COMBat: Visualizing Co-Occurrence of Annotation Terms

Remko B. J. van Brakel
Eindhoven University of Technology

Mark W. E. J. Fiers
Center for the Biology of Disease (VIB), Leuven

Christof Francke
HAN University of Applied Sciences, Nijmegen

Michel A. Westenberg
Eindhoven University of Technology

Huub van de Wetering*
Eindhoven University of Technology

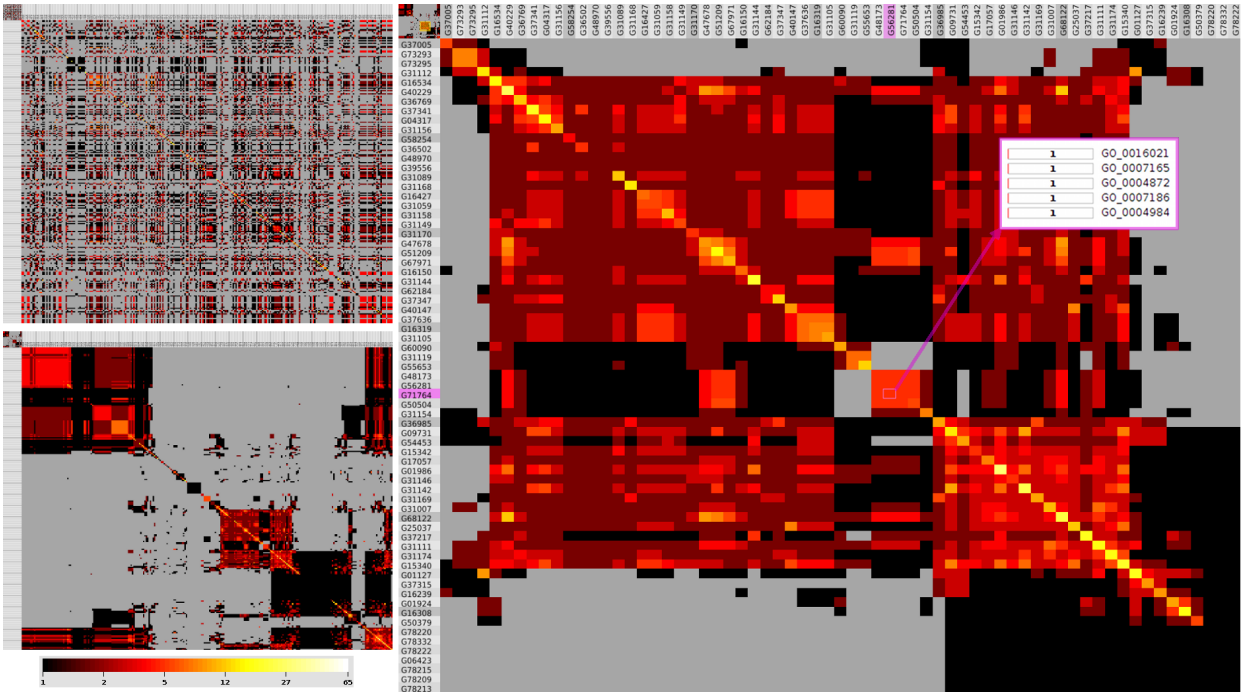


Figure 1: COMBat applied to a dataset with 268 genes—annotated with in total 634 gene ontology (GO) identifiers [1]—on both the rows and the columns. Three property matrices are shown, in which each cell contains the set of GO identifiers that the corresponding genes have in common. Each cell is colored by the number of elements in the set as given by the colormap (see legend in the lower-left corner). The cell is gray if no data is associated with it. In the top-left corner, the initial, un-sorted property matrix is shown. The image just below it shows the same data, but now sorted using a similarity sort. Typically, the resulting blocks of constant color correspond to genes with the same set of GO identifiers. At the right, more details are visible after zooming in on a region somewhat to the right and below the center of the matrix; in the top-left corner of the property matrix, a minimap provides information about the location and size of this zoomed region relative to the whole matrix. The cell corresponding to gene G71764 and G56281 is highlighted, and the GO identifiers that they have in common are shown in a separate inset.

Abstract

We propose a visual analysis approach that employs a matrix-based visualization technique to explore relations between annotation terms in biological data sets. Our flexible framework provides various ways to form combinations of data elements, which results in a co-occurrence matrix. Each cell in this matrix stores a list of items associated with the combination of the corresponding row and column element. By re-arranging the rows and columns of this matrix, and color-coding the cell contents, patterns become visible. Our prototype tool COMBat allows users to construct a new matrix on the fly by selecting subsets of items of interest, or filtering out uninteresting ones, and it provides various additional interaction techniques. We illustrate our approach with a few case studies concerning the identification of functional links between the presence of particular genes or genomic sequences and particular cellular processes.

Index Terms: J.3 [Computer Applications]: Life and Medical Sciences—Biology and genetics

1 Introduction

The rapid advance in high throughput -omics technology has resulted in a wealth of 'relational' data on biomolecules. The data, as made available by several comprehensive public databases, comprise for instance: protein sequences (and thereby genes) linked to functional domains [22, 30], genes linked to proteins, to reactions, to compounds and to pathways [4, 20], reactions linked to enzymes and co-factors, to genes and to organisms [25], signal molecules to regulators, to regulatory sequences and to controlled genes [18], or proteins functionally and/or physically linked to other proteins and to organisms [7]. A central effort in the exploitation of -omics data

*e-mail: christof.francke@han.nl, mark.fiers@cme.vib-kuleuven.be, {m.a.westenberg,h.v.d.wetering}@tue.nl

IEEE Symposium on Biological Data Visualization 2013
October 13 - 14, Atlanta, Georgia, USA
978-1-4799-1659-7/13/$31.00 ©2013 IEEE

is the identification of the biological role of a particular biomolecule in an organism of choice or, the other way around, the identification of the biomolecules that play a role in a particular organismal physiology.

One of the main strategies in such an effort is the extraction and correlation of the relevant information on function from the relational databases. In case only two types of data should be compared, the analysis is relatively straightforward and can be performed computationally using scripts or visually by, for instance, tag clouds [32]. However, in practice, an analysis including multiple types of data often appears non-trivial and involves cumbersome rounds of sorting and highlighting of rows and columns in spreadsheets before possibly relevant types of data are identified. Thus there is a clear need for (visualization) tools that support and speed up the comprehensive analyses of multiple types of data. In this paper, we propose an analysis approach that employs a matrix-based visualization technique. Our contributions are:

Property matrix: A visualization approach that provides a matrix-based view of relations between data elements and data types of heterogeneous datasets. Any combination of data elements can be assigned to the axes of the matrix for visual analysis of co-occurrences.

COMBat: A tool (Co-Occurrence Matrix Browser for Annotation Terms) that implements the proposed approach, and which supports various interaction techniques including zooming, panning, filtering, selection, and sorting.

Case studies: A demonstration of our approach and tool to support the identification of: (i) the evolutionary conserved role in cellular physiology of a particular regulatory protein Sigma-54; (ii) the regulatory module related to high glutamine levels across different species of bacteria; and (iii) the potential role of a highly repetitive genome sequence element in *Lactococcus lactis cremoris SK11*, a bacterium used to produce cheese.

2 Background and Requirements

Biological databases exist in various formats, but we can abstract away from the specific details, and think of annotation data provided as a table. The rows in this table contain items that have several attributes, which correspond to the columns. Typically, a table has a header, which provides names to the attributes. We distinguish three data types: *sources*, *annotation types*, and *annotation elements*. The sources are unique basic items, such as sequences, genomes, species, and locus tags. The annotation types are the names of the annotation attributes, for example, start position of a sequence, genome size, phylum, and synonym. The annotation elements, or simply *annotations*, are the actual values that the sources have for their respective annotation types. In terms of the table: a row corresponds to a source, a column header to an annotation type, and the contents of a cell to an annotation.

Table 1: Possible data type combinations for the property matrix dimensions and the resulting contents of the cells.

row	column	cell contents
S_1	S_2	list of annotations in both S_1 and S_2
	T	list of annotations of type T in S_1
	A	number of appearances of A in S_1
T_1	T_2	list of sources with both T_1 and T_2
	A	list of sources with both A and T_1
A_1	A_2	list of sources with both A_1 and A_2

$S =$ source, $T =$ annotation type, $A =$ annotation

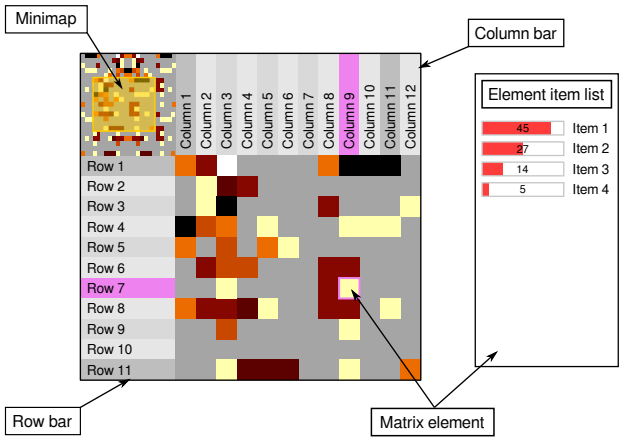


Figure 2: Concept drawing of the property matrix visualization. The row and column bar display row and column labels, respectively. The matrix elements are color-coded depending on their content. One of the matrix elements is highlighted by a pink border, and its contents is shown in a separate element item list view. The row and column labels of the highlighted matrix element are highlighted as well, to ease navigation. The small bar charts in this view indicate the number of times the particular item occurs. The minimap in the top-left corner provides the location and size of the current view within the whole dataset.

Our main goal is to provide various ways to form combinations of these data types interactively. The visualization of these combinations should be intuitive and easy to navigate. The observer must be able to detect patterns in the data by rearrangement and grouping options. Further, to drill down to interesting details and obtain biological insight, filtering for specific text or values has to be supported. Finally, for a given combination of data types, it must be possible to visualize all items associated with that combination.

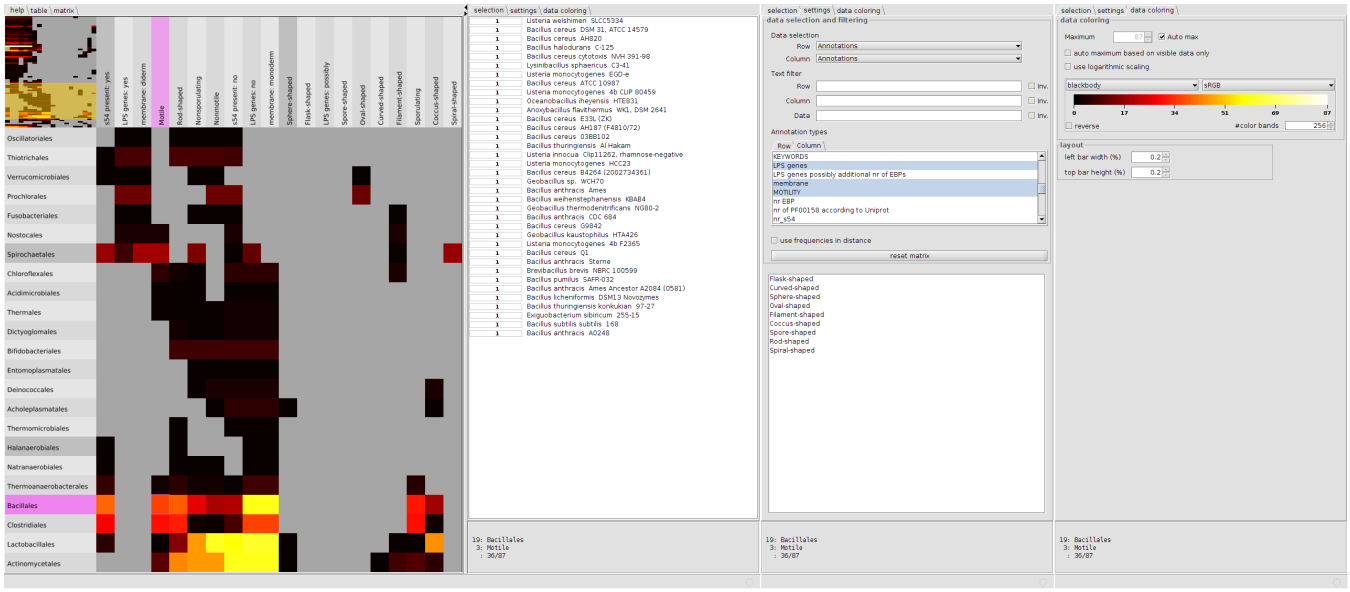
Since annotation data is heterogeneous, our approach should be able to handle mixed forms of annotations, including text and nominal values, and treat them in a uniform way. Generation of additional attributes, based on the dataset should be possible as well. For example, numerical attributes could be binned to make comparison easier. Extraction of keywords from multi-word annotation texts is another example, which would help to find common words used across the data types.

In this paper, we limit our approach to non-hierarchical annotation data, as this covers a large amount of important data related to function. Examples of annotations of this type are: presence/absence of X, protein location, domain function (Pfam [22]), clusters of ortholog groups of proteins (COGs [28]), trivial gene names, protein names, EC numbers, morphology-related features, protein partners, associated genes in a network, to name a few.

For visualization, a matrix-based approach is a natural choice and well suited for our purpose [2]. It provides a uniform space that is intuitive and easy to navigate, and it is, to some extent, scalable to large datasets.

3 Related Work

The concept of matrix visualization was introduced by Bertin [2]. Since then, matrix visualization has become a versatile technique to explore associations of thousands of items and their interactions [34]. Key to revealing structure is to permute the rows and columns by suitable reordering algorithms, referred to as *seriation* methods. This can be done interactively by allowing a user to reorder the matrix according to one attribute (row or column) [23] to explore various orderings. However, this allows only one-dimensional sorting,



(a) property matrix visualization (b) selection panel (c) settings panel (d) color panel

Figure 3: COMBat user interface elements. (a) Property matrix visualization; a matrix element and corresponding row and column labels are highlighted. (b) Selection panel containing information about the hovered (highlighted) matrix element. (c) Settings panel which controls the data types mapped to rows and columns, and which offers selection and filtering options. (d) Color panel which allows color map choice and manipulation.

and it also does not allow sorting on several attributes simultaneously. A large number of automatic seriation techniques has been proposed over the past decades, and detailed reviews can be found in Henry *et al.* [13], Wu *et al.* [34], and Wilkinson *et al.* [31].

In bioinformatics, the matrix view is known as a *heatmap*, and has been popularized by the work of Eisen [5] on visualizing clustering results of gene expression experiments. In biochemistry, matrix views are used to visualize protein sequence similarity, where they are known as dot plots [10]. In this case, the visualization is simply binary: a cell is colored black if both sequences are identical in that position and white otherwise; reordering is not used either, as the sequences are assumed to be aligned.

In the computational statistics domain, the software package GAP [33] implements matrix visualization and various clustering algorithms. The tool focuses strongly on clustering a given initial matrix, and exploring the clusters in the matrix context. It requires domain knowledge in clustering to use its full power, which makes it less suitable to be used by domain experts in biology.

Our proposed approach is different in the sense that it can handle heterogeneous data (annotations can be a mix of numerical, categorical, or textual data), and that we allow the users to construct a new matrix on the fly by selection and filtering. To our knowledge, this flexibility is not offered by any other approach.

4 Visualization Approach

4.1 Property matrix

The property matrix is an interactive visualization to explore the relations between data elements and data types of a dataset. In the property matrix, the data types introduced in Section 2 can be assigned in various combinations to the rows and columns for cross referencing. Table 1 provides an overview of all possible combinations. To make this more concrete, suppose the dataset consists of a list of N species (sources $S = \{S_1, S_2, ..., S_N\}$) annotated with phylum (annotation type T_1) and genome size (annotation type T_2). The corresponding values for these types are M phyla annotations in $P = \{P_1, P_2, ..., P_M\}$ and three genome size annotations in $C = \{\text{small}, \text{medium}, \text{large}\}$. The combination species on the rows and annotation types on the columns (S vs. $\{T_1, T_2\}$) results in an $N \times 2$ matrix, in which each cell contains exactly one item, namely, the corresponding phylum in the first column and the genome size in the second column. Another example is the combination P vs. C of the values of the annotation types phylum and genome size. This combination results in an $M \times 3$ matrix. According to the A vs. A entry in Table 1, each cell in this matrix contains a list of species of a given phylum (row) and genome size (column).

4.2 Property matrix visualization

The property matrix visualization is composed of a horizontal and vertical bar with a dot-plot [10] in between, see Fig. 2. The vertical and horizontal bars are called *row bar* and *column bar*, respectively, and they are used to draw the labels of the corresponding data types. Every row or column element displays a single line of text. The column element's text is rotated by 90 degrees, because this orientation uses less screen space and provides better readability than a downward cascade of letters [3]. The texts are clipped at the right or top side of the bar elements, when needed. To make the individual rows and columns better visible, their background colors alternate between two shades of gray. Every 10th row or column is given a darker shade of gray, which helps the user to make a rough estimate of the total number of rows and columns in the matrix.

The dot-plot elements, or matrix elements, provide a color-coded representation of their contents. If the matrix elements contain a list, then the length of this list is mapped to a color. If the element contains a numerical value, then this is used directly for color mapping. If there is no data in the element, it is colored gray (or some other color that clearly differs from the ones used in the color map). The actual content of the matrix element is provided in a separate view. This view provides a list of all items, and shows a small bar chart which displays the number of times that item occurs.

The top-left corner (between the two bars) provides an overview that indicates which part of the dataset is currently visible within the visualization. The light gray area represents the dataset as a

whole, and a (smaller) dark gray rectangle shows the location and size of the visible part of the matrix.

4.3 Property matrix sorting

Rows and columns of the property matrix can be reordered without affecting the data represented by the visualization. Consequently, sorting schemes may be applied in order to reveal the dataset's structure and to improve readability of the visualization.

Sorting rows and columns by label. The most simple form of sorting only concerns the row or column labels in the two bars. The property matrix allows to sort either alphabetically or numerically.

Sorting by a single row or column. Sorting by the numerical values in a single row or column (and the remaining rows/columns accordingly) can be used to investigate the distribution of a single property.

Sorting rows and columns by frequency. This approach counts for each row (column) how many cells in that row (column) contain data. The rows (columns) are then sorted in increasing order. When looking at the distribution of annotation terms over genomes, for example, this sorting can be used to put the most frequently occurring terms close together (see Fig. 8 from one of the case studies).

Sorting the whole matrix by similarity. There are many similarity measures that can be used to reveal clusters in the matrix [33, 34]. We use an approach that is inspired by the Gray code [6, 11], and which sorts the matrix by minimal difference between rows and/or columns. The sorting algorithm treats the matrix elements as if they are binary. Each row (column) is represented as a sequence of symbols 0 and 1, depending on whether the corresponding matrix elements contain data or not. As a similarity measure, we use the Hamming distance [12], which gives the number of positions at which the binary patterns are different. As a result, rows (columns) that are similar in their binary pattern are grouped together. This sorting technique was applied in Fig. 1. For an $n \times m$ matrix, the time complexity of the sorting algorithm is $O(n^2m + m^2n)$. The result of the sorting algorithm is not unique, because it is dependent on the starting point. Also, an optimal sorting is hard to achieve, because that problem is related to the minimum linear arrangement problem, which is known to be NP-hard [9].

5 COMBat Tool

The property matrix visualization is implemented in a tool called COMBat. It was built in Java, and it relies on the Java OpenGL (JOGL) bindings for rendering the property matrix. A screenshot of the most important user interface elements is shown in Fig. 3. In the following, we will explain these in more detail.

5.1 Supported data formats

COMBat accepts flat text files in the form of tab-separated values (TSV) and in GFF [24, 26] format. For TSV-files, the first row (or the table header) provides the annotation types. The first column is treated as the source, and each subsequent column becomes an annotation type.

DNA sequence data are commonly annotated in GFF format. This is a simple flat file consisting of 9 tab-separated columns. The sequence identifier in the first column is treated as the source. The last column contains an arbitrary amount of annotation attributes, specified as combinations tag = value, separated by semicolons. COMBat relies on the BioJava [14] libraries to read and interpret these files.

5.2 Additional annotations

COMBat can generate additional annotation types based on the input data. For GFF files, two additional annotation types are generated from the structured annotation in column 9. Annotations with

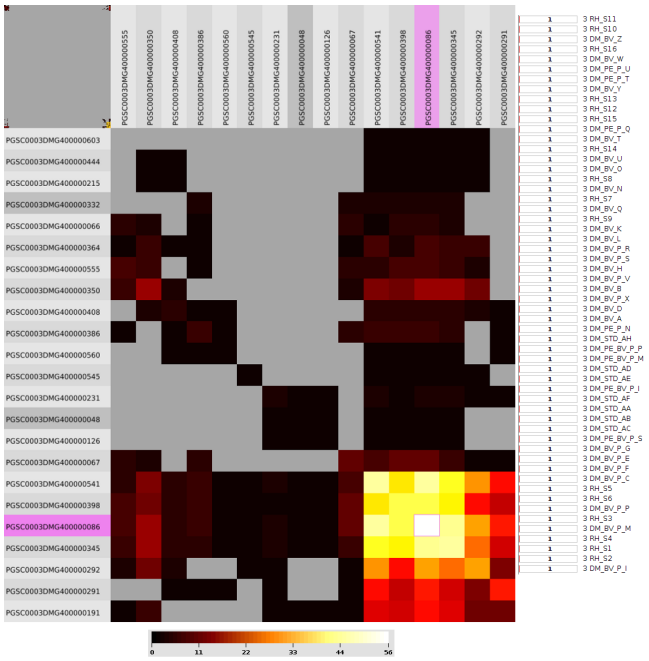


Figure 4: Highly expressed genes in multiple tissues, extracted by a filtering on gene expression annotation. The gene expressions have been discretized in 5 bins, and only the top 2 bins are shown here. The highlighted cell shows a sequence that has high gene expression in a large number of tissue types shown in the list at the right.

type SPECIES are constructed from species declarations that appear in texts of other annotations. Simple keyword extraction is applied on the values of special tags and the resulting keywords are added as annotations of type KEYWORDS. Due to the flexibility in the GFF standard, it is unclear which tags should be considered special. Currently, only the tag "description" is supported as such.

Numerical annotations may be binned for ease of comparison. All annotations a of a binned type t are added as an annotation '$n\,t$' of the additional annotation type BINS, where n is the bin number of annotation t. For example, `[RH_ABC]=622.3` may result in `BINS=2 [RH_ABC]`, when the bin size is set to 300. The number of bins and the size of the bins can be chosen by the user when a data file is opened. COMBat automatically determines which annotation types are eligible for binning, and presents a dialog to the user in which the binning parameters can be modified.

5.3 Interaction and filtering

COMBat supports the standard interactions, such as zooming and panning (see also the accompanying video). The property matrix allows for zooming the rows and columns independently by mouse wheel rotation in the row and column bar, respectively. Synchronized zooming occurs when the mouse wheel is rotated if the mouse hovers over a matrix element. Panning works by holding the left mouse button down in the row or column bar and moving the mouse. The same panning operations can also be performed in the minimap view. This view additionally allows the user to select a rectangular region, which then becomes the new view of the property matrix.

If the mouse hovers over a matrix element, the corresponding row and column labels in the side bars are highlighted in a purple color, and a purple boundary is drawn around the hovered element (Fig. 3(a)). This makes it easier to navigate the property matrix. The items that are stored in the matrix element are shown in a list view in the selection panel. For the element hovered, these items

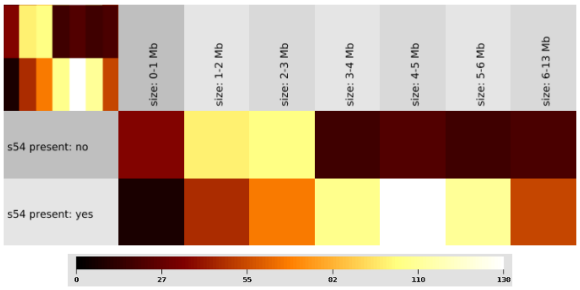


Figure 5: Relation between genome size and presence of Sigma-54. The number of species in each cell is color coded (see legend at the bottom).

are four species from the Legionella genus, see Fig. 3(b). At the bottom of the panel, some more information about the hovered item is shown.

Data filters provide a means to manipulate which data is shown in the property matrix. Annotation types can be included or excluded from the visualization by selecting or deselecting them in the settings panel (Fig. 3(c)). Another filter is a regular-expression-based (case-insensitive) text filter, which can be configured to include or exclude row/column/data elements that contain a particular text pattern. This filter is useful for reducing the number of displayed rows and columns, or to search for specific properties within the elements.

The data coloring tab (Fig. 3(c)) contains options to choose and manipulate the color mapping. By default, a heated-body color map is used. Optionally, the data values can be scaled logarithmically to obtain more contrast in skewed distributions. The color legend provides labeled tick marks, which makes interpretation of the colors easier.

5.4 Binning and filtering example

We use the publicly available data from the Potato Genome Sequencing Consortium [29] that consists of 500 genes of the potato (*Solanum tuberosum* L.) and corresponding annotations including gene expression levels measured in various tissues. The data set contains measurements for two genotypes, *S. tuberosum* group Phujera DM1-3 516 R44 and *S. tuberosum* group Tuberosum RH89-039-16, represented on a logarithmic scale.

We convert the gene expression levels into discrete 'annotations' by applying binning in 5 bins, which results in annotations of the form "`i [label]`", where $i = 0,\ldots,4$, and `label` is a tissue name. Please note that this example merely illustrates how binning can be used, and that we do not propose to analyze gene expression data using COMBat. Specialized tools exist for this purpose. Genes that are highly expressed in multiple tissues can now be revealed by putting the sequences on both the row and column bars, and filtering the data of type BINS with the regular expression "`^[34]+`" to keep only the high expression values corresponding to bins 3 and 4. After sorting the matrix, we can see that only a small number of genes, grouped in the lower right corner satisfy the filtering criterion. Zooming in on this region reveals more details, see Fig. 4. At the highlighted cell, we see that the sequence PGSC0003DMG400000086 has a high expression in a large number of tissue types (see the list at the right) in both genotypes.

6 Case Studies

We demonstrate the effectiveness of our approach through three case studies. In the first two, we test the ease and speed of the approach by re-analyzing data tables that were used to support the identification of the conserved biological role of particular regulators (see [8, 15]). In the third case study, we use our approach to help attribute a potential function to conserved sequence elements.

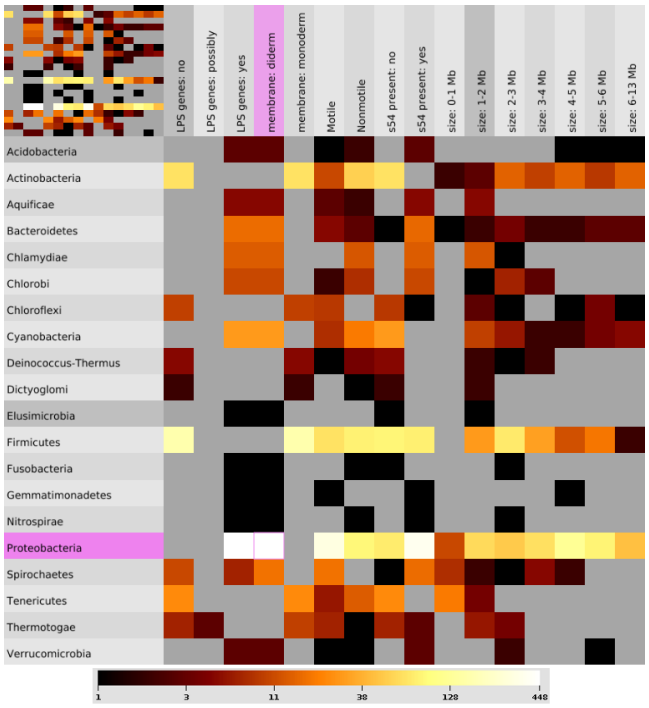


Figure 6: Overview visualization for the comparative genome analysis case study. Rows: species aggregated at the phylum level; Columns: (1–3) composition of the outer membrane; (4,5) cell morphology in terms of number of membranes; (6,7) cell motility; (8,9) presence of Sigma-54 encoding genes; (10–16) genome sizes. Cells are colored according to the number of species associated with them.

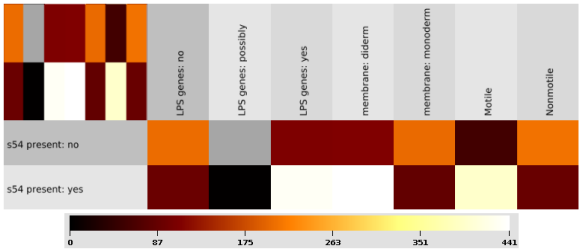


Figure 7: Visualization of the presence of Sigma-54 and relations to composition of the outer membrane (columns 1–3), cell morphology in terms of number of membranes (columns 4,5), and cell motility (columns 6,7). There is a clear correlation between the presence of Sigma-54 and cell motility, diderm cells, and LPS.

6.1 Finding a link between bacterial physiology and a particular regulator

In bacteria, the transcriptional response to stress is often controlled by particular so-called sigma factors. It was long believed that one of these, Sigma-54, which is found in about 60% of the sequenced bacteria, constituted an exception because no common role could be discerned in the plethora of cellular processes it was linked to in different organisms. Recently, a clear link between the protein and the control over the make-up of the bacterial exterior was established [8]. The conclusion was amongst others based on a comparative analysis of the cellular properties of the organisms that have the protein versus the organisms that lack the protein. Most

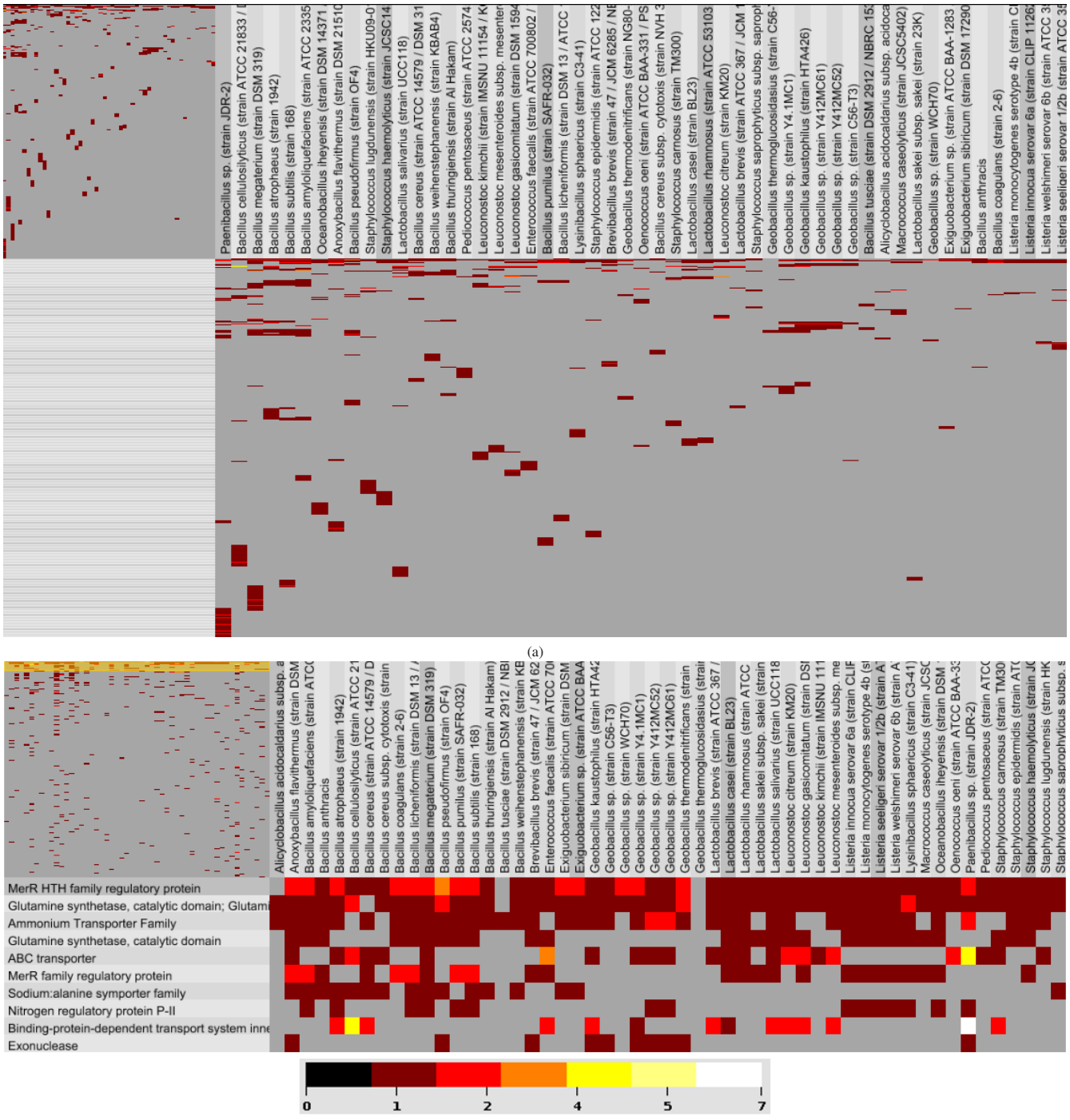


Figure 8: Finding conserved regulatory associations with the GlnR regulon across different species of bacteria. Visualization of 532 Pfam annotations (rows) over genomes (columns). (a) The whole dataset ordered using similarity sorting, (b) the top ten rows of an ordering on the number of genomes related to the row's Pfam annotation.

of the taxonomic, physiological, and morphological data related to the analyzed organisms was derived from the Genomes Online Database [16]. The resulting table consisted of 840 rows related to individual bacterial species/strains and 93 columns containing annotation data. For our purpose, an additional column was added containing information concerning the presence of lipopolysaccharides (LPS) in the outer membrane (data from [27]).

A first visualization of the data set using our tool is given in Fig. 5. Each matrix cell shows the number of species associated to it, encoded in colors. The rows and columns relate to the presence/absence of the regulator and the size of the genome, respectively. The representation shows directly that there is a clear positive correlation between the size of the genome and the presence of the sigma factor. The matrix representation provides the same infor-

mation as given in Fig.2A of [8], where the content of the original spreadsheet had to be converted into a histogram first to provide a similar clarity.

A second visualization (Fig. 6) involved the complete annotation dataset, where the rows represent the organisms aggregated at phylum level. The various phyla represent the whole time range of bacterial evolution. The depicted columns show the types of data that were selected for the analysis, such as the composition of the outer membrane (first three columns), the cell morphology in terms of number of membranes (columns 4 and 5), cell motility (columns 6 and 7), whether Sigma-54 encoding genes are present (columns 8 and 9), and the genome size in discrete bins (last 7 columns). Each cell is colored according to the number of species associated with that cell (see the legend at the bottom of the figure). To show more contrast, a logarithmic scaling is applied. This suppresses the dominance of the phyla Proteobacteria and Firmicutes, which contain many more species than the other phyla. The presented overview provides direct visual clues for relationships that could be explored (this in contrast to the original spreadsheet). There appears to be a correlation between the presence of Sigma-54 and the presence of LPS, and also between Sigma-54 and the number of membranes, with the exception of the phylum Firmicutes. These correlations can be further explored like depicted in Fig. 7 (compare Fig. 1. and 2C. in [8]). The rows now show the presence of Sigma-54, and the columns show LPS, membrane, and motility. There is a clear relation between motility and the presence of Sigma-54: if this protein is present, the vast majority of cells (80%) are motile (in the highlighted cell). A similar correlation with presences of Sigma-54 can be observed for the number of membranes and LPS.

6.2 Determining the commonality in a particular regulatory module

Glutamine is one of the central metabolites and its levels are therefore tightly controlled. In species of the well-studied phylum Firmicutes (which contains many food-related bacteria), the transcription factor GlnR has been related to that control. Recently, the genes associated to GlnR-mediated control were identified for 53 sequenced Firmicutes genomes by the genome-wide identification of putative binding sites [15]. The resulting data table contains function annotation information for 1360 genes as obtained from various sources, including [21] and [30]. In [15] annotation overrepresentation was determined by multiple sorting of the table and manually counting the various annotations. Fig. 8(a) gives an overview of the dataset using COMBat. The annotation data, given as rows (532 in total), related to the functional Pfam domain attributes as provided by [22, 30]. The columns are ordered alphabetically and the rows with the similarity sorting described in Section 4.3. The visualization shows some clustering of annotation terms on specific genomes. However, the most interesting clustering is shown at the top, where some annotation terms appear to occur in almost every genome. The clustering is better specified in Figure 8(b), in which the rows are ordered according to the number of genomes associated with the corresponding annotation term. The most frequently occurring terms are at the top. The figure shows directly that the common regulatory associations of GlnR are with glutamine synthase and ammonium transport as was also reported by [15].

6.3 Identifying a potential role for highly conserved sequence elements

All sequenced genomes contain small (i.e. < 30 nucleotides) sequence elements that occur significantly more often in the genome than expected by chance. The high level of conservation and the high number suggests that each of these elements has a particular biological role. However, in most cases their role has not yet been identified. An example is the nucleotide sequence ACCCGAATTGCT, which is found 117 times in the genome of the sub-

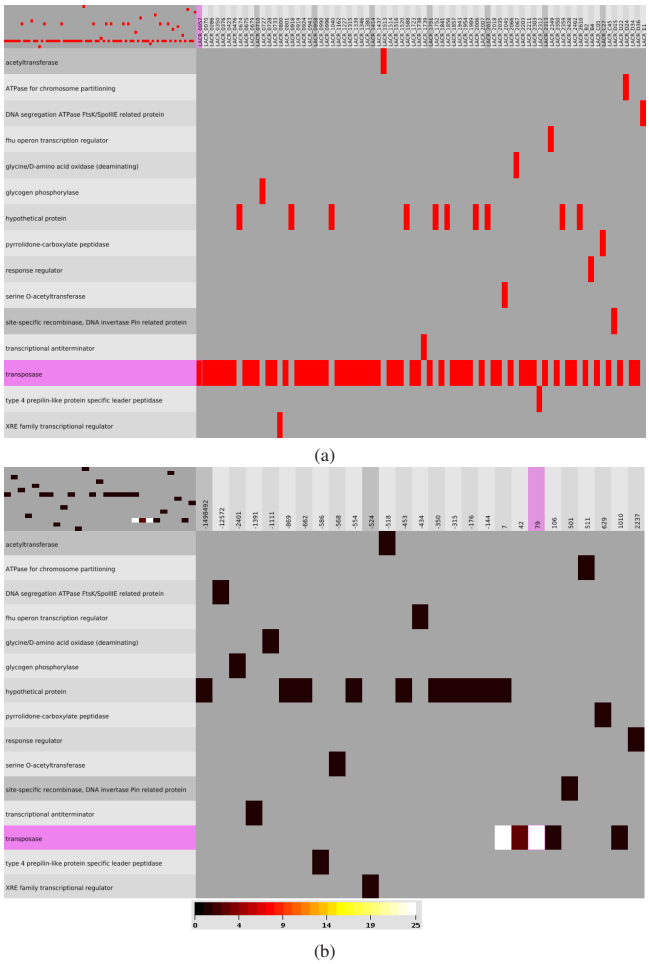


Figure 9: Role of a small highly conserved sequence element in *Lactococcus lactis cremoris SK11*. (a) Co-occurrence of gene products (rows) and locus tags (columns). The highlighted row corresponds to transposase. (b) Relations between gene products (rows) and their distances to the element (columns). These observations suggest a relation between the sequence and the process of transposition, and that it might be part of an insertion sequence.

species *Lactococcus lactis cremoris SK11* but less than 10 times in all other species. A way to infer the potential role of the sequence elements is an analysis of common features in the genetic context. In this case, we found the sequence element associated to 28 annotated locus tags (i.e. genes). To understand the function of this sequence, Figure 9(a) shows the co-occurrences of gene products (rows) and locus tags (columns). The figure shows clearly that the sequence is linked to transposase. Another interesting observation is made in Fig. 9(b), which shows that the element has a constant distance of either 7 or 79 nucleotides to the transposase. Both observations are indicative to a relation between the sequence and the process of transposition and suggest it could be part of the insertion sequence [17].

7 Conclusion

We have proposed the property matrix for visual analysis of relations between annotations in a dataset. The property matrix provides a framework to cross-reference any combination of data types, and assures a uniform visualization style for all the supported data combinations.

We have demonstrated the added value and flexibility of our ap-

proach by presenting three diverse case studies. In the first two, we have reproduced published findings without much effort, while using the original datasets. The relative ease contrasted with the published approach which involved many manual steps and the extensive use of spreadsheets and other tools. In the third, we could easily derive a previously unknown relationship between a particular sequence element and a particular cellular process (i.e. transposase mediated DNA rearrangement). The latter case suggests that our approach can indeed support discovery.

Our approach is currently limited to non-hierarchical annotation data, and it counts co-occurrences to find overrepresented terms. However, some annotation data is hierarchical, for example, the Gene Ontology (GO [1]), in which case simply counting co-occurring terms may not reveal relevant associations. To deal with this, our approach relies on a preprocessing step in which statistically overrepresented terms are pre-computed. Only these terms are used as annotations.

Our prototype implementation COMBat supports the initial requirements. However, the concepts can be further generalized, and additional interaction mechanisms to dynamically group attributes to create new ones are of interest for future work. We plan to include additional seriation algorithms, which may be more relevant to specific biological questions. A challenging problem is scalability: larger data sets containing large amounts of annotation data are a concern, both in terms of memory usage and more fundamental computational performance issues related to matrix sorting. This is also a challenge for visualization, because the number of rows and columns to display quickly exceeds the number of pixels available. To address these issues, we consider both hierarchical approaches that would allow us to aggregate the data at various levels of detail and schemes for guaranteed visibility [19].

REFERENCES

[1] M. Ashburner. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[2] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. University of Wisconsin Press, 1983.

[3] M. Byrne. Reading vertical text: Rotated vs. marquee. *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting*, pages 1633–1635, 2002. Santa Monica, CA: Human Factors and Ergonomics Society.

[4] R. Caspi, T. Altman, J. M. Dale, et al. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic Acids Research*, 38(suppl 1):D473–D479, 2010.

[5] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression data. *Proc. Natl. Acad. Sci. USA*, 95(25):14863–14868, 1998.

[6] C. Faloutsos. Gray codes for partial match and range queries. *IEEE Transactions on Software Engineering*, 14:1381–1393, 1988.

[7] A. Franceschini, D. Szklarczyk, S. Frankild, M. Kuhn, M. Simonovic, A. Roth, J. L. J, P. Minguez, P. Bork, C. von Mering, and L. J. Jensen. STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, 41(Database issue):D808–815, 2013.

[8] C. Francke, T. G. Kormelink, Y. Hagemeijer, L. Overmars, V. Sluijter, R. Moezelaar, and R. J. Siezen. Comparative analyses imply that the enigmatic Sigma factor 54 is a central controller of the bacterial exterior. *BMC Genomics*, 12:385, 2011.

[9] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, STOC '74, pages 47–63, New York, NY, USA, 1974. ACM.

[10] A. J. Gibbs and G. A. McIntyre. The diagram, a method for comparing sequences. its use with amino acid and nucleotide sequences. *Eur. J. Biochem*, 16:111, 1970.

[11] F. Gray. Pulse code communication, 1953. U.S. Patent 2 632 058.

[12] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, April 1950.

[13] N. Henry and J.-D. Fekete. MatrixExplorer: a dual-representation system to explore social networks. *IEEE Trans. Visualization and Computer Graphics*, 12(5):677–684, 2006.

[14] R. C. G. Holland, T. A. Down, M. Pocock, A. Prli, D. Huen, K. James, S. Foisy, A. Drger, A. Yates, M. Heuer, and M. J. Schreiber. Biojava: an open-source framework for bioinformatics. *Bioinformatics*, 24(18):2096–2097, 2008.

[15] T. G. Kormelink, E. Koenders, Y. Hagemeijer, L. Overmars, R. J. Siezen, W. M. de Vos, and C. Francke. Comparative genome analysis of central nitrogen metabolism and its control by GlnR in the class Bacilli. *BMC Genomics*, 13:191, 2012.

[16] K. Liolios, I. M. Chen, K. Mavromatis, N. Tavernarakis, P. Hugenholtz, V. M. Markowitz, and N. C. Kyrpides. The Genomes On Line Database (GOLD) in 2009: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Research*, Database issue(38):D346–354, 2010.

[17] J. Mahillon and M. Chandler. Insertion sequences. *Microbiol. Mol. Biol. Rev.*, 62(3):725–774, 1998.

[18] Y. Makita, M. Nakao, N. Ogasawara, and K. Nakai. DBTBS: database of transcriptional regulation in *Bacillus subtilis* and its contribution to comparative genomics. *Nucleic Acids Research*, 32:D75–77, 2004.

[19] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. TreeJuxtaposer: scalable tree comparison using Focus+Context with guaranteed visibility. *ACM Trans. Graph.*, pages 453–462, 2003.

[20] H. Ogata, S. Goto, K. S. K, W. Fujibuchi, H. Bono, and M. Kanehisa. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, 27(1):29–34, 2000.

[21] L. Overmars, R. Kerkhoven, R. J. Siezen, and C. Francke. MGcV: the microbial genomic context viewer for comparative genome analysis. *BMC Genomics*, 14(1):209, 2013.

[22] M. Punta, P. C. Coggill, R. Y. Eberhardt, J. Mistry, J. Tate, C. Boursnell, N. Pang, K. Forslund, G. Ceric, J. Clements, and et al. The Pfam protein families database. *Nucleic Acids Res*, 40(Database issue):D290–D301, 2012.

[23] R. Rao and S. K. Card. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In *Proc. Conf. Human Factors in Computing Systems (CHI)*, pages 318–322, 1994.

[24] Sanger Institute. GFF (general feature format) specifications document. `http://www.sanger.ac.uk/resources/software/gff/spec.html`, September 2000.

[25] I. Schomburg, A. Chang, S. Placzek, C. Söhngen, M. Rother, M. Lang, C. Munaretto, S. Ulas, M. Stelzer, A. Grote, M. Scheer, and D. Schomburg. BRENDA in 2013: integrated reactions, kinetic data, enzyme function data, improved disease classification: new options and contents in BRENDA. *Nucleic Acids Research*, 41:764–772, 2013.

[26] L. Stein. Generic feature format version 3. `http://www.sequenceontology.org/gff3.shtml`, December 2010.

[27] I. C. Sutcliffe. A phylum level perspective on bacterial cell envelope architecture. *Trends in Microbiology*, 18(10):464–470, 2010.

[28] R. Tatusov, N. Fedorova, J. Jackson, et al. The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, 4(1):41, 2003.

[29] The Potato Genome Sequencing Consortium. Genome sequence and analysis of the tuber crop potato. *Nature*, 475:189–197, 2011.

[30] UniProt Consortium. Update on activities at the Universal Protein Resource (UniProt) in 2013. *Nucleic Acids Research*, 41(Database issue):D43–47, 2013.

[31] L. Wilkinson and M. Friendly. The history of the cluster heat map. *The American Statistician*, 63(2):179–184, 2009.

[32] Wordle – Beautiful Word Clouds. http://www.wordle.net.

[33] H. M. Wu, Y. J. Tien, and C. H. Chen. GAP: a graphical environment for matrix visualization and cluster analysis. *Computational Statistics and Data Analysis*, 54:767–778, 2010.

[34] H. M. Wu, S. Tzeng, and C. H. Chen. Matrix visualization. In C. H. Chen, W. Hardle, and A. Unwin, editors, *Handbook of Computational Statistics (Volume III): Data Visualization*. Springer-Verlag, Heidelberg, 2008.