Kernel–based Visualisation of Genes with the Gene Ontology

Hamid Ghous[1] **Paul J. Kennedy**[1] **Daniel R. Catchpoole**[2]
Simeon J. Simoff [3]

[1] Faculty of Engineering and Information Technology,
University of Technology, Sydney,
PO Box 123, Broadway NSW 2007, AUSTRALIA,
Email: Hamid.Ghous@student.uts.edu.au, paulk@it.uts.edu.au

[2] Tumour Bank, The Children's Hospital at Westmead,
Locked Bag 4001, Westmead NSW 2145, AUSTRALIA,
Email: DanielC@chw.edu.au

[3] School of Computing and Mathematics, University of Western Sydney,
NSW, AUSTRALIA,
Email: s.simoff@uws.edu.au

Abstract

With the development of microarray–based high–throughput technologies for examining genetic and biological information en masse, biologists are now faced with making sense of large lists of genes identified from their biological experiments. There is a vital need for "system biology" approaches which can allow biologists to see new or unanticipated potential relationships which will lead to new hypotheses and eventual new knowledge. Finding and understanding relationships in this data is a problem well suited to visualisation. We augment genes with their associated terms from the Gene Ontology and visualise them using kernel Principal Component Analysis with both specialised linear and Gaussian kernels. Our results show that this method can correctly visualise genes by their functional relationships and we describe the difference between using the linear and Gaussian kernels on the problem.

Keywords: kernel–based visualization, Gene Ontology, biomedical datasets.

1 Introduction

It is well recognised that improvements in health are universally driven by gains in understanding of the biology behind human disease. With the completion of the Human Genome Project and the development of microarray–based high–throughput technologies for examining genetic and biological information en masse, biologists are now seeking to assess systems of biological information rather than single genes. Consequently they have to deal with large amounts of information such as lists of hundreds or even thousands of genes. There is a vital need for tools which not only relate this mass of data to current knowledge through bioinformatic approaches but can assess this data to allow us to see new or unanticipated potential relationships within the system which will lead to new hypotheses and eventual new knowledge. In other words, "systems biology" approaches are required. Finding relationships in this morass of data is a problem that is well suited to unsupervised data mining methods such as visualisation. Unsupervised learning methods deal with similarity measures between items of interest, in this case genes. To find similarities, lists of genes must be augmented with additional information. In this paper, we will augment genes with their associated terms from the Gene Ontology (GO) (Ashburner et al. 2000), which is a massive Internet database curated by biologists that defines over 25000 terms in a controlled vocabulary describing genes and gene products.These gene terms fit into three disjoint hierarchies or subontologies: cellular components, molecular functions and biological processes. Terms in the cellular component hierarchy are associated with the physical structure of gene products and generally relate to where the gene product is found in the cell. Terms in the molecular function hierarchy describe the biochemical activity of gene products. Finally, terms in the biological process hierarchy relate to the biological objective to which genes or gene products contribute. The Gene Ontology takes the form of a large database and allows GO terms to be found for specific genes and gene products. Other information, including cross references to other bio–databases, may also be found for genes. We calculate the similarity between genes using the similarity between their component terms.

1.1 Related Work

Several other researchers have explored the problem of applying unsupervised learning methods to lists of genes. Work generally falls into three main areas: describing groups of genes in terms of their annotations; measures for calculating similarity between genes using GO annotations; and clustering and visualisation of genes using GO annotations. The last two of these are similar because clustering and visualisation methods require similarities to be calculated for the genes.

Methods to describe groups of genes from GO annotations include methods that view the Gene Ontology as a simple collection of terms without exploiting too much the structural interrelationships (eg. Gat-Viks et al. (2003) or Shah & Fedoroff (2004)). Others including Beißbarth & Speed (2004) and Zeeberg et al. (2003) use statistical methods to analyze the GO categories. Cheng et al. (2004) use the Bootstrap Test on GO cliques to determine the statistical categorizing of GO categories. Lee et al. (2004) introduced an algorithm to find the significant biological features of a gene cluster or group of interest through the tree structure of the Gene Ontology. They applied a transformation of the GO directed acyclic graph structure

with a distance function. Their graph theoretic algorithm extracts common or representative GO terms for a gene cluster by taking the multi–functionality of genes into account. Popescu et al. (2004) construct a functional summary of clusters of genes using GO terms. They build a "most representative term" (MRT) for each cluster by making a hierarchical clustering of the genes and then applying fuzzy methods to find the GO terms of highest frequency. Liu et al. (2005) describe a tool called DYnGO, which allows users to conduct batch retrieval of GO annotations for a list of genes and semantic retrieval of genes and gene products sharing similar GO annotations. Results are shown in a tree format sorted by GO term.

Similarity measures usually use the hierarchical structure of the Gene Ontology. Mathur & Dinakarpandian (2007) describe an approach to computing gene product similarity by considering both the hierarchical nature of GO and the co–occurrence of GO terms in annotations. Their approach considers numbers of annotations and differences in the frequency of usage of GO terms with a set–based similarity function. Sanfilippo et al. (2007) categorise GO based similarity approaches into two main categories: similarities based on hierarchical relationships within a GO subontology (of which there are three subontologies) and similarities based on associative relationships of genes across the three subontologies. This latter approach predicts annotations in a subontology for a specific gene based on annotations for similar genes. They propose a method called cross–ontological analytics that merges these approaches. They also integrate textual data from biomedical literature with GO knowledge.

Clustering and visualisation approaches go further and apply gene similarity measures to understanding the natural structure of groups of genes and gene products. Lee et al. (2005) propose an ontology–based clustering algorithm (CLUGO) that identifies clusters of significant GO terms within a distribution of terms (eg. that arise from some previous clustering exercise). Kennedy & Simoff (2003) describe a technique for clustering genes based on GO terms using the MBSAS clustering algorithm. However, the method is sensitive to gene order and does not scale to large numbers of genes.

Speer et al. (2005) and Fröhlich et al. (2007) describe a kernel–based approach to clustering genes using Gene Ontology annotations. They define a kernel based on information–theoretic measures to calculate similarities between genes (based on the maximum similarity between terms). They state that their information–theoretic approach better models the variable branching and density of the GO graph and that it should perform better than link distance based measures like the one we use. They apply a dual k–means clustering to groups of genes and provide an R tool.

Like Fröhlich et al. (2007) we devise a kernel function. However, our measure is link distance based rather than information–theoretic. Also, Fröhlich et al. (2007) focusses on clustering, specifically a dual k–means clustering approach. Our motivation, on the other hand, is to visualise the genes with the longer term goal of explaining why a particular set group the way they do.

The rest of this paper is organised as follows. Section 2 elaborates on our approach to visualising a list of genes. Section 3 details the dataset used in this paper to validate our approach: a dataset derived from the KEGG (Kanehisa et al. 2008) database. Also, in section 3 we describe a series of experiments applying variants of our approach together with results. In section 4 we list potentially fruitful areas for future research. Finally, section 5 concludes the paper.

2 Method

This section describes our approach to visualisation of lists of genes. First we describe in more detail the Gene Ontology and the type of data we extract from it. Then we describe the unsupervised visualisation approach we apply, namely kernel Principal Component Analysis (kPCA).

2.1 Gene Ontology

The Gene Ontology provides a controlled vocabulary to describe genes and gene product attributes in many organisms. It is a collaborative effort beginning in 1998 and spans many organisms including but not limited to Drosophila, Saccharomyces, mouse and human.

The building blocks of the Gene Ontology are the terms. Each entry in GO has (i) a unique alphanumerical identifier (GO :#######); (ii) term name, e.g. cell, fibroblast growth factor receptor binding or signal transduction; (iii) synonyms (if applicable); and (iv) a definition. Each term is also assigned to one of the three hierarchies, which are structured as directed acyclic graphs. Most terms have a textual definition, with references stating the source of the definition. If any clarification of the definition or remarks about term usage is required, these are held in a separate comments field.

Each gene has one or more terms related to it and a term may have multiple parents on the tree. The terms provide us with a description of the functionality of a gene.

Table 1 shows three example genes with their related terms. Following each term name is the Gene Ontology accession number for the term. One of the challenges with using terms from the Gene Ontology is that terms give different amounts of information. For example, the gene Aldh1a7 in Table 1 contains some very specific terms such as "retinal metabolic process" or "aldehyde dehydrogenase (NAD) activity" which give specific and useful information along with other terms such as "cytoplasm" or "metabolic process" which are more general (high in the hierarchy) and shared by many other genes. These latter terms do not confer much useful information. Also, some genes have been investigated thoroughly and have many annotations (such as Aldh1a7) whilst others are not well annotated (such as Srpx2). In short, the information associated with genes in the Gene Ontology is of mixed quality. This presents challenges for its use in augmenting lists of genes.

Table 1: Example of three genes from the Gene Ontology.

Gene Name	Term Name and Accession
Aldh1a7	cytoplasm (GO:0005737) oxidoreductase activity (GO:0016491) aldehyde dehydrogenase (NAD) activity (GO:0004029) metabolic process (GO:0008152) retinal metabolic process (GO:0042574)
Srpx2	electron transport (GO:0006118) extracellular region (GO:0005576)
Tspan7	biological process (GO:0008150) molecular function (GO:0003674) integral to membrane (GO:0016021) membrane attack complex (GO:0005579)

As illustrated in Fig. 1, GO terms are related in

two main ways: "is–a" and "part–of". The "is–a" relationship is the main relationship seen in the Gene Ontology and represents a simple class–subclass relationship. For example, the figure shows that the term "extracellular space" is an "extracellular region part" and that an "extracellular region part" is a "cellular component". Cellular component is the root of the hierarchy. Less commonly seen is the "part–of" relationship which signals containment. If C is "part–of" D it means that whenever C is present, it is always a part of D, but that C does not always have to be present. For example, in the figure "extracellular region part" is part of "extracellular region".

The Gene Ontology database allows SQL queries of the terms associated with genes, the relationships between terms (parent and child) as well as finding the distance between terms in number of "hops". There are also many web–based tools available to query the databases.

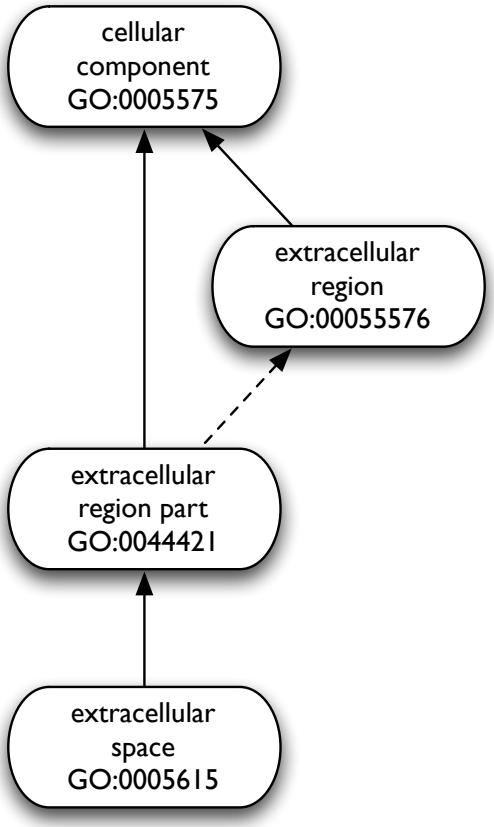

Figure 1: Example of small part of the hierarchical structure of GO terms. Solid lines represent "is–a" relationships and dashed line represents a "part–of" relationship.

2.2 Kernel Principal Component Analysis

The visualisation approach we use in this paper is a kernel–based extension to Principal Component Analysis (PCA) (Jolliffe 2004, Haykin 1999). Principal Component Analysis is a well known data transformation method that rotates a dataset into a different coordinate system. The coordinates of the transformed dataset (called principal components) are orthogonal linear combinations of the original coordinates. The principal components are ordered in descending order by the amount of variance they explain in the data. Generally, most of the variance in the dataset can be explained by many fewer coordinates than in the original dataset (e.g. less than ten) with the last principal coordinates often associated with noise components of the original data. Consequently, PCA is often used for compression of data or feature selection. Principal Component Analysis allows visualization of datasets by plotting the first two or three principal components of the data. However, due to the fact that the principal components are linear combinations of the original dataset, PCA has the limitation that it can model only linear relationships in the data.

When applying PCA the dataset can be viewed as a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ where n is the number of data items each containing d attributes and the d–dimensional row vector x_i represents each data item. The principal components of the dataset are the eigenvectors of the covariance or correlation matrix of $\mathbf{X}$ ordered by decreasing value of the associated eigenvalue. So the first principal component is the eigenvector of the covariance/correlation matrix with the largest eigenvalue. The data is transformed into the principal component space by projecting each data item x_i along the principal components.

Several approaches have been devised to extend PCA to recognise nonlinear relationships among data attributes. One approach is kernel PCA (kPCA) (Müller et al. 2001, Haykin 1999, Shawe-Taylor & Cristianini 2004) which transforms the dataset $\mathbf{X}$ into a feature space using a (nonlinear) kernel function κ before the PCA is done. Kernel PCA returns the principal components of the data items in the feature space. The input to kPCA is a Gram kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ which is a representation of the original dataset transformed with the kernel function. Each element k_{ij} of the kernel matrix can be viewed as a similarity between the data items x_i and x_j and is defined as

$$k_{ij} = \kappa(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \tag{1}$$

where x_i and x_j are the data items, $\phi(x_i)$ is the transformation of x_i into the "feature" space and $\langle \cdot, \cdot \rangle$ is the dot product operator. Generally it is not necessary to compute $\phi(x_i)$ explicitly. Instead, $\mathbf{K}$ is computed directly from the dataset. This is called the "kernel trick" and it means that the feature space can be very large without making generation of $\mathbf{K}$ inefficient. It also means that non–vectorial data types can be handled using special kernels such as string kernels (e.g. (Leslie et al. 2004)). In kPCA the principal components are the eigenvectors of the kernel matrix.

Two common kernel functions are the *linear kernel* and the *Gaussian kernel*. The linear kernel is defined as

$$\kappa(x_i, x_j) = \langle x_i, x_j \rangle \tag{2}$$

and is simply the dot product of the two data items. The whole linear kernel matrix $\mathbf{K}$ can be easily computed as $\mathbf{K} = \mathbf{X}\mathbf{X}'$ where $\mathbf{X}'$ denotes the transpose of $\mathbf{X}$. Kernel PCA using the linear kernel is analogous to the standard (linear) PCA.

The Gaussian kernel explicitly considers the distance between data items and is defined as

$$\kappa(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) \tag{3}$$

where σ is a control parameter governing the "width" of the Gaussian curve. The Gaussian kernel can also be viewed as a series of transformations applied to the linear kernel. Specifically, $\kappa(x_i, x_j) = \exp\left(\langle x_i, x_j \rangle / \sigma^2\right)$ and then normalised (Shawe-Taylor & Cristianini 2004).

In this study, we employ kPCA with both linear and Gaussian kernel matrices. However, as described

below, we use a slightly different linear kernel matrix to be able to use the Gene Ontology data. Also, before applying kPCA, we centre and normalise the data through the kernel matrix.

2.3 The Kernel Function for the Gene Ontology Data

Given a set of genes G, we query the Gene Ontology to find all GO terms directly associated with the genes. Define T as the set of GO terms directly associated with any of the genes in G.

From a list of genes we create a matrix $\mathbf{X} \in \mathbb{R}^{n \times t}$ where n is the number of genes (ie. $|G|$) and t is the number of GO terms (ie. $|T|$). Each element x_{ij} of $\mathbf{X}$ has the value 1 if the gene i is directly associated with term j and 0 otherwise. This kind of scheme is similar to approaches used in computational linguistics where genes are replaced by documents and terms are replaced by words.

The linear kernel matrix would normally be defined as $\mathbf{XX}'$ except that this ignores relationships between the GO terms. Therefore, we create an additional matrix $\mathbf{P} \in \mathbb{R}^{t \times t}$ called the proximity matrix with each element p_{ij} representing the proximity (or similarity) between GO term i and j. Terms with a close relationship have values close to 1, with the diagonal elements $p_{ii} = 1$. The proximity between GO terms is based on the number of links (or distance) between them and is defined as

$$p_{ij} = \frac{1}{d_{ij} + 1} \tag{4}$$

where d_{ij} is the minimum distance between terms i and j over the hierarchy. The distance can be extracted from the Gene Ontology. Clearly $\mathbf{P}$ is symmetric.

The kernel matrix for the gene data, then, is defined as

$$\mathbf{K} = \mathbf{XPP}'\mathbf{X}' \tag{5}$$

The proximity matrix weights up GO terms in $\mathbf{X}$ that are close to one another. Proximity matrices have been used before for text kernels (eg. (Shawe-Taylor & Cristianini 2004)).

The Gaussian extension to this kernel is straightforward as alluded to in the last section. Working from equation (5) we simply scale by σ^2, take the exponent and normalise.

Consequently, in this paper we apply a linear kernel for comparing genes based on their GO terms using equation (5) and a Gaussian kernel based on the linear kernel.

3 Experiments

In this section we describe several experiments validating our method. First we describe the KEGG data set we used. Following this we give results for kernel PCA visualisation of the dataset using linear and Gaussian kernels.

3.1 Dataset

In this study we visualise a subset of genes from the Internet KEGG database. The KEGG dataset contains a list of genes classified into different classes of behaviour. The rationale behind using this dataset is to validate our approach with genes of known functional similarity.

The Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa et al. 2008) is a biological resource which aims to link genomes to the biological systems they govern. The resource takes the form of a series of interconnected databases of biological systems that interrelate (i) genes and proteins, (ii) chemical building blocks, (iii) molecular interaction pathway diagrams and (iv) hierarchies and relationships of biological objects. We are interested in the last of these databases (KEGG BRITE) which links genes into a functional hierarchy called the KEGG Orthology (KO). Importantly, this hierarchy is different to that of the Gene Ontology and has been constructed independently. This allows us to validate our visualisation by extracting genes that are similar according to their KO terms and then to visualise them using their GO terms. Consequently, our KEGG dataset contains a subset of genes from five classes of KO: ribosome (ko03010), RNA polymerase (ko03020), transcription (ko01210), pentose phosphate pathway (ko00030) and pentose and glucoronate interconversions (ko00040). Table 2 shows the interrelationships between the classes in terms of the parent KO terms to the selected classes. From these interrelationships we expect to see that classes 1, 2 and 3 are similar (with classes 2 and 3 more similar to one another than to class 1). Classes 4 and 5 should also be similar to one another but different to the other three classes.

A subset of genes was chosen from the lists given by KEGG. From the list of genes on KEGG, we chose those that were also accessible in the Gene Ontology database. The number of genes chosen for each class is given in Table 2.

3.2 Visualising the KEGG dataset

Initially we visualised the KEGG dataset with a linear kernel. This is equivalent to applying linear Principal Component Analysis to the dataset. For the genes listed in the KEGG dataset, we extracted the GO terms associated with the genes and generated $\mathbf{X}$ and $\mathbf{P}$ matrices as described in Section 2.3. Next, using equation (5) we generated the basic kernel matrix $\mathbf{K}$. Finally, we applied kernel PCA to this kernel as described in section 2.2.

Figure 2 shows a plot of the eigenvalues (λ) found for the principal components. These values are associated with the variance of the data explained by the corresponding principal component. As can be seen, the first principal component is very large compared to the rest. This is often a sign of one attribute dominating the principal component (or use of a covariance matrix rather than a correlation matrix). As described in section 2.2 above, we scale the kernel matrix which is equivalent to using a correlation matrix. Also, since we "fold" the original data attributes into kernel values we can no longer easily investigate the original principal component vectors to see whether one term dominates.

However, investigation of the terms associated with the genes suggests that the first principal component is a "size" component as described in Jolliffe (2004). "Size" components are found in (Jolliffe 2004) by checking the values of the principal component vector. When all (or most) values in the vector are strongly positive for each attribute then Jolliffe suggests that the principal component measures the general size of the data items. We cannot check the values of the principal component vector for each of the attributes because we are using a kernel–based approach rather than standard PCA and the original data attributes (ie. the terms) are hidden in the kernel value.

In Fig. 3 we plot the genes in the KEGG dataset according to principal component axes PC1 and PC2. The graph shows two groups of genes separated by principal component 1. The separation along PC1

Table 2: Description of the 5 classes of genes in our KEGG dataset. Column 1 contains the class identifier and the symbol used in our graphs. Column 2 gives the list of KO terms leading to the class and column 3 lists the number of genes in the class.

Class	KO structure	Count
1 +	genetic information processing : translation : **ribosome**	20
2 ×	genetic information processing : transcription : **RNA polymerase**	19
3 ∘	genetic information processing : **transcription**	11
4 □	metabolism : carbohydrate metabolism : **pentose phosphate pathway**	11
5 •	metabolism : carbohydrate metabolism : **pentose and glucoronate interconversions**	8

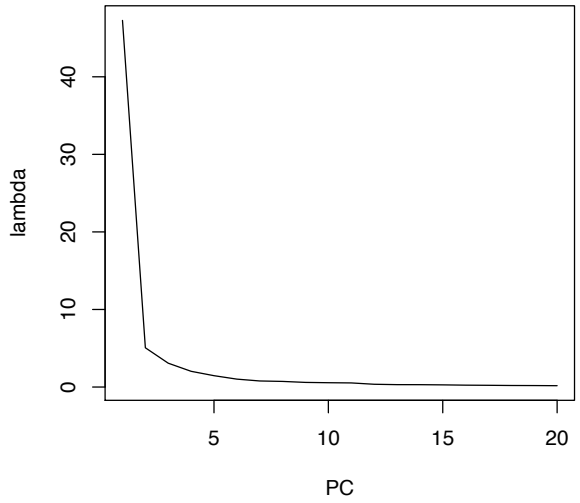

Figure 2: Sorted eigenvalues λ associated with the first 20 principal components of the visualisation of the KEGG dataset using the linear kernel. The first eigenvalue is very high compared to the rest.

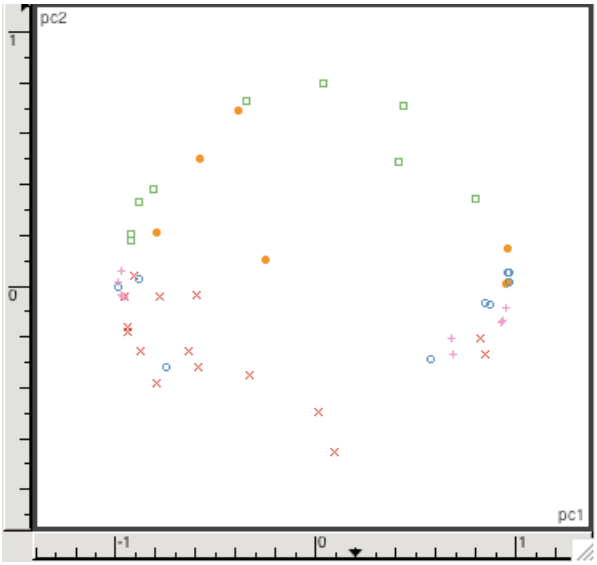

Figure 3: Plot of genes from KEGG dataset according to PC1 and PC2 using the linear kernel. Key: + = ribosome, × = RNA polymerase, ∘ = transcription, □ = pentose phosphate pathway, • = pentose and glucoronate interconversions.

does not reflect the KEGG class of the gene, although PC2 does discriminate by class to some extent. Investigation of the genes in each of the two clusters along PC1 show that genes on the left hand side have fewer terms associated with them compared to genes on the right hand side and that genes in the middle have a count of associated terms mid way between the extremes. For example, the four genes on the extreme left hand side of Fig. 3 are NUSG (with one associated GO term) and RPMF, RPSF and RPLD (each with 2 GO terms). The four genes on the extreme right hand side are RHO (27 terms), Elp3 (20 terms), Eda (18 terms) and Clpx (17 terms). This suggests that our interpretation of PC1 as a "size" component is the correct one for this dataset.

Consequently, in Fig. 4 we plotted the genes according to the next two principal components: PC2 and PC3. This figure shows that PCs 2 and 3 result in a visualisation that reflects the classes of genes. Genes that are similar to one another (ie. fall within a class) group together and those that are different are generally separated. Genes in classes 4 (□, pentose phosphate pathway) and 5 (•, pentose and glucoronate interconversions) group very closely together as expected. These are generally far apart from the genes in the other classes except for some overlap at the origin. Genes in class 1 (+, ribosome) cluster tightly and there is a closer relationship between genes in class 2 (×, RNA polymerase) and class 3 (∘, transcription) than the translation related genes of class 1. Principal component 2 contrasts the carbohydrate metabolism related genes with the genetic information processing related genes. This accords well with what we would *a priori* expect to be the main variance in the genes. Principal component 3 then contrasts the different kinds of genetic processing related genes.

Next, we applied the Gaussian kernel to the linear kernel $\mathbf{K}$ generated above as described in section 2.3. We explored various settings of the σ parameter and empirically found that when $\sigma = 3$ it starts showing different clusters but $\sigma = 10$ gave reasonable visualisations where the genes did not end up on top of one another or spread out like the linear kernel. Plotting the eigenvalues (λ) does not make sense in this case because the principal components relate to the infinite dimensional feature space induced by the Gaussian kernel. Figure 5 plots the genes according to principal components 1 and 2. The genes at the ends of the tails in Fig. 5 are the same as those in Fig. 3 which again suggests that the first principal component contrasts the number of GO terms associated with the genes. Specifically, the gene marked ∘ at the end of the left hand tail of Fig. 5 is RHO (27 terms). The next is ELP3 (20 terms) followed by EDA (18 terms) and Clpx (17 terms). These are the same as in the linear diagram and are ordered by the number of terms. At the other end are NUSG (1 term), RPMF, RPSF and RPLD (2 terms).

Figure 6 graphs the genes by the second and third principal components. As with the linear case, the genes cluster mostly according to functionality and KEGG class. Genes that were at the origin of the linear graph (Fig. 4), however, have moved to the left hand "spike" of the Gaussian graph (Fig. 6). The grouping of RNA polymerase genes (×) at the top left

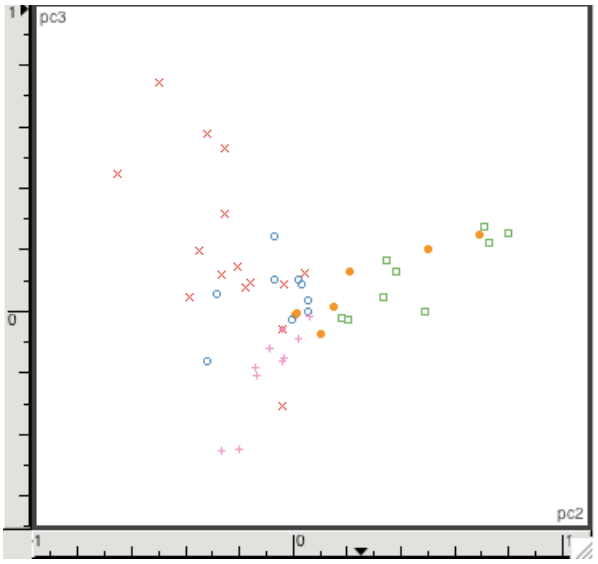

Figure 4: Plot of genes from KEGG dataset according to PC2 and PC3 using the linear kernel. Key: + = ribosome, × = RNA polymerase, ∘ = transcription, □ = pentose phosphate pathway, • = pentose and glucoronate interconversions.

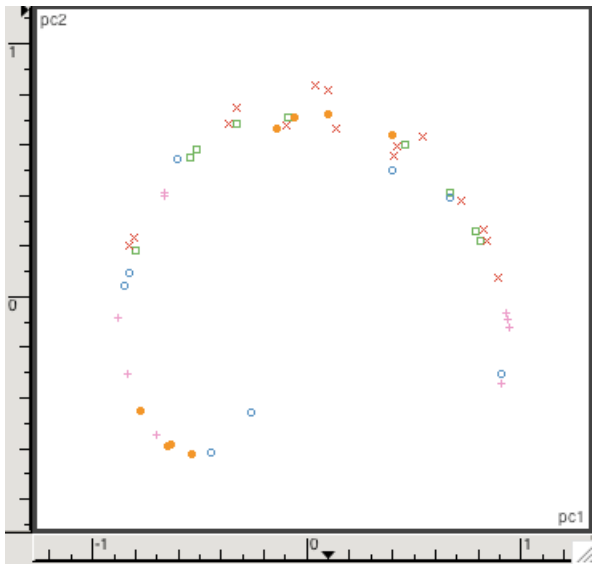

Figure 5: Plot of genes from KEGG dataset according to PC1 and PC2 using the Gaussian kernel with $\sigma = 10$. Key: + = ribosome, × = RNA polymerase, ∘ = transcription, □ = pentose phosphate pathway, • = pentose and glucoronate interconversions.

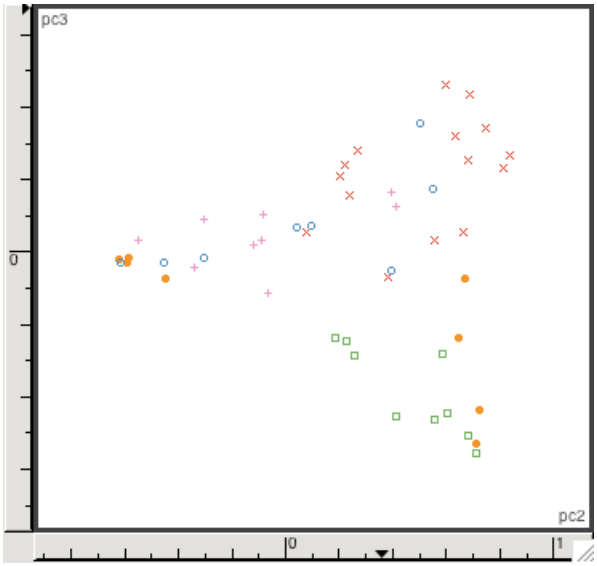

Figure 6: Plot of genes from KEGG dataset according to PC2 and PC3 using the Gaussian kernel with $\sigma = 10$. Key: + = ribosome, × = RNA polymerase, ∘ = transcription, □ = pentose phosphate pathway, • = pentose and glucoronate interconversions.

of Fig. 4 still cluster separately (top right of Fig. 6) and the small group of pentose phosphate pathway (□) and pentose and glucoronate interconversions (•) genes at the right hand side of Fig. 4 have grouped more closely in the bottom right of Fig. 6.

We also examined visualisations of the data with different σ values. Specifically, we examined $\sigma = 0.1$, 1, 1.5, 2, 2.5, 3, 5, 7, 25, 50, 75, 100, 500, 1000. Using the value of 0.1 condensed the genes on top of one another. At value 2 genes starts to open up and at $\sigma = 3$ the genes start to make shape.Values of 50, 75, 100, 500 and 1000 look the same as the linear kernel, as expected. Figures 7 and 8 show visualisations using $\sigma = 1$ of the first two principal components and the second two components respectively. Many of the genes sit on top of one another so jitter (small random adjustments) has been added to the genes on these figures. With $\sigma = 1$, the first principal component no longer seems to act as a "size" component. However, the first two principal components contrast most of the ribosome (+) and transcription (∘) genes from the others, as does the third principal component. The fourth principal component expresses the variance associated to the RNA polymerase (×) genes. Although not shown, it is not until later principal components that the carbohydrate metabolism genes get distinguished from the others. Since there are many more genetic information processing genes in the dataset compared to the carbohydrate metabolism genes (see Table 2) it is expected that the earlier principal components are concerned with this variation. Also, the narrower focus of the σ gives a finer grained distinction between genes. This suggests that use of the Gaussian kernel rather than the linear kernel (ie. ordinary PCA) is important for distinguishing between different genes. The statistical properties of the Gaussian kernel are useful to the visualisation. Choice of the σ parameter is anticipated to be problematic for datasets where the relationship between genes is unknown and tuning of this parameter will be the subject of a future investigation.

Finally, we also explored use of a different distance function between terms to generate the proximity matrix $\mathbf{P}$. Rather than simply counting the links between the terms using equation (4) we instead

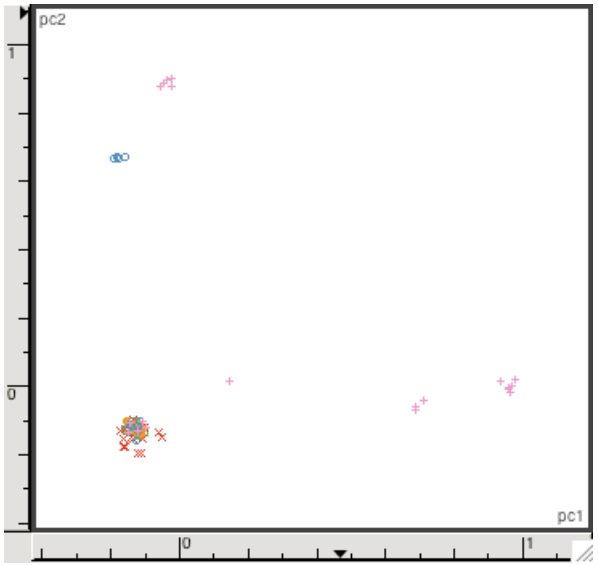

Figure 7: Plot of genes from KEGG dataset according to PC1 and PC2 using the Gaussian kernel with $\sigma = 1$ with a small amount of jitter applied to the values. Key: $+$ = ribosome, $\times$ = RNA polymerase, $\circ$ = transcription, $\square$ = pentose phosphate pathway, $\bullet$ = pentose and glucoronate interconversions.

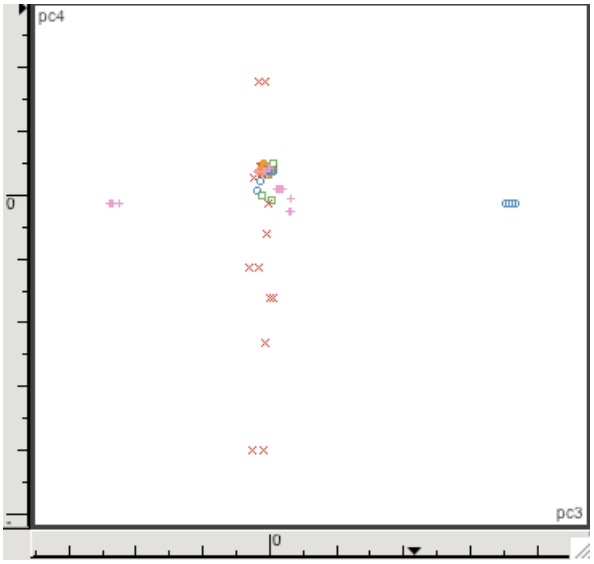

Figure 8: Plot of genes from KEGG dataset according to PC3 and PC4 using the Gaussian kernel with $\sigma = 1$ with a small amount of jitter applied to the values. Key: $+$ = ribosome, $\times$ = RNA polymerase, $\circ$ = transcription, $\square$ = pentose phosphate pathway, $\bullet$ = pentose and glucoronate interconversions.

weighted down long paths. The motivation behind doing this is to emphasise close relationships between genes rather than distant relationships (where terms are related only through the very high level and overly general GO terms). The discounting distance function is defined as

$$d'_{ij} = \sum_{k=0}^{d_{ij}-1} c^k \qquad (6)$$

where d_{ij} is the distance between terms reported by the Gene Ontology and $c \in [0, 1]$ is a discounting constant set to 0.9 in our experiments. However, the visualisations were very similar for both the linear and Gaussian kernels so we do not show them here. A more appropriate way to discount the distance would be to weight down the distance to the closest parent of the terms i and j following equation (6). However, the distance to the closest common parent term was not easily accessible from the Gene Ontology database, so we did not pursue the approach.

4 Future Work

There are several areas that we think warrant further investigation. The most important involves investigating how to decide whether one visualisation is "better" than another. This is useful because it allows tuning of parameters and should be used to decide on whether one algorithm is better than another. Along these lines we plan to investigate the "trustworthiness" metric of Venna & Kaski (2007) which uses notions based on precision and recall to compare visualisations of microarray data. Once a "ruler" for comparing visualisations is established we can turn to tuning of the σ parameter for the Gaussian kernel. We plan also to compare our similarity measure with others, most notably the information–theoretic one of Speer et al. (2005) to see which gives better visualisations. We plan also to examine other kernel–based visualisation methods and variants of the discounting distance function given in equation (6). Finally, but not least importantly, we will visualise datasets from experiments by biologists to gain a better understanding of their needs and the questions they want answered.

5 Conclusion

This paper describes an approach to visualising genes using kernel Principal Component Analysis. We define a specialised linear kernel based on computational linguistics and a Gaussian variant that was able to find similarities between genes using terms from the Gene Ontology. Functional relationships between genes chosen from classes within KEGG were correctly visualised with the technique.

References

Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J. et al. (2000), 'Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.', *Nat Genet* **25**(1), 25–9.

Beißbarth, R. & Speed, T. (2004), 'GOstat: finding statistically over expressed Gene Ontologies within groups of genes', *Bioinformatics* **20**(9), 1464–1465.

Cheng, J., Cline, M., Martin, J., Finkelstein, D., Awad, T., Kulp, D. & Siani-Rose, M. A. (2004), 'A knowledge–based clustering algorithm driven by Gene Ontology', *Journal of Biopharmaceutical Statistics* **13**(3), 687–700.

Fröhlich, H., Speer, N., Poustka, A. & Beißbarth, T. (2007), 'GOSim–An R-package for computation of information the-oretic GO similarities between terms and gene products', *BMC Bioinformatics* **8**, 166.

Gat-Viks, I., Sharan, R. & Shamir, R. (2003), 'Scoring clustering solutions by their biological relevance', *Bioinformatics* **19**(18), 2381–2389.

Haykin, S. (1999), *Neural networks: a comprehensive foundation*, 2nd edn, Prentice–Hall.

Jolliffe, I. T. (2004), *Principal Component Analysis*, Springer Series in Statistics, second edn, Springer, New York.

Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T. & Yamanishi, Y. (2008), 'KEGG for linking genomes to life and the environment', *Nucleic Acids Research* **36**, 480–484.

Kennedy, P. J. & Simoff, S. J. (2003), CONGO: clustering on the Gene Ontology, *in* 'Proceedings Australasian Data Mining Workshop', pp. 181–198.

Lee, I.-Y., Ho, J.-M. & Chen, M.-S. (2005), CLUGO: a clustering algorithm for automated functional annotations based on Gene Ontology, *in* 'Proceedings of Fifth IEEE International Conference on Data Mining', IEEE.

Lee, S., Hur, J. & Kim, Y. (2004), 'A graph-theoretic modeling on GO space for biological interpretation of gene clusters', *Bioinformatics* **20**(3), 381–388.

Leslie, C., Kuang, R. & Eskin, E. (2004), Inexact matching string kernels for protein classification, *in* B. Schölkopf, K. Tsuda & J.-P. Vert, eds, 'Kernel methods in computational biology', MIT Press, pp. 95–112.

Liu, H., Hu, Z.-Z. & Wu, C. H. (2005), 'DynGO: a tool for visualizing and mining of Gene Ontology and its associations', *BMC Bioinformatics* **6**(201).

Mathur, S. & Dinakarpandian, D. (2007), 'A New Metric to Measure Gene Product Similarity', *Bioinformatics and Biomedicine, 2007. BIBM 2007. IEEE International Conference on* pp. 333—338.

Müller, K., Mika, S., Rätsch, G., Tsuda, K. & Schölkopf, B. (2001), 'An introduction to kernel–based learning algorithms', *IEEE Transactions on Neural Networks* **12**, 181–201.

Popescu, M., Keller, J., Mitchell, J. & Bezdek, J. (2004), Functional summarization of gene product clusters using Gene Ontology similarity measures, *in* 'Proceedings of IEEE Intelligent Sensors, Sensor Networks and Information Processing Conference', IEEE, pp. 553–558.

Sanfilippo, A., Posse, C., Gopalan, B., Riensche, R., Beagley, N., Baddeley, B., Tratz, S. & Gregory, M. (2007), 'Combining Hierarchical and Associative Gene Ontology Relations With Textual Evidence in Estimating Gene and Gene Product Similarity', *Nanobioscience, IEEE Transactions on* **6**(1), 51–59.

Shah, N. H. & Fedoroff, N. V. (2004), 'CLENCH: a program for calculating Cluster ENriCHment using the Gene Ontology', *Bioinformatics* **20**(7), 1196–1197.

Shawe-Taylor, J. & Cristianini, N. (2004), *Kernel Methods for Pattern Analysis*, Cambridge University Press, Cambridge.

Speer, N., Fröhlich, H., Spieth, C. & Zell, A. (2005), Functional grouping of genes using spectral clustering and gene ontology, *in* 'Proceedings of the IEEE International Joint Conference on Neural Networks', pp. 298–303.

Venna, J. & Kaski, S. (2007), 'Comparison of visualization methods for an atlas of gene expression data sets', *Information Visualization* **6**, 139–154.

Zeeberg, B., Feng, W., Wang, G., Wang, M., Fojo, A., Sunshine, M., Narasimhan, S., Kane, D., Reinhold, W., Lababidi, S. et al. (2003), 'GoMiner: a resource for biological interpretation of genomic and proteomic data', *Genome Biol* **4**(4), R28.

Conferences in Research and Practice in
Information Technology

Volume 87

Data Mining and Analytics 2008

Data Mining and Analytics 2008

Proceedings of the
Seventh Australasian Data Mining Conference (AusDM'08),
Glenelg, South Australia, 27-28 November, 2008

John F. Roddick, Jiuyong Li, Peter Christen and
Paul Kennedy, Eds.

Volume 87 in the Conferences in Research and Practice in Information Technology Series.
Published by the Australian Computer Society Inc.

Published in association with the ACM Digital Library.

Data Mining and Analytics 2008. Proceedings of the Seventh Australasian Data Mining Conference (AusDM'08), Glenelg, South Australia, 27-28 November, 2008

Conferences in Research and Practice in Information Technology, Volume 87.

Editors: **John F. Roddick**
School of Computer Science, Engineering and Mathematics
Flinders University
GPO Box 2100, Adelaide, SA, 5001, Australia
Email: john.roddick@flinders.edu.au

Jiuyong Li
School of Computer and Information Science
University of South Australia, Mawson Lakes
GPO Box 2471, Adelaide, SA, 5001, Australia
Email: jiuyong.li@unisa.edu.au

Peter Christen
Department of Computer Science
Faculty of Engineering and Information Technology
The Australian National University
Canberra ACT 0200 Australia
Email: peter.christen@anu.edu.au

Paul J. Kennedy
Faculty of Engineering and Information Technology
University of Technology, Sydney
Broadway, NSW, 2007, Australia
Email: paulk@it.uts.edu.au

Series Editors:
Vladimir Estivill-Castro, Griffith University, Queensland
John F. Roddick, Flinders University, South Australia
Simeon Simoff, University of Western Sydney, NSW
`crpit@csem.flinders.edu.au`

Publisher: Australian Computer Society Inc.
PO Box Q534, QVB Post Office
Sydney 1230
New South Wales
Australia.

Conferences in Research and Practice in Information Technology, Volume 87
ISSN 1445-1336
ISBN 978-1-920682-68-2

Printed November 2008 by Flinders Press, PO Box 2100, Bedford Park, SA 5042, South Australia.
Cover Design by Modern Planet Design, (08) 8340 1361.

The *Conferences in Research and Practice in Information Technology* series aims to disseminate the results of peer-reviewed research in all areas of Information Technology. Further details can be found at http://crpit.com/.