given by:

$$CPCC(\mathbf{D}_1, \mathbf{D}_2) = \frac{(1/M) \sum_{i=2}^N \sum_{j=1}^{i-1} \mathbf{D}_1(i, j) \mathbf{D}_2(i, j) - \mu_{\mathbf{D}_1} \mu_{\mathbf{D}_2}}{\sqrt{\left[ (1/M) \sum_{i=2}^N \sum_{j=1}^{i-1} \mathbf{D}_1(i, j)^2 - \mu_{\mathbf{D}_1}^2 \right] \left[ (1/M) \sum_{i=2}^N \sum_{j=1}^{i-1} \mathbf{D}_2(i, j)^2 - \mu_{\mathbf{D}_2}^2 \right]}}$$

for the lower triangular. Here,  $N$  is the number of points in the dataset, and  $M = N \cdot (N - 1) / 2$ .  $\mu_{\mathbf{D}_1}$  and  $\mu_{\mathbf{D}_2}$  equal the means of matrices  $\mathbf{D}_1$  and  $\mathbf{D}_2$  respectively, and are given by:

$$\mu_{\mathbf{D}_1} = (1/M) \sum_{i=2}^N \sum_{j=1}^{i-1} \mathbf{D}_1(i, j), \quad \mu_{\mathbf{D}_2} = (1/M) \sum_{i=2}^N \sum_{j=1}^{i-1} \mathbf{D}_2(i, j)$$

Values of the CPCC can range between -1 and 1. High values indicate close correlation between the distance matrices.

Unfortunately, we cannot determine the significance of CPCC values as we lack a function for (or a means to approximate) the CPCC probability distribution. This distribution is dependent on all parameters involved in ranking and clustering (typically, the number of OMIM articles, the number of features, the proximity measure used, the clustering algorithm used). Hence, one can only approximate the CPCC probability distribution by performing Monte Carlo simulations. Such simulations involve calculation of proximity matrices and clustering for a high number (typically  $>100$ ) of random-valued feature matrices (Theodoridis and Koutroumbas, 1999). For our high numbers of species (i.e. OMIM articles) and features this would be far too time consuming, and therefore not feasible. Hence, CPCC values only provide some indication of the correlation between phenotype-similarities and genotype-similarities in OMIM; CPCC significance is undetermined.

## 3 Results

### 3.1 Comparing feature matrices

To determine to what extent OMIM article ranking and clustering is affected by (1) the length of phenotype descriptions, and (2) the addition of term hyperonyms, we work with the following four matrices:

- A. a feature matrix based on OMIM keyword listings (CS field), without addition of term hyperonyms.
- B. a feature matrix based on OMIM keyword listing (CS field), with addition of term hyperonyms.
- C. a feature matrix based on the full OMIM record (TI/TX/MN/CS fields), without addition of term hyperonyms.
- D. a feature matrix based on the full OMIM record (TI/TX/MN/CS fields), with addition of term hyperonyms.

Articles are included in the feature matrices only if they match one or more MeSH terms. As a result, matrices A and B describe 4,332 OMIM articles (29.2 %), C and D describe 13,893 OMIM articles (93.6 %).

Logically, the following assumptions should hold: (1) feature matrices based on keyword listings contain fewer features per article (i.e. match less terms per article) than the matrices based on the full record. (2) Addition of term hyperonyms increases the number of features per article, but decreases average feature specificity (added hyperonyms are by definition less specific than the original terms).

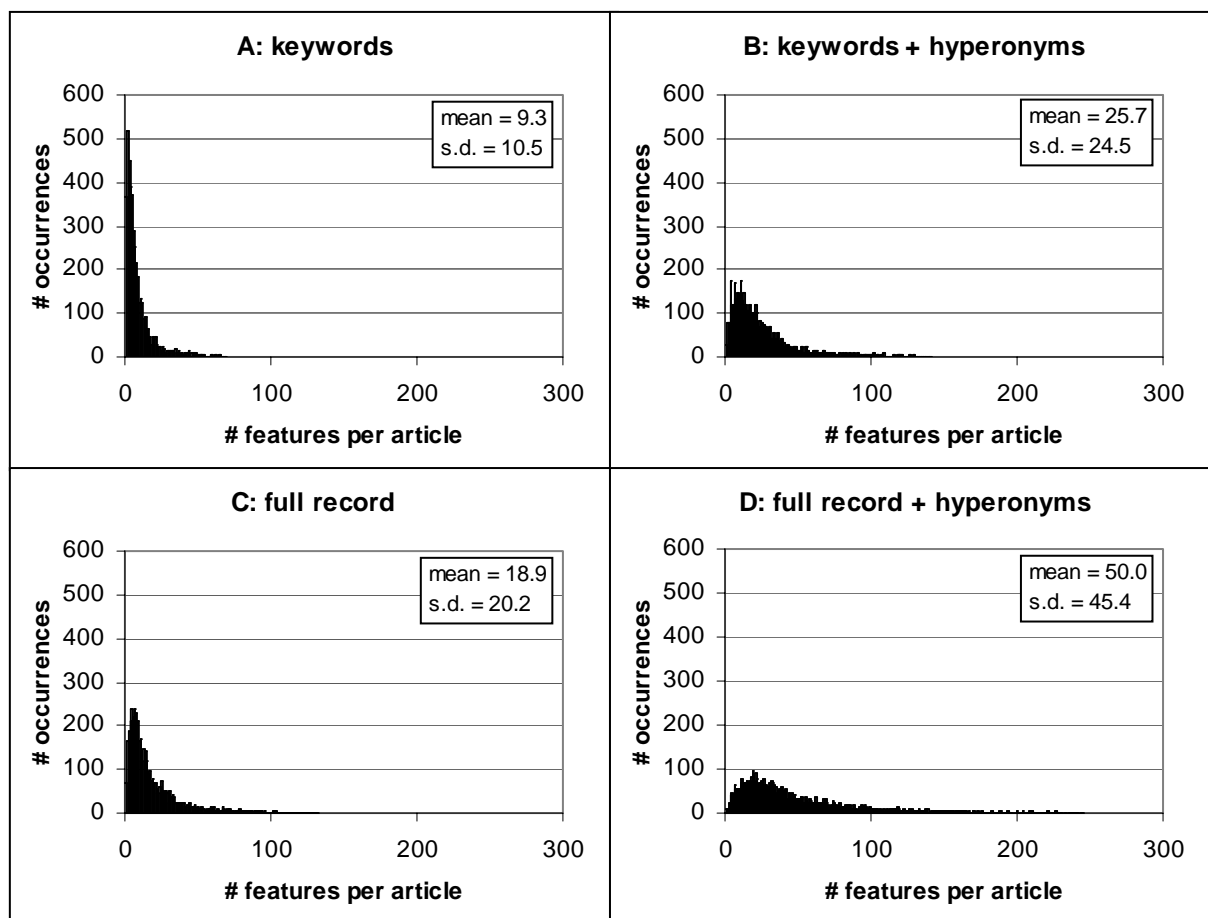
To visualize how the feature count per article is affected by the phenotype source and hyperonym addition, we plot for each feature matrix a histogram of the feature count per article in figure 3.1. Herein, one can see how many articles (y-axis) contain particular numbers of features (x-axis). One can clearly see that full-record based feature matrices contain more features per article than keyword-based matrices (C vs. A, D vs. B), and that hyperonym addition increases the number of features per article (B vs. A, D vs. C).

To visualize how hyperonym addition decreases the specificity of the feature matrix (crudely put: ‘waters it down’) we plot cumulative distributions of feature specificity in figure 3.2. This figure shows what part of the feature matrix (y-axis) consists of features with specificity less than or equal to a particular value (x-axis). The specificity of a feature (i.e. MeSH entry) is derived from its position in the MeSH tree. It is defined as the inverse of (1 + the number of offspring nodes). Note that offspring nodes include children as well as grandchildren, great-grandchildren, etc. Thus, if feature specificity equals 1, the corresponding MeSH entry has no children; if it equals 0.5, the entry has 1 child node, etc. The ‘presence’ of a feature (i.e. MeSH entry) equals the sum of its values in the feature matrix, divided by the sum of all feature matrix values. Cumulative feature presence for a given feature specificity  $a$  equals the sum of the presence-values of all features with specificity  $a$ . Figure 3.2 clearly shows that hyperonym addition decreases average feature specificity in the feature matrix.

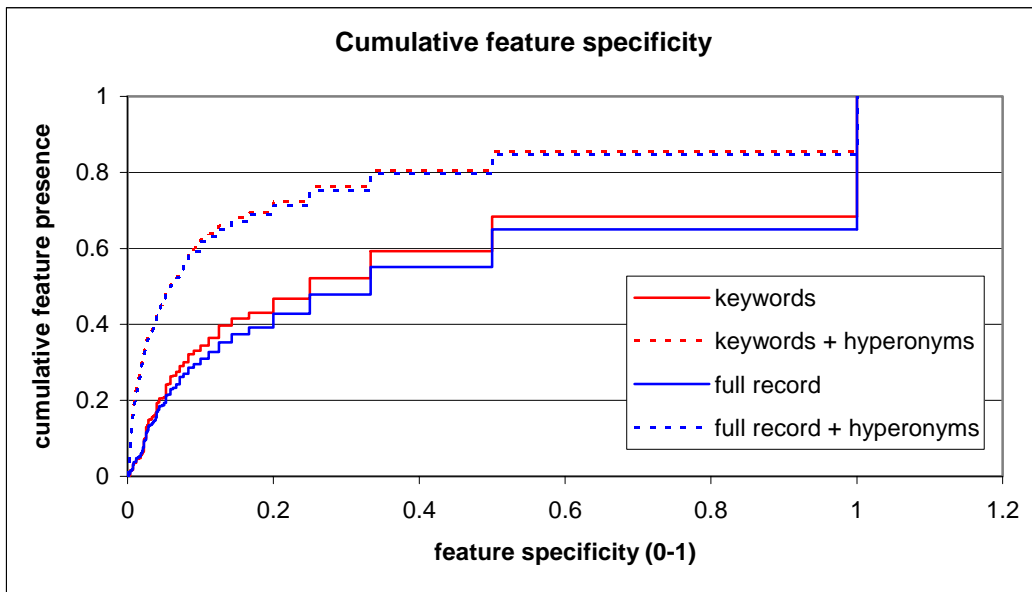
In both figure 3.1 and 3.2, all diagrams are based on the same subset of 4,332 OMIM articles: those that have one or more features in the keyword-based (CS) feature matrix; articles present only in the full-record matrices were ignored. Thus, one can meaningfully compare diagrams for the different feature matrices.



Figures 3.1 and 3.2 clearly show the assumptions on feature count and feature specificity hold. Additionally, one can observe that the addition of term hyperonyms to the keyword-based matrix (B vs. A) causes it to approach the per-article feature count of the full-record matrix (C), at the cost of feature specificity.



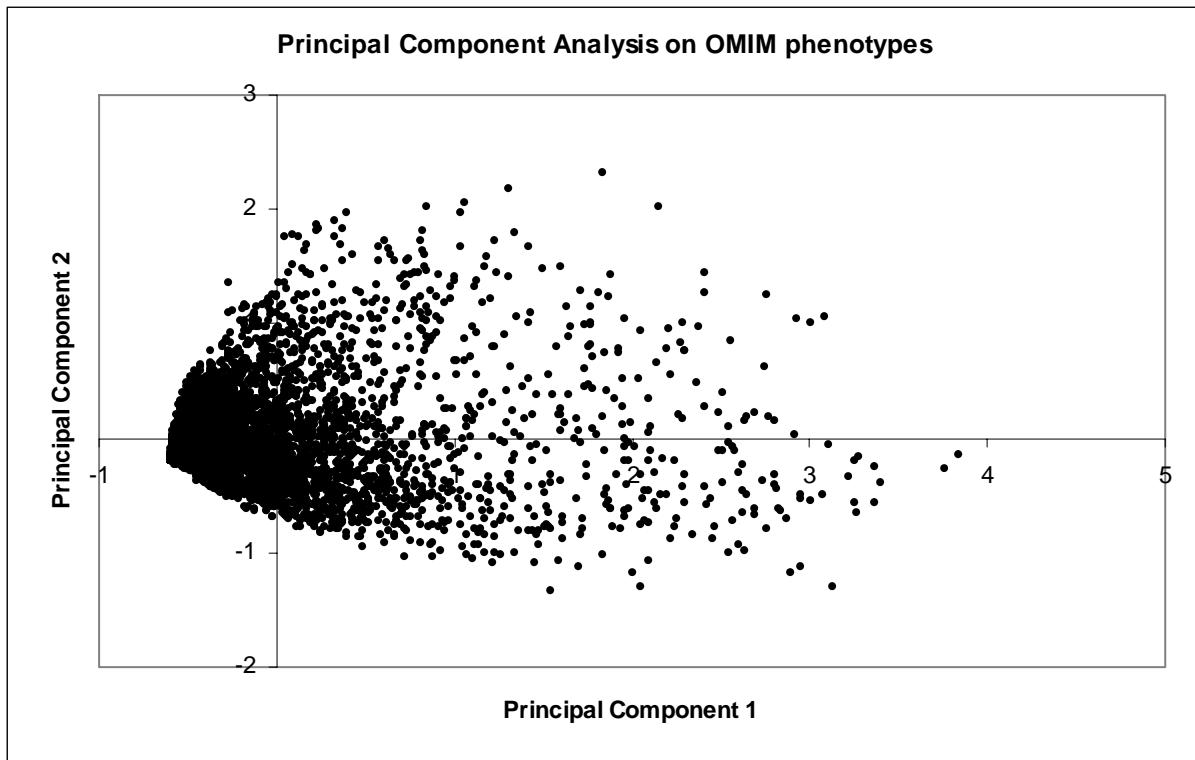
**Figure 3.1.** Histograms of the feature count per OMIM article, taken over each of the four feature matrices (see text). Note that the feature count corresponds to the number of MeSH entries matched by an article, and is independent of actual feature values. Features count as 1 if their value exceeds 0, and are ignored otherwise.



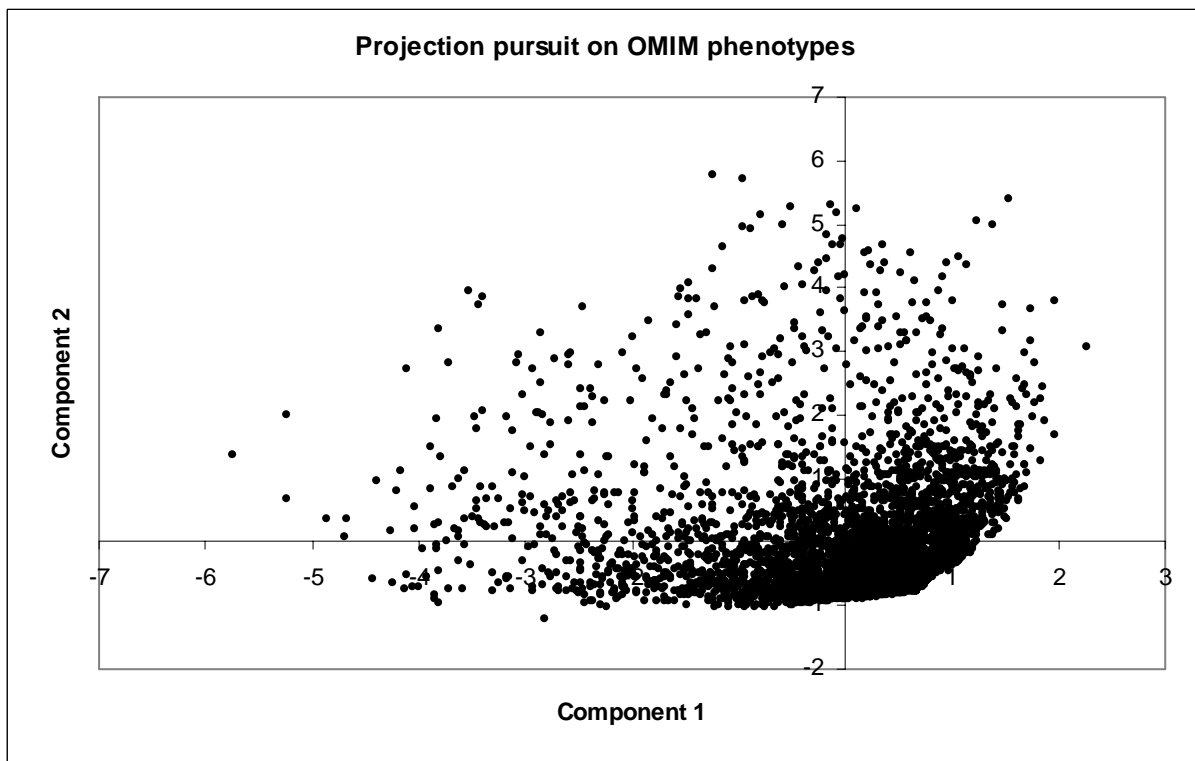
**Figure 3.2.** Cumulative distribution of feature specificity for each feature matrix.

### 3.2 Clustering tendency

Qualitatively, all evaluated multidimensional scaling techniques present the same result when applied to the phenotype feature matrices: no evident clusters are discernable. Typical results are shown in figures 3.3 and 3.4. These show the feature vectors scaled from 4,000+ dimensional space to two-dimensional space, using respectively Principal Component Analysis and Projection Pursuit. In addition, we scaled the feature vectors to three-dimensional space (results not shown), but this did not significantly improve cluster visibility.



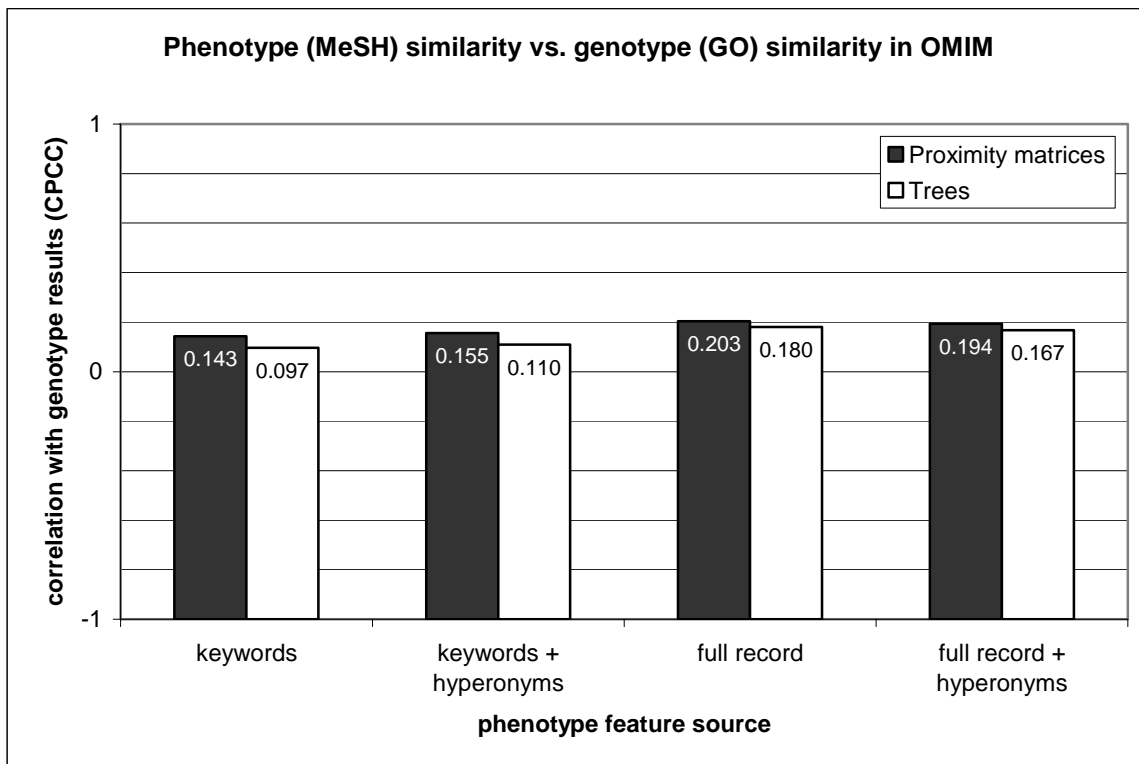
**Figure 3.3.** Principal component analysis on a subset of the full-record-based phenotype feature matrix (no hyperonyms). The subset was used to reduce memory consumption and required computer time. It includes all articles with one or more features in the keyword-based phenotype feature matrix. The first two principal components shown explain 5.6 % and 2.6 % of the variance, respectively.



**Figure 3.4.** Projection pursuit on a subset of the full-record-based phenotype feature matrix (no hyperonyms). The subset was used to reduce memory consumption and required computer time. It includes all articles with one or more features in the keyword-based phenotype feature matrix. The projection pursuit components corresponds to the most ‘interesting’ (i.e. non-Gaussian) directions in the feature matrix.

### 3.3 Phenotype similarity vs. genotype similarity

To establish whether phenotype similarity can provide an indication of genotype similarity, we compare OMIM proximity matrices and trees based on phenotype (MeSH matches) with those based on genotype (GO links), using the CoPhenetic Correlation Coefficient. CPCC values for both proximity matrices and trees are shown for all four phenotype feature matrices in figure 3.5.



**Figure 3.5.** Values of the cophenetic correlation coefficient (CPCC) for (phenotype proximity matrix, genotype proximity matrix) and (phenotype tree, genotype tree) combinations for OMIM. CPCC values can range between -1 and 1, 1 implying perfect correlation. Phenotype proximity matrices and trees are based on MeSH-match feature matrices, genotype proximity matrices and trees are based on GO-link feature matrices. All proximity matrices and trees include only the 1,063 OMIM articles with one or more features in both the keyword-based phenotype feature matrix and the genotype feature matrix. Thus, one can meaningfully compare CPCC values for the different phenotype sources.

As CPCC values can range between -1 (inverse correlation) and 1 (perfect correlation) one can only conclude from figure 3.5 that there is at most a moderate – and probably non-significant – positive correlation between phenotype-based and genotype-based results. CPCC values are not nearly as high as typical CPCC correlation between distance matrices and corresponding cophenetic matrices ( $> 0.8$ ), for instance.

### 3.4 Examples of ranking and trees

Phenotype similarities as defined by our system of ranking and clustering are summarized in proximity matrices and trees. Unfortunately, both structures are difficult – if not impossible – to visualize in full, particularly in print. This is due to the large number of OMIM articles involved: the phenotype feature matrix that provides the best indication of genotype similarity (see figure 3.5) describes about 14,000 OMIM articles. Thus, we have a  $14,000 \times 14,000$

proximity matrix – which, even if it could be visualized, could hardly be called informative –, and a tree with 14,000 nodes.

Visualization of the complete tree could be insightful, as it in essence represents a new classification of human genes and inheritable traits, one based on phenotype. However, even though much progress has been made in the visualization of large tree structures (particularly for on-screen display, see e.g. Munzner et al., 2003), one simply cannot compress all the information in the tree in a way that maintains visibility of individual phenotypes, and yet fits on one sheet of paper. Therefore, we cannot and do not show the full tree structure.

Fortunately, practical applications seeking to use phenotype similarities as a source of candidate genes do not require simultaneous visualization of all phenotypes. Rather, such applications take interest in a subset of  $x$  phenotypes most similar to a reference phenotype  $r$  (which may be another OMIM article or an external, user-defined phenotype). Thus, one can suffice with a view of those  $x$  articles, either as a ranked list (based on proximities), or as the tree branch that includes  $r$  and its  $x$  nearest neighbors<sup>1</sup>.

Figure 3.6 and 3.7 show examples of partial views (ranking and tree branches, respectively) for two OMIM reference phenotypes: Stickler syndrome type I and Alzheimer disease. For OMIM articles associated with one or more genes (in the OMIM gene map), these genes are also listed. To a certain extent, this allows one to judge whether the nearest phenotypes indeed share a similar genotype. For instance, one can see that several phenotypes nearest to the collagen-associated Stickler syndrome either are directly associated with a collagen peptide, or a peptide with similar function (fibrillin is like collagen a constituent of the extracellular matrix, and plays a role during organism development). Likewise, several phenotypes listed next to Alzheimer disease are associated with proteins that have been suggested to play a role in Alzheimer, e.g. the prion protein, amyloid beta (A4) and presenilin.

---

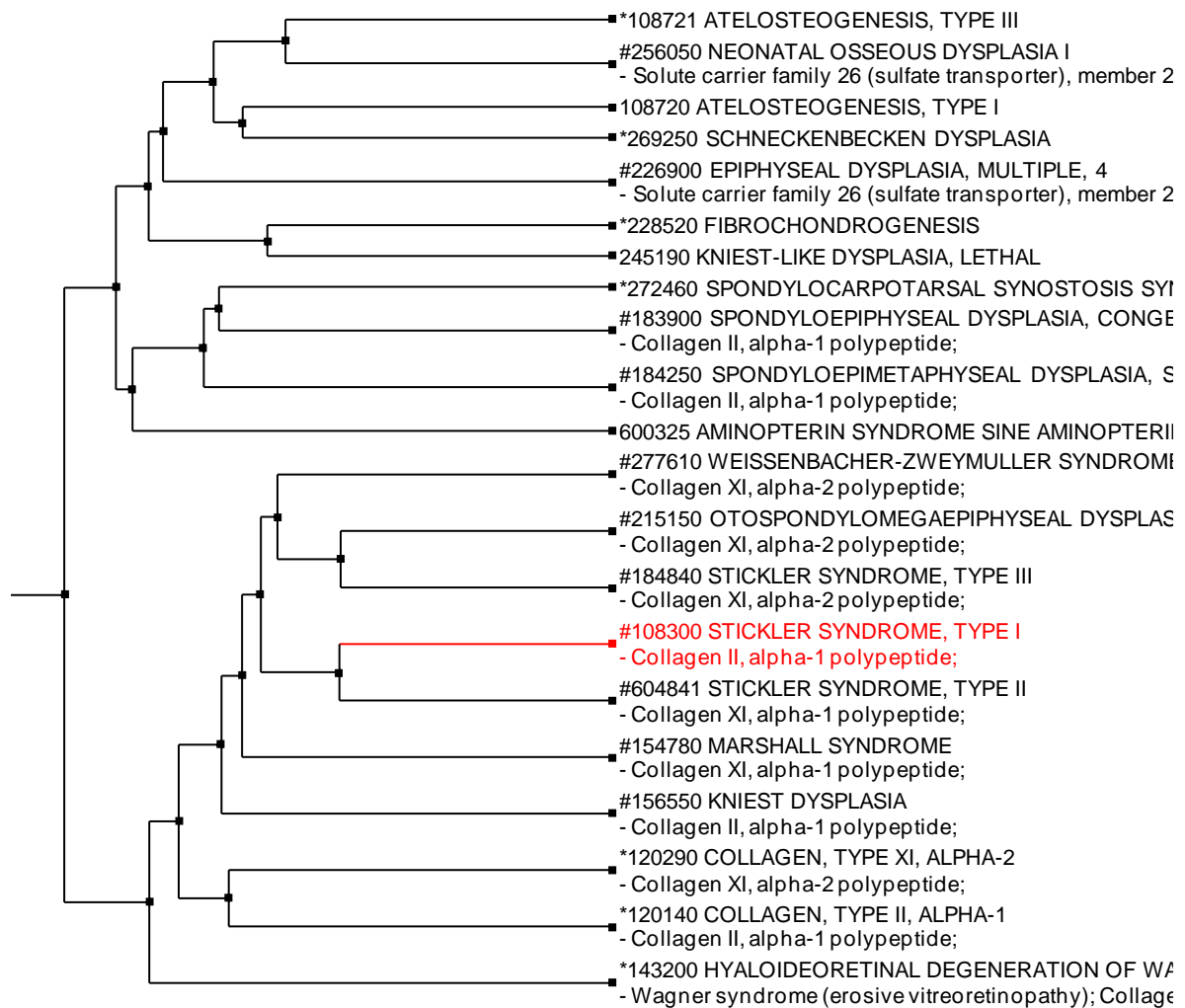
<sup>1</sup> One can rarely display a branch with exactly  $x$  nearest neighbors, as two nodes merged into an intermediate node often each have multiple end nodes as offspring. Moving up one level (i.e. merger) in the tree therefore often increase the number of end nodes in the branch with more than 1. Thus, one can only approximate  $x$  in a branch view.

Proximity	OMIM article	Associated gene(s)
1.0000	#108300 STICKLER SYNDROME, TYPE I	- Collagen II, alpha-1 polypeptide
0.7225	#604841 STICKLER SYNDROME, TYPE II	- Collagen XI, alpha-1 polypeptide
0.6356	#184840 STICKLER SYNDROME, TYPE III	- Collagen XI, alpha-2 polypeptide
0.6197	#215150 OTOSPONDYLOMEGAEPIPHYSEAL DYSPLASIA	- Collagen XI, alpha-2 polypeptide
0.5983	#154780 MARSHALL SYNDROME	- Collagen XI, alpha-1 polypeptide
0.5916	#156550 KNIEST DYSPLASIA	- Collagen II, alpha-1 polypeptide
0.5912	#277610 WEISSENBACHER-ZWEYMULLER SYNDROME	- Collagen XI, alpha-2 polypeptide
0.5867	*143200 HYALOIDEORETINAL DEGENERATION OF WAGNER	- Wagner syndrome (erosive vitreoretinopathy) - Collagen II, alpha-1 polypeptide`
0.5801	*120290 COLLAGEN, TYPE XI, ALPHA-2	- Collagen XI, alpha-2 polypeptide
0.5556	#183900 SPONDYLOEPIPHYSEAL DYSPLASIA, CONGENITAL TYPE	- Collagen II, alpha-1 polypeptide
0.5315	166240 OSTEOGENESIS IMPERFECTA WITH OPALESCENT TEETH	
0.5238	600920 MARDEN-WALKER-LIKE SYNDROME WITHOUT PSYCHOMOTOR RETARDATION	
0.5207	*272460 SPONDYLOCARPOTARSAL SYNOSTOSIS SYNDROME	
0.5191	*121050 CONTRACTURAL ARACHNODACTYLY, CONGENITAL	- Fibrillin-2
0.5174	601353 BRACHYCEPHALY, DEAFNESS, CATARACT, MICROSTOMIA, AND MENTAL RETARDATION	
0.5076	601560 MULTIPLE EPIPHYSEAL DYSPLASIA WITH ROBIN PHENOTYPE	
0.5026	136760 FRONTAL NASAL DYSPLASIA	
0.4985	*120140 COLLAGEN, TYPE II, ALPHA-1	- Collagen II, alpha-1 polypeptide
0.4917	*309583 MENTAL RETARDATION, X-LINKED, SNYDER-ROBINSON TYPE	- Snyder-Robinson X-linked mental retardation syndrome
0.4916	184000 SPONDYLOEPIPHYSEAL DYSPLASIA, MYOPIA, AND SENSORINEURAL DEAFNESS	

**Figure 3.6a.** Phenotypes similar to ‘Stickler syndrome, type I’, with associated genes. The ranking is based on the full-text phenotype matrix, supplemented with hyperonyms.

Proximity	OMIM article	Associated gene(s)
1.0000	#104300 ALZHEIMER DISEASE	- Presenilin 2 - Alzheimer disease 6 - Angiotensin I converting enzyme (dipeptidyl carboxypeptidase-1) - Alzheimer disease 8
0.6215	*104760 AMYLOID BETA A4 PRECURSOR PROTEIN	- Amyloid beta (A4) precursor protein
0.5819	*104311 PRESENILIN 1	- Presenilin 1
0.5220	*176640 PRION PROTEIN	- Prion protein (p27-30)
0.5180	*107741 APOLIPOPROTEIN E	- Apolipoprotein E
0.4756	#176500 DEMENTIA, FAMILIAL BRITISH	- Integral membrane protein 2B (BRI gene)
0.4633	*157140 MICROTUBULE-ASSOCIATED PROTEIN TAU	- Microtubule-associated protein tau
0.4622	*602917 DOWN SYNDROME CRITICAL REGION GENE 1	- Myocyte-enriched calcineurin-interacting protein 1
0.4620	#600274 FRONTOTEMPORAL DEMENTIA	- Presenilin 1 - Microtubule-associated protein tau
0.4513	*603381 FILAMIN B	- Filamin B, beta
0.4505	*176990 S100 CALCIUM-BINDING PROTEIN, BETA	- S100 calcium-binding protein, beta (neural)
0.4470	*147450 SUPEROXIDE DISMUTASE 1	- Superoxide dismutase-1, soluble
0.4454	*192977 VERY LOW DENSITY LIPOPROTEIN RECEPTOR	- Very low density lipoprotein receptor
0.4442	#104310 ALZHEIMER DISEASE 2	- Alzheimer disease-2, late-onset
0.4385	#172700 PICK DISEASE OF BRAIN	- Microtubule-associated protein tau
0.4365	*190198 NOTCH, DROSOPHILA, HOMOLOG OF, 1	- Notch, Drosophila, homolog of, 1, translocation-associated
0.4358	#127750 DEMENTIA, LEWY BODY	
0.4343	#181500 SCHIZOPHRENIA	- Schizophrenia susceptibility locus, chromosome 1q-related - Disrupted in schizophrenia 1 - Disrupted in schizophrenia 2 - Schizophrenia susceptibility locus, chromosome 5-related - Schizophrenia susceptibility locus, chromosome 6-related - Dystrobrevin-binding protein 1 - Schizophrenia susceptibility locus, chromosome 6q-related - Schizophrenia susceptibility locus, chromosome 8p-related - Schizophrenia susceptibility locus, chromosome 11-related - D-amino-acid oxidase - Schizophrenia susceptibility locus, chromosome 13q-related - G72 gene - Schizophrenia susceptibility locus, chromosome 15q-related - Schizophrenia susceptibility locus, chromosome 18-related - Schizophrenia susceptibility locus, chromosome 22-related - Catechol-O-methyltransferase - Proline dehydrogenase (proline oxidase) - Apolipoprotein L1 - Apolipoprotein L2 - Apolipoprotein L4
0.4317	*600759 PRESENILIN 2	- Presenilin 2
0.4308	*143100 HUNTINGTON DISEASE	- Huntingtin

**Figure 3.6b.** Phenotypes similar to ‘Alzheimer disease’, with associated genes. The ranking is based on the full-text phenotype matrix, supplemented with hyperonyms.



**Figure 3.7a.** Neighbors of ‘Stickler syndrome, type I’ in the UPGMA phenotype tree. For phenotypes that are associated with one or more genes, the genes are displayed below the phenotype name. The tree is based on the full-text phenotype matrix, supplemented with hyperonyms.





**Figure 3.7b.** Neighbors of ‘Alzheimer disease’ in the UPGMA phenotype tree. For phenotypes that are associated with one or more genes, the genes are displayed below the phenotype name. The tree is based on the full-text phenotype matrix, supplemented with hyperonyms.

## 4 Conclusions

In constructing any system that identifies phenotype similarities, one deny the fact that this involves a large number of fundamentally subjective decisions: in selection of a description source (keyword listings, full text), choosing a method for feature extraction (term matching with a MeSH-based dictionary), incorporation of feature relations, application of various techniques for feature value refinement (corpus size normalization, feature weighting), selection of a proximity measure and clustering method (UPGMA). Each decision has a significant impact on the final rankings and clusterings; these results are therefore by no means guaranteed to present a satisfying indication of phenotype similarity, let alone genotype similarity.

We cannot evaluate the impact of every single decision. Rather, we focus on answering the following questions:

- How do the format and size of the phenotype description (keyword listing vs. the full record) affect result quality?
- Does the addition of hyperonyms improve result quality?
- Can phenotype similarities serve as an indication of genotype similarities?

### 4.1 Phenotype sources

Figure 3.1 clearly shows that keyword-based feature vectors contain less (non-zero) features than full-record-based vectors; feature count averages differ by approximately a factor 2. This was obvious upfront, as keyword listings contain less text, and therefore match less MeSH terms<sup>1</sup>. It is interesting to notice that the difference in feature count does not affect average feature specificity (figure 3.2, series A and C). Apparently, keyword listings do not contain more or fewer specific terms than the full record.

Ultimately, figure 3.5 shows that phenotype similarities derived from keyword listings are very slightly worse than similarities derived from the full records, if judged by correlation with genotype similarities. From this, one can draw two conclusions. First, phenotype ranking and clustering has little to gain from keyword listings; the full-text record contains sufficient – even better, or more – information on the phenotype<sup>2</sup>. Thus, for databases lacking keyword fields, intervention of domain experts creating keyword listings is unnecessary. Secondly, we can turn this argument around: there is little to gain in quality by choosing full-text phenotype descriptions over keyword listings, particularly since the time taken by feature extraction scales linearly with the size of the phenotype description. Processing keyword listings only is therefore much more efficient. Thus, for any database that includes keyword listings for all records, one could optimize phenotype characterization by processing the keyword listings only.

---

<sup>1</sup> In fact, ‘full-text’ articles are a superset of ‘keyword’ articles, as ‘full-text’ comprises the title (TI), abstract (MN), text (TX) and keyword (CS) fields.

<sup>2</sup> One might (rightly) argue that full-text phenotypes also include keyword listings, and therefore are not independent from keyword listings. To determine to what extent the full-text phenotypes are influenced by the included keyword listings, we repeated the CPCC tests with a phenotype matrix with only the TI, TX and MN fields (i.e. full-text without keyword listings). For this phenotype matrix, CPCC correlation between phenotype-based distance/cophenetic matrices and genotype-based ones was still higher: 0.196/0.200 without hyperonyms, and 0.187/0.186 with hyperonyms. In other words: phenotype matrices based on text truly produce slightly better results than phenotype matrices based on keyword listings.

## 4.2 Hyperonym addition

Incorporation of hyperonyms has the following two effects: (1) the feature count per article increases by a factor 2.7 (figure 3.1), and (2) average feature specificity decreases (figure 3.2). The consequences of these changes in the feature matrix are difficult to predict: in general, an increase in feature counts causes more overlap between feature vectors, and hereby allows for better founded comparisons (i.e. more refined proximity values). However, this is only valid up to a point: when all feature vectors share the same feature set – typically, this is indicated by low average feature specificity – all vectors are alike and detailed comparisons become troublesome. From these two expected consequences, we can derive that hyperonym addition will be relatively more useful for keyword-based feature vectors (with low feature counts) than for full-record-based vectors. Figure 3.5 appears to confirm this: hyperonyms addition slightly improves phenotype-genotype correlation for the keyword-based feature matrix, but worsens this correlation for the full-record-based matrix. Note that both effects are only barely noticeable, however. Likely, a measure of statistical significance for between-CPCC differences would rule all CPCC values equal (i.e. their differences non-significant).

## 4.3 Identification of phenotype similarity to find genes

Ultimately, we face the following question: can we deliver a system that allows for phenotype similarities to be used as an indication of genotype similarities?<sup>1</sup> The cophenetic correlation coefficients presented in figure 3.5 provide some indication, but essentially tell us little more than that the phenotype-based proximity matrices and trees are more similar (CPCC>0) than dissimilar (CPCC<0) to the genotype-based ones. More detailed interpretation of the CPCC values suffers greatly from the lack of an indication of CPCC significance. From the ranking and clusterings samples (figures 3.6 and 3.7), we can at least conclude that in several cases, phenotype similarities indeed indicate genotype similarities. On-screen browsing of the phenotype tree suggests this is likely the case for the majority of phenotypes, particularly those that are well-documented and –consequently – translate into an extensive and specific set of features.

---

<sup>1</sup> This is independent of the question whether one would expect genotype similarities to translate into genotype similarities.

## 5 Discussion

### 5.1 Consequences of dictionary-based feature extraction

In the evaluation of the results, we found several (minor) problems that affect the process of feature extraction. These problems are not specific to our approach. Rather, they will affect any system that uses an internal dictionary for feature extraction.

#### 5.1.1 Ambiguity

The first problem is caused by ambiguous terms: terms that can have different meanings, dependent on the context. This for instance applies to the phrase ‘zinc finger protein’. This protein is thus named because its structure resembles two fingers pinching a DNA strand, not because it is through function or localization associated with the physical finger. Yet, the phrase matches the MeSH term ‘finger’ (under the ‘anatomy’ root), causing the OMIM article to be marked down as a phenotype defined in part by characteristics of the patient’s fingers. Similarly, an article mentioning ‘hearing aids’ gets unjustly listed as a phenotype related to the Acquired ImmunoDeficiency Syndrome (recall our matching of MeSH terms is case-insensitive). Thus, ambiguous terms result in erroneous feature associations.

Whether ambiguity ultimately affects the quality of our phenotype similarity measure varies per article, as it depends on the remainder of the article’s feature set: if an article (correctly) matches a large number of MeSH terms alongside of an ambiguous term, the latter’s impact on the overall feature set is negligible. If an article matches few or no terms aside of the ambiguous term, its similarity to other articles is (almost) fully defined by the one erroneous ambiguous feature. The latter is the case for numerous articles dealing with zinc finger proteins. Most of these articles contain little text aside of repeated mentions of ‘zinc finger protein’. Consequently, they cluster around various finger related traits, e.g. brachydactyly (shortening of the digits), ectrodactyly (a cleft hand is where the patient is missing up to 3 rays in a hand).

Solving ambiguity-related problems is difficult. It requires one to determine (1) which terms are ambiguous, and (2) how to deal with such terms. Identification of ambiguous terms might be done by comparing all MeSH terms to a standard English dictionary (e.g. WordNet, see Fellbaum, 1998). All terms listed in the dictionary as non-medical could then be assumed to be ambiguous. Deciding how to deal with ambiguous terms is more difficult, as simple exclusion of ambiguities would likely imply loss of numerous important terms. One way this might be solved is by placing less weight on ambiguous terms, e.g. by dividing their feature value by, say, a factor 2 (this would then have to be done during rather than after feature extraction, as not all terms referring to a particular MeSH entry are ambiguous).

#### 5.1.2 ‘Inhibiting context’

A slightly different problem is presented if articles contain non-ambiguous terms in a context that explicitly excludes those terms from the phenotype. For instance, one OMIM article contains the phrase ‘[...] Berr et al. (1989) found no evidence of an excess of dementia cases [...]’. As ‘dementia’ is a MeSH term, the article gets incorrectly listed as one concerning dementia. Another article mentions: ‘Myopic individuals had no clinical evidence of connective tissue abnormalities [...]’, causing the article to be put down as a phenotype related to connective tissue.

Like ambiguity-related problems, the problem with ‘inhibiting context’ only affects those articles that contain little text beside the inhibiting phrase. Such articles are relatively

rare, because authors tend to focus primarily on aspects characteristic of (present in) the phenotype. Inhibiting phrases therefore play a relatively minor role.

One cannot solve the problem of ‘inhibiting context’ while maintaining a dictionary-based method of feature extraction. The very nature of dictionary-based methods is an ignorance of context. One solution could be to move to a different, Natural Language processing (NLP) based method of feature extraction, but then one is subject to the problems mentioned in the introduction. Fortunately, one can safely assume that ‘inhibiting phrases’ have little to no effect on the final proximity values.

### **5.1.3 Overweight compound terms**

An advantage of the MeSH thesaurus is its extensive set of compound terms (phrases of multiple words). In essence, these provide a set of highly specific phenotype features. However, many compound terms are composed of other MeSH terms, e.g. ‘cleft palate’ contains ‘palate’. When such compound terms are encountered in an article, they are added multiple times to the feature set: once as compound term, and once for every MeSH part in the term. In effect, this implies more weight is placed on this type of compound term; if alongside of just one other simple term, such compound terms overrule when phenotype similarities are calculated.

It is debatable whether the overweighing of certain compound terms is an advantage or a disadvantage. As these terms are generally highly specific, one could argue that they justly play a major role in the feature set, compared to simple terms. However, it may be preferable to place more weight on such terms in a controlled manner, rather than letting it be determined by the MeSH presence of term components.

One might argue that the compound term overweighing is easily eliminated by restructuring the feature extraction process in such a way that compound terms are tried first, and text matched once is not allowed to be matched again. However, this would have adverse effects: the compound terms would then be judged to be completely unrelated to their individual components. For instance, ‘cleft palate’ would be completely unrelated to ‘palate’ (phenotypes matching the former term would have no feature overlap with phenotypes that match the latter term). This is clearly undesirable. We can only conclude that compound term overweighing is a necessary evil<sup>1</sup>; no elegant, straightforward way exists to deal with such terms.

## **5.2 Suggestions for improvement**

Various stages in the process of determining phenotype similarities allow for some fine-tuning. Particularly, one could evaluate various methods for corpus size normalization and feature weighting, and different proximity measures. In our opinion, however, most gain in quality can be obtained through the following other modifications.

### **5.2.1 A larger dictionary**

The OMIM vocabulary comprises 168,354 unique terms, excluding compound terms. Yet, the phenotype-related MeSH categories contain a total of only 49,740 terms (for 5,391 descriptors). That leaves over OMIM 100,000 terms unaccounted for: these are ignored when constructing phenotype feature vectors. In reality, the number of ignored terms will be even higher, as not all phenotype-related MeSH terms appear in OMIM, and many MeSH terms are compound terms.

---

<sup>1</sup> Though a benevolent evil, as some would interpret it as an added bonus.

Naturally, many ignored terms (present in OMIM but not in MeSH) will bear no relation to the phenotype whatsoever: they are so-called ‘stop words’. Such terms are therefore rightfully ignored. However, several ignored terms undoubtedly do describe the phenotype. For instance, MeSH lacks the vast quantity of Latin terms physicians employ to describe parts of the anatomy. Such terms are clearly relevant to the phenotype, but cannot affect the phenotype features vectors in the current feature extraction system.

To allow for relevant-but-ignored terms to play a role in the final feature vectors, the dictionary used during feature extraction requires expansion. One might attain this by obtaining another dictionary (either a pure medical dictionary, or a more general dictionary that explicitly identifies medical entries), and adding its terms to the current MeSH-based dictionary. However, such a dictionary likely lacks the hierarchical structure of the MeSH thesaurus. Therefore, we cannot add hyperonyms for the added entries as we did for MeSH, making the overall approach a bit dualistic.

Instead, one could also opt to utilize other reference libraries to supplement term sets of current MeSH entries, rather than adding more entries. For instance, one could query an English-Latin (medical) dictionary or another medical thesaurus for synonyms of each MeSH term, and add those newly found synonyms to the term set of the MeSH term queried for. In this approach, every dictionary entry still represents a MeSH descriptor, and hierarchical information remains available for each entry. Hyperonym addition can thus be done globally, and the system’s transparency is maintained.

### **5.2.2 Weighing features based on specificity**

Many systems that perform dictionary-based feature extraction derive a measure of importance for every dictionary term, and adjust the weight of the term (i.e. its feature value) according to this measure. The ‘measure of importance’ can simply describe the rarity of the term (typically the inverse of the proportion of articles containing the term: the Inverse Document Frequency), or it can be more sophisticated (e.g. Wilbur and Yang, 1996). Especially for feature extraction systems that maintain an internal dictionary (see introduction), the introduction of importance-based weighing has been shown to improve results.

In our case, one of the most natural indications of term importance can be derived from the MeSH tree: highly specific terms (deep in the tree, with few children) are likely more important than more general terms (high up in the tree, with many children). In other words, the occurrence of a highly specific term tells us more about the phenotype than the occurrence of a general term. To exploit this characteristic, one could define the importance of a term as the inverse of its number of offspring nodes in the MeSH tree (as we did in figure 3.2), or as the level it occupies in the tree (e.g. 1 for the root nodes, 2 for one level deeper, etc.). Subsequently, one could weight feature values according to this measure through multiplication with (the logarithm of) the ‘importance’ value. In our opinion, incorporation of feature weighing based on term importance could prove both an effective and straightforward means of improvement of the quality of phenotype similarity values.

### **5.2.3 Order of feature value refinements**

Between the process of feature extraction and the calculation of phenotype proximities, we employ a number of techniques to refine feature values. Respectively, we do the following: (1) addition of hyperonyms, (2) corpus size normalization and (3) feature weighing based on number of term occurrences. If one would decide to incorporate global feature weighing based on term specificity (see previous paragraph), this would add yet another stage to this

process. Each distinct step is well motivated, and expected to improve result quality. However, the exact order of these steps is arbitrary, yet influences the results to a great extent.

For instance, addition of hyperonyms (1) before feature weighing (3) allows for the hyperonyms to exert more influence than if this order were reversed. This can easily be seen: hyperonyms are by definition assigned a lower feature value than direct matches, while the process of feature weighing attributes more weight to features with low values. In our case, feature weighing ensures any non-zero feature gets value 0.5 (half of the maximum feature value) at minimum (function:  $0.5 + 0.5 x$ ). Thus, even the most distant MeSH ancestor plays a relatively major role. At the same time, differences between ancestor distances are obscured, as all hyperonym feature values are low and their final feature value will be little above 0.5.

Similarly, performing corpus size normalization (2) in advance of feature weighing (3) renders other results than if this order were reversed. In fact, corpus size normalization (i.e. division of feature vectors by a per-article constant factor) would be of no use if performed as final step in feature refinement, as the proximity measure used (length-normalized correlation) is insensitive to the length of the vectors (see chapter 2). Thus, division of a feature vector by some value cannot affect between-vector proximities. By performing normalization before feature weighing, the weighing routines place more weight on features with values that are small relative to the other article features, rather than small in the absolute sense of the word.

We cannot avoid the fact that the exact order of feature refinement steps is arbitrary. Even if we realize how the order influences results, a completely objective choice on the refinement sequence is impossible to make. However, we can conclude that for hyperonym distances to be adequately represented in feature vectors, the process of hyperonym addition might better be moved to the final stage of feature refinement (though this will also cause hyperonyms to be less important compared to direct matches). In addition, corpus size normalization currently allows for feature weighing base on the relative (per article) importance of features, rather than their absolute value, and would be of no effect otherwise. The normalization-before-weighing sequence therefore is likely best preserved.

#### 5.2.4 Result analysis

This project mainly deals with the development of a system that calculates phenotype similarities; steps taken to evaluate the quality of the similarities as a function of description size and hyperonym addition may therefore appear to have been somewhat of an afterthought. To a certain extent, this is the case: many more sophisticated methods might be used to test result quality under different conditions. In particular, one might construct feature matrices based on random subsets of the full-record matches (i.e. randomly removing a fixed number of matches from each article), and subsequently evaluate result quality with and without hyperonym addition (with the CPCC). However, such methods of evaluation are beyond the scope of this project.

To determine the quality of phenotype similarities, we calculate the correlation between phenotype (MeSH-based) and genotype (GO-based) similarities in OMIM. This renders a measure specifying to what extent similar phenotypes are associated with similar genotypes, and is therefore a highly suitable indication of the practical usability of our method. However, the current method of determining phenotype-genotype correlation suffers from a number of problems.

First, the genotype feature matrix has very low feature counts per article (2.4 on average). This implies there will be little overlap between genotype features of different OMIM articles; often, articles will be completely dissimilar. As a result, genotype similarities

merely provide a qualitative indication of similarity; they have little quantitative value. As these similarities are in turn used to calculate the correlation (CPCC) between phenotype- and genotype-based similarities, these correlation coefficients will consequently hold little quantitative information.

To improve the quality of CPCCs, one can only seek to extend the genotype feature matrix. This might be done by including other GO categories in addition to the current one ('biological process'), but these other categories were explicitly excluded because they described genotype aspects unrelated to the phenotype (typically non-specific terms, e.g. 'ion transport', 'enzyme', 'ATP-binding'). Inclusion of such categories might prove beneficiary provided non-specific term were filtered out. Even without inclusion of other GO categories, however, we can expect the genotype feature count per article to increase as research into gene function progresses (causing more GO terms to be linked to OMIM-associated genes), and the GO dictionary expands. Thus, even with current restrictions on GO terms, the quality of the genotype feature matrix will improve.

Additionally, interpretation of the phenotype-genotype correlation coefficients (CPCC) suffers from the lack of a measure of CPCC significance. Such a measure can only be found with Monte Carlo simulations, requiring the calculation of CPCC values for hundreds of proximity matrices and trees, defined for random phenotype feature matrices. This seems not feasible, as the combined time required for one simulation would likely already exceed 1 hour. However, it might pay to research the application of Monte Carlo simulations further, as such techniques are ubiquitously applied and therefore extensively studied and optimized. Alternatively, one may opt to use another measure of distance matrix correlation than the CPCC, preferable one which statistical significance is more easily determined. Such statistics stem from the booming field of clustering validation techniques, which – though often ignoring or evading the topic of hierarchical clustering – offers several correlation measures that may be suitable (e.g. Theodoridis and Koutroumbas, 1999).

### 5.3 General applicability

In our opinion, the OMIM phenotype similarities we calculate are of sufficient quality for them to be useful in practice; the genes associated with an OMIM phenotype similar to a reference phenotype  $r$  are often related to the gene causing  $r$  (figures 3.6 and 3.7). Thus, our system can prove useful in the identification of candidate genes for unmapped inheritable human traits or diseases.

However, for our method to be useful in other areas of research (for instance crop/cattle breeding), the phenotype descriptions and the dictionary used for feature extraction must satisfy several requirements. We can make the following recommendations:

- Phenotypes should be described in relative detail, using an extensive vocabulary. This is particularly important if the collection of phenotypes studied is large.

If descriptions lack detail, the number of features one can obtain per description is limited. If the vocabulary is small, the global number of features (taken over the complete phenotype set) is limited. Both characteristics impede well-founded quantitative judgments on phenotype similarity, as such judgments require a balance between zero and full feature overlap (implying completely dissimilar and completely identical phenotypes, respectively). A too small number of features per phenotype will cause phenotypes to be completely dissimilar to most others, whereas a too small global number of features will cause many phenotypes to be identical.



- The dictionary used for feature extraction must contain most phenotype-related terms employed in phenotype descriptions. Only then can one translate the full set phenotype characteristics into features. Note that this requirement is eliminated if phenotypes are described with a controlled and documented vocabulary (e.g. as in FlyBase, FlyBase Consortium, 2003); in that case, the controlled vocabulary may serve as dictionary.
- If the phenotype descriptions contain synonyms of dictionary entries (either true ‘linguistic’ synonyms, or conjugations), the dictionary should contain those synonyms, grouped together with the ‘base term’ into one entry. Note that synonyms typically occur only in full-text descriptions and abstracts, not in keyword lists.
- Preferably, one should possess information on relationships between dictionary entries, such as provided by the MeSH thesaurus through hierarchical links. By incorporating this information in the feature vectors (through hyperonym addition), one can ensure the presence of closely related features increases – rather than decreases – phenotype similarity.

In many areas of research, these recommendations will currently be difficult to follow. Phenotype descriptions often leave much to be desired; both a lack of description detail and a too small vocabulary are common. This is for instance typical for databases that describe plant phenotypes. Also, high-quality (i.e. extensive) dictionaries for non-human phenotype terms are rare. Human phenotypes are typically characterized by medical terms, which can be found in any of the numerous medical dictionaries available. Resources that contain terms describing the anatomy and appearance of other species are far less common. For certain animal species, the MeSH thesaurus might still suffice, as it contains the category ‘animal structures’ (a subset of ‘anatomy’). However, this category lists only the most basic anatomical terms, and may therefore be too limited to significantly aid phenotype feature extraction in non-human species.

Fortunately, we can expect both the quality of the phenotype descriptions and the availability of dictionaries to improve in the future, as the scientific community settles on standardized vocabularies to describe phenotypes, and research into gene-phenotype relations progresses.

## 5.4 Concluding...

Undoubtedly, the best phenotype characterization would be obtained with a context-sensitive grammar parser, capable of recognizing and understanding not only the clinical terms themselves, but also term relationships (e.g. MeSH hierarchical links). [In other words: a human. But a rare species of human indeed: one willing to study thousands and thousands of articles, while taking no more than a few seconds per article at maximum...] However, such a complex, sophisticated text-processing system would require far, far more time than a simple text matching approach. When we observe that our relatively simple text matching routines already take 14 hours to process OMIM’s 14,000+ articles (let alone MedLine’s millions of abstracts), it is clear that we simply are not there yet. As long as computer processing power is limited, one needs to compromise between perfect interpretation of articles, and doing so in an acceptable amount of time. Given this, fast text matching, combined with information on synonyms, conjugations and hierarchical term relationships may well prove to be an excellent solution.

## 6 References

- Blaschke C, Hirschman L, Valencia A** (2002). Information extraction in molecular biology. *Briefings in Bioinformatics* **3**(2): 154-65.
- Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, Gasteiger E, Martin MJ, Michoud K, O'Donovan C, Phan I, Pilbout S, Schneider M** (2003). The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res* **31**(1): 365-70.
- European Bioinformatics Institute** (2002). Gene Ontology Annotation (GOA), <http://www.ebi.ac.uk/GOA/>.
- Fellbaum C** (1998). *WordNet: an electronic lexical database*. Cambridge, Mass, MIT Press.
- Felsenstein J** (1993). PHYLIP (Phylogeny Inference Package) 3.5c/3.6 beta, Distributed by the author, Department of Genetics, University of Washington, <http://evolution.genetics.washington.edu/phylip/>.
- FlyBase Consortium** (2003). The FlyBase database of the Drosophila genome projects and community literature. *Nucleic Acids Res* **31**(1): 172-5, <http://flybase.org/>.
- Gene Ontology Consortium** (2000). Gene ontology: tool for the unification of biology. *Nat Genet* **25**(1): 25-9, <http://www.geneontology.org/>.
- Halkidi M, Batistakis Y, Vazirgiannis M** (2001). On Clustering Validation Techniques. *Journal of Intelligent Information Systems (JIIS)* **17**(2-3): 107-145.
- Halkidi M, Batistakis Y, Vazirgiannis M** (2002). Cluster Validity Methods: Part I. *SIGMOD Record* **31**(2): 40-45.
- Hamosh A, Scott AF, Amberger J, Valle D, McKusick VA** (2000). Online Mendelian Inheritance in Man (OMIM). *Human Mutation* **15**(1): 57-61, <http://www.ncbi.nlm.nih.gov/omim/>.
- Hyvärinen A, Oja E** (2000). Independent component analysis: algorithms and applications. *Neural Networks* **13**(4-5): 411-430.
- Munzner T, Guimbretière F, Tasiran S, Zhang L, Zhou Y** (2003). TreeJuxtaposer: Scalable Tree Comparison Using Focus+Context With Guaranteed Visibility. submitted to the SIGGRAPH 2003 conference.
- R Foundation for Statistical Computing** (2002). R 1.6.2, <http://www.r-project.org/>.
- Rolph FJ** (1970). Adaptive hierarchical clustering schemes. *Systematic Zoology* **19**: 58-82.
- Sneath PHA, Sokal RR** (1973). *Numerical taxonomy*. San Francisco, CA, USA, Freeman.
- Tanabe L, Wilbur WJ** (2002). Tagging gene and protein names in biomedical text. *Bioinformatics* **18**(8): 1124-32.
- Tanimoto T** (1958). *An elementary mathematical theory of classification and prediction*, IBM Corp.
- Theodoridis A, Koutroumbas K** (1999). *Pattern recognition*. San Diego, CA, USA, Academic Press.

**U.S. National Library of Medicine** (2003). Medical Subject Headings (MeSH),  
<http://www.nlm.nih.gov/mesh/>.

**Wilbur WJ, Yang Y** (1996). An analysis of statistical term strength and its use in the indexing and retrieval of molecular biology texts. *Computers in Biology and Medicine* **26(3)**: 209-22.

## 7 Acknowledgements

Thanks...

To Jack Leunissen, for putting up with my (seemingly) random disappearances to Amsterdam, and the reappearances when-you-least-expect-them; for impressive lectures on those beautiful programming languages whose rise and fall predated my birth (still, give me C# anytime ;-)); for appreciating the world's most serious authors (Terry Pratchett, P.G. Wodehouse) and the qualities of beer, a smoke and a walk in the garden; for playing an afternoon of Stones...

To Marc van Driel, for making a house call at the VU to discuss the results (quite a change from my common 4-hour-a-day train traveling schedule); for presenting my first true 'kerstpakket', one week before the official distribution date, and the day I left for Wageningen; for daring to study my Perl code...

To Bob Kooi, for numerous career suggestions (and yet, I still don't know... go figure), and advice on correct mathematical notation (dare to call it preference!).

To Travis Longcore, for keeping my interest in INCA (<http://www.urbanwildlands.org/INCA>, subliminal advertising anyone?) going by supplying user comments, and – \*sigh\* – bug reports... Creating the 1.21 update provided some much needed diversion...

To Bernd Brandt, for suggesting writing to Jack for a 'stage' in the first place.