Biomedical Literature Mining

Mario Fruzangohar

In the fulfilments of the degree of

Doctor of Philosophy

A thesis by prior publications submitted to

Discipline of Genetics

School of Biomedical and Health Sciences

The University of Adelaide

February 2014

Table of Contents

Acknowledgments

I first wish to thank my principal supervisor Prof. David Adelson who is one of the most patient people I have ever met, always welcoming me, even when I had ideas of weird experiments! Thank you David for the enduring support you have given me throughout my candidature. I would also like to thank my co-supervisor Prof. Hong Shen from computer science school. I also truly acknowledge the help and support I have received from Dr. Esmaeil Ebrahimi and also his precious experiences he shared with me.

This research project would not have been possible without the bacterial data providedbymycolleagues at the Research Centre for Infectious Diseases, namely Dr. David Ogunniyi, Dr. Layla Mahdi and Prof. James Paton. I am grateful to all of them for their time and patience.

I must not and cannot forget the significance of the friendships I have made during my candidature here in University of Adelaide. I do not want to miss anyone by naming people individually. I have never overlooked the value of a friendly chat, motivating me through the rest of the day.

Finally and most sincerely, I wish to deeply thank my precious family and friends who gave me the strength and courage to continue my studies by their support and love.

Abstract

Thousands of biomedical articles are published every year containing many newly discovered biological interactions and functions. Manually reading and classifying this information is a difficult and laborious task. Literature mining contains mechanisms and tools to automate the process of extracting biological relationships, storing them in biological databases and finally analyse and present them in a biological meaningful way. In the first stage of literature mining, articles are parsed and get segmented, sentences separated, tokenized and finally annotated by part of speech tags (POS).

POS tagging is the most challenging part because the training corpus is relatively small compared to the large number of biological names therefore limiting the lexicon. There are a number of solutions to address this problem including extending the lexicon manually or using character features of the word. There is no empirical comparison between different solutions. So we developed a complete list of tools including article parser, segmentation, sentence detector, sentence tokeniser, POS tagger and finally noun phrase detector using JAVA and PostgreSQL technologies. We tailored these tools for biomedical texts, and empirically compared them with other tools and we demonstrated increased efficiency of our tools compared to others.

Once biological relationships are extracted they are ready to be stored in databases to be used and shared by others. There a wide range of databases that store annotation data related to genes, proteins and other biological entities. Among them Gene Ontology annotation database is the key database that connects all the other biological entities through a standard vocabulary together. In fact a Gene Ontology (GO) is a controlled vocabulary to annotate proteins based on their molecular function, biological process and cellular components. There are a number of public databases that provide data regarding GO and GO-protein relationships. We collected all relevant data from several public databases and built our specialized updatable GO database on the PostgreSQL platform.

GO classification in a particular sample of genes (up/down regulated) or whole genome of a species can reveal the biological mechanisms related to its activity. Moreover, comparing the GO classification of a species under different biological conditions can elucidate its biological pathways, which can result in the discovery of novel genes to be used in therapies.

We developed a web server using the PHP MVC framework connected to our specialized GO database. In this web server we developed novel visual and statistical methods to perform GO comparisons among multiple samples and genomes.

We also included transcriptome based gene expression levels in GO analysis, resulting in novel meaningful biological reports. This also made comparison of whole genome gene expression across multiple biological conditions possible.

Furthermore, we devised a method to dynamically construct and visualize GO regulatory networks for any gene set sample. Such a network can reveal regulatory relationships between genes helping to explain the correlated expression of genes. The topology of such a network classifies genes based on their connections, and can be used as a new method to detect important genes based on their function as well as their connectivity in the network.

We demonstrated the efficiency of our developed methods in our web server by several case studies using previously published transcriptome data.

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

The author acknowledges that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Mario Fruzangohar *Date*

List of Publications

1. Fruzangohar M, Kroeger TA, Adelson DL (2013) Improved part-of-speech prediction in suffix analysis. PloS one 8: e76042.

2. Fruzangohar M, Ebrahimie E, Ogunniyi AD, Mahdi LK, Paton JC, et al. (2013) Comparative GO: A Web Application for Comparative Gene Ontology and Gene Ontology-Based Gene Selection in Bacteria. PloS one 8: e58759.

3.Fruzangohar M, Ebrahimie E, Adelson DL (2014)Application of Global Transcriptome data in Gene Ontology Classification and Gene Ontology Interaction Networknetwork. Manuscript Prepared

1 Introduction

1.1 Data Mining

Generally, Data mining in biology refers to methods used to extract any meaningful biological relationship from biological raw data using statistical methods. These biological data range from qualitative and quantitative measurements of genes and proteins to biological interactions reported in the literature.

1.2 Biomedical Literature Mining

Thousands of biological articles are published every year in numerous journals. These articles report the results of huge numbers of experiments that report individual biological evidence. This makes the task of searching for a particular biological fact very difficult. Biomedical literature mining refers to the methods and standards to extract, store and present biological relationships. This type of literature miningcan be viewed asa subset of general natural language processing (NLP).

The whole process of literature mining can be divided to three sub-processes: Extract the biological relationships, store them in databases where they are accessible for search, analyse and present the results in meaningful reports. In the following 3 sections we briefly describe each individual process.

1.3 Biological Relationships

A biological relationship in the field of genetics can be any protein-protein[1] or gene-protein interaction involved in any biological pathway. Text mining in the field of genetics, in particular, refers to automating the task of extracting evidence of biological interactions from the literature using statistical methods.

Traditionally, text curators read articles manually and extract concepts by hand. In the past decade as a result of the emergence of high-throughput sequencing technology and subsequent discovery of new transcripts, proteins and biological pathways in different

organisms, the number of biological entities has increased dramatically. This overwhelming number of biological entities has made the task of human curation very difficult and time consuming. So the essence of an efficient text mining tool which is to automate the whole process iscrucially important.

In addition, new tools employing machine learning methods are being developed to predict the function of genes and proteins without performing any lab experiments. These prediction tools have shifted the scale of discovered biological relationships to a much higher level than before.

1.4 Storing Biological Relationships

After extracting biological facts, they are stored in relational databases to construct a biological database. Biological databases play important roles in integrating and sharing common biological relationships between biologists all around the world. One problem in using existing databases is the data redundancy they contain. In other words, one biological fact might exist in multiple databases developed simultaneously by multiple organizations. The opposite problem when using databases is data scattering. This happens when multipletypes of annotations of one biological entity are divided between multiple databases. For example, the polypeptide sequence of one protein is stored in database A, but the molecular function of the same protein is stored in database B. One of the main challenges in using databases is to choose and merge the appropriate available databases to create a new specialized database that is comprehensive and non-redundant. Once constructed, the next challenge is to maintain concurrency of the new database with the original databases. As many public databases are updated daily, we need to automate the process of synchronizing data in our databases with the original public databases.

1.5 Analysis and Presentation of Biological Relationships

During thefinal stage of literature mining, raw data stored in relational databases are aggregated, clustered and statistically compared to identify a putative biological relationship and to present this relationship via a user friendly visual report.

In addition, statistical machine learning models like Artificial Neural networks, Support Vector Machine (SVM), Hidden Markov Model (HMM) and Conditional Random Field (CRF) are trained based on available data and used to predict new biological results.

Artificial Neural networks inspired from human neural networks are excellent non-linear regressors and classifiers which are used to predict a status or class from a number of input signals. Where there is no rule-based method to predict a biological condition from input biological information, artificial neural networks can be trained and employed as efficient prediction tools. For example an artificial neural network can be used as a diagnostic tool to predict cancer status of a breast tumour from visual microscopic features of tumour cells including radius, concavity, symmetry, texture and smoothness of the cells [2].

Support vector machines are high dimensional classifiers and regressors. Where the number of input features of a biological entity is large, an SVM can predict the output class of that entity efficiently. SVMs are the best classifiers for large numbers of articles that must be classified into different subjects. In this application, different keywords (words that are commonly used in one subject) used in one article are designated as input features and the subject of the article is the output class. It is obvious that the number of keywords in one article can exceed several hundred. As another application of SVM, we will show in section 1.6.6 that how an SVM is employed as a biological entity recognition tool.

Hidden Markov Models and Conditional Random Fields are both types of dynamic Bayesian Networks (BN). HMM and CRF are used to label a new sequence of variables. A sequence of variables can be words in a sentence or nucleotides of a DNA fragments. For a sentence, a label is part of speech (POS) tag and for a DNA sequence a label can be any genetic annotation such as a Transcription Factor (TF) binding site [3].

In the following sections we describe each process of literature miningin more detail.

1.6 Extracting Biological Relationships

The text mining process is the first stage of literature mining and is used to detect biological relationships from a sentence or paragraph of text. A biological relationship is represented by a triplet [1,4]. A triplet consists of two biological entity words and one relationship word, which is normally a verb. For example, “Drug X inhibits protein P” or “Protein P participates in molecular function F.”

Text mining methods as other natural language processing methods in any level of processing can be divided into two groups. The first group is Rule Based or Knowledge Based methods. A rule can be any grammatical or any lexical rule that defines a relationship between specific parts of a sentence. These methods are complex and they need extensive field knowledge.

The second group uses machine learning methods [5]. They are based on statistical models that are formulated using training data (explained in section 1.5). The advantage of machine learning methods is their simplicity and ease of implementation compared to rule based methods. However they require training data that in most cases are manually prepared. If training data are already prepared and available, implementing a machine learning method can take less time and effort compared to a rule based model and istherefore preferable.

When an article is processed it undergoes several levels of processing[5,6]. In thefollowing sections we describe each process.

1.6.1 Segmentation of articles

An article is organized based on a format that contains some or all of the following segments; abstract, introduction, methods, results, acknowledgements, etc. Dependingon requirements not all the segments need to be processed. Fortunately, PubMed central the pre-eminent repository for biomedical articles provides them in XML format thatis associated with a DTD (Document Type Definition) file that defines different segments of XML documents. Given the DTD file, we can extract any segment of an article as a node of the XML tree structure.

1.6.2 Sentence Detection

A sentence detector identifies the sentence terminator and separates sentences for further analysis. Apart from ambiguity related to the end of sentence character in general texts, this is more challenging in biomedical texts as many biological names contain characters resembling end of sentence characters. To our knowledge, there is no established method to solve this problem efficiently for biomedical texts.

1.6.3 Sentence Tokenization

The goal of this stage is to separate tokens in a sentence. A token is the smallest part of a sentence that has a relevant Part of Speech Tag (POS). It can be a word or any punctuation character in a sentence. The use of parentheses and punctuation characters makes this task more difficult than it seems. In fact, many biological words contain brackets, quotes and punctuation characters where distinguishing them from real punctuation characters is challenging. Unfortunately, in some published articles an uneven number of brackets and quotes have been used. In other words, there is an opening bracket without matching closing bracket. Such syntax errors can easily cause a tokenizer to fail. An efficient tokenizer should detect asymmetrical use of brackets and quotes with symmetrical checks. A rule based programming technique utilizing regular expressions can accurately tokenize sentences.

1.6.4 Part of speech tagging

Given a sentence, the objective of this stage is to assign a part of speech tag (Noun, Verb, Adjectives and ...) to each word or token in the sentence. Machine learning POS taggers have been shown to be more accurate comparing to rule based methods[7]. Among machine learning methods, Hidden Markov Model [8,9], Conditional Random Field [10]and Maximum Entropy based models[11] are more successful. In the following sections we explain different aspects of an efficient POS tagger:

- Training Corpus

  To use any POS tagger, first we need to train itwith a training corpus. A training corpus is a hand-annotated text that has each word labelled with a POS tag. The accuracy of a POS tagger depends heavily on the type of training corpus that has been used to train it. For example, if a POS tagger is expected to POS tag a biomedical text, then using biomedical training corpus gives better results compared to a financial or a historical training corpus.

- Training POS tagger and Parameter Estimation

  Expectation maximization or modification (EM) recursive algorithm [8] is major algorithms usedfor parameter estimation in a HMM or CRF based POS tagger.It can be proved that by having a large number of observations, the result of EM algorithm converges to Maximum Likelihood Estimation (MLE) values [8]. Therefore, when we have thousands of sentences in our training corpus, then the MLE valuesare best estimations of HMM parameters [9,12].

- Data Sparsity and Smoothing Algorithms

  We can divide ordered sequences of POS tags into two groups, sequences that are seen in the training corpus and those that are unseen. When a POS tagger is trained, the probability of unseen sequences in the second group is zero. This zero probability makes calculations difficult. On the other hand, when a tag sequence is unseen, it does not mean that this tag sequence cannot occur in the text. The common strategy is to smooth sparse probabilities by discounting seen tag sequences and counting unseen tag sequences. There are several methods used for smoothing including additive smoothing, Good-Turing smoothing and linear interpolation smoothing [13,14].

- Handling Unknown Words

  When a POS tagger is trained, it not only learns about the sequence of POS tags but also learns about words appearing in the corpus.Based on these words an internal lexicon of words is built. This lexicon is substantially smaller than the set words a POS tagger will handle after training. This is particularly problematic in the field of biology, where a large number of new biological terms and names emerge every year, making this limitation more obvious.

  The second issue with a limited lexicon is the problem of incomplete data. One word can exist in the lexicon with some POS tags but not all possible POS tags. The common solution to this problem is to extend the lexicon manually as is proposed in[15]. This is impractical considering the volume of new words in biomedical texts. Therefore, an automated method is required to handle unknown words. In fact, the structure of a word such as its suffix and special characters are the most predictive aspects for its POS tag[14]. Using character features of an unknown biological word can potentially help to predict its POS tag. However to our knowledge there has beenno previous work aimed at evaluating the importance of character structure of an unknown word compared to extending the lexicon. Furthermore, there has been no published empirical comparison of suffix and character based POS taggers performance in tagging biomedical texts, particularly for tagging unknown biomedical words.

1.6.5 Phrase Detection

Phrase refers to a group of words in a sentence that function as a unit (noun or verb). The two main phrase types are noun and verb phrases. A noun phrase may contain a reference to a biological entity and a verb phrase can contain a reference to a biological interaction.

The aim of this stage is to extract noun and verb phrases from a sentence. The output of the previous stage is input for the phrase detection stage. The sequence of POS tags that constitutes a phrase can be determined using rule based methods whichare essentially grammar rules. But ambiguity is always present,particularlyat the boundaries of phrases. Employing a Finite State Automaton (FSA) machine with hand-annotated trained data [16]can help efficiently resolve this ambiguity and make a clear distinction between phrases.

1.6.6 Entity and Relationship Recognition

Once a noun phrase is extracted, it can be searched for in biological entity databases or processed by atrained classifier (Support Vector Machine)[17] to determine its biological type (name entity classification) or its exact biological identity (name entity detection)[18].

After recognizing entities in a text, by using grammar rules, syntactic parsing and semantic interpretation [6]three parts of a biological relationship can be extracted [19].

1.7 Storing Biological Relationships

Whether a biological relationship is extracted from a text or is predicted by a data mining tool, it must be stored in order to analyse it and share it with other researchers. In the field of genetics a typical database stores information about different organisms' genomes including their genes and variations (alleles, Polymorphism), gene's products (proteins, RNAs) and also their interactions.

One of the most useful databases contains information about proteins and their functions or biological processes. Gene Ontology (GO) refers to a controlled vocabulary to standardize all the entities in the field of genetics[20,21]. A GO annotation can describe a Molecular Function (like protein binding, recombinase activity and ...) or Biological Process (like

catabolic process, methylation and ...) or Cellular Component (membrane, organelle ...). The gene ontology consortium is responsible for defining and maintaining the GO term database[22]. They not only define GO terms but also many types of relationships between them. On the other hand, some other organisations like Uniprot [23] and the European Bioinformatics Institute (EBI) [24] provide annotation databases to associate proteins with GO terms. These associations are manually extracted from articles or by text mining tools where experimental results have supported this association. Many of these associations are also the results of statistical predictive tools after comparing an unknown protein polypeptide sequence with known functional domains.

Integrating taxonomy, genes, proteins and GO association annotation databases provides a valuable unique database for different GO based analyses. One of the challenges in using such a database is the variability of a gene name class. One gene in a taxonomy database usually has multiple name classes including primary name, synonyms, ORF name and ordered locus name. However, other protein databases refer to a gene by using one of its name classes. This data scattering problem makes the task of finding gene functions more difficult. Integrating a comprehensive gene name database found at the National Centre for Biotechnology Information (NCBI) with other protein databases can improve efficiency of a gene-gene function search engine significantly. To our knowledge few public websites[25]have provided similar integrated databases. Such databases just support a limited number of model organisms with limited gene name classes. They all have limited gene name gene classes or need an extra manual step of gene name conversion. Providing an efficient and fast relational database that can search genes by all available name classes remains a challenging task.

1.8Data Analysis and Biological Reports

1.8.1 Gene Ontology Classification

As we stated in the previous sections, gene ontology-gene association is one important result from literature mining. Gene ontology analysis performed on multiple transcriptome datasets related to a species can explain many biological mechanism and also their involved genes.

The protein enrichment of a particular GO term is estimated based on the number of proteins annotatedwith that GO term. By having a list of genes from a genome we can determine protein enrichment of all the related GO terms, using the GO database. There are a number of GO analysis tools available[26-28], but only a few of them are implemented as web servers [25], so theyrequire manual installation and manual downloading and updating of the GO database. The common use scenario is that a user submits a list of genes from a species and then GO enrichment of this list is compared against the species' genome GO enrichment. Then a Fisher exact test or hyper-geometric distribution comparison is performed [28,29] to determine GO terms that are over represented compared to the entire genome. Genes related to over represented GO terms are usually of particular interest for further functional studies.

1.8.3 Comparative Functional Genomics

Comparing GO enrichment levels among multiple gene samples from multiple treatments can reveal important mechanismsby identifying specific biological pathways. For example once a virus or bacteria infects a host, it usually progressively invades multiple tissues. Comparison of GO enrichment of that pathogen in multiple tissues can reveal specific mechanisms associated with pathogenesis. As another example, we can compare GO enrichment of cancer cells that have undergone different treatments to detect important genes encoding transcription factors. To our knowledge, there is no tool with the ability to study and compare GO enrichment from multiple gene lists, such as from a time course experiment.

Another major advantage of GO analysis is for the development of quality-based gene selection strategies compared to the common approach of gene selection in bacteria which is solely based on the level of gene expression (quantity based gene selection). It should be noted that expression level cannot be proposed as a sole index of gene significance because some genes with lower expression level (such as transcription factors) play a prominent role in bacterial systems biology. An integrative approach, combining quality-based metrics such as GO classification, promoter analysis, and network construction in conjunction with quantity-based gene selection criteria provides a more robust approach for elucidating key bacterial genes and understanding bacterial systems biology. This approach can lead to the discovery of genes associated with specific function(s) for investigation as a novel vaccine or pathway.

1.8.4 GO Internal Relationships

GO terms are linked by hierarchical relationships[21], so one can build a directed acyclic graph (DAG) from these relationships. Visual representation of GO DAG is challenging especially using web based tools. Visual comparisons by means of user friendly graphs and also relevant statistical tests between multiple gene lists can discover new biological mechanisms, especially when this comparison is performed on an arbitrary level of the GO DAG. An efficient visualization tool should provide the ability to navigate across GO DAG nodes smoothly and support statistical tests at any level.

1.8.5 Hypothesis Testing

The selection of appropriate statistical hypothesis test when comparemultiple GO protein enrichment lists is also challenging. In most of biological comparisons with the assumption of normality, parametric test are used. But the assumption of normality in the case of GO protein enrichments for multiple lists is likely to be incorrect. So selection of appropriate data transformation to impose normality or selection of a suitable non-parametric test for this type of comparison is essential. To our knowledge none of the available tools for GO analysis have used either approach.

1.8.6 Expression Level based GO Classification

Functional genomics of bacterial pathogens during disease progression or associated with emerging new highly pathogenic strains is still in its infancy. Bacteria are attractive organisms for GO analysis since they have less post-transcriptional gene silencing compared to animal and plant kingdoms. Therefore gene expression levelsprovide an accurate estimation of protein expression levels[30].

The common approach in transcriptome analysis experiments is that GO analysis is performed on a short list of genes with statistically significant differential expression (up/down regulation). But this means that all significant genes contribute equally in the final GO classification regardless of their actual expression levels.

The major criticism to this approach is that the original level of gene expression can remarkably affect protein production and consequently GO term enrichment. In addition, even genes with low and non-statistically significant expression levels can participate in final GO enrichment through accumulation of small effects.

If we consider expression levels when estimating GO enrichment, we can increase the accuracy of reports and results. By having accurate protein levels of GO terms in a time series of biological samples, one comparison report can determine GO functions that have been consistently up or down regulated as a function of time. Genes related to these GO terms are thus excellent subjects for further investigations.

Furthermore, applying gene expression levels can provide the opportunity to enrich GO terms in a whole genome context (instead of samples with of a short list of genes) and allow us to compare all the genes of a species across multiple biological conditions.

1.8.7 GO Regulatory Network

GO terms are similar to genes in that they interact with each other in a directed acyclic network. Compared to common gene networks, GO networks can provide the key functional genomics based interactions in a broader sense. Classifying a large number of genes in a small number of GO classes and visualising the GO networks significantly decreases the network complexity and, more importantly, offers a new approach for gene selection by considering the genes which contribute to the centre of GO networks.

Despite the availability of GO regulatory relationships in the GeneOntolgy.org database, to our knowledge construction of GO regulatory networks has not yet been dynamically implemented.

Applying expression levels of genes to GO regulatory networks can produce a network representation that explains not only gene/gene function regulatory relationships but also reveals the effect of this regulation on protein production for each GO term in the network.

Construction and visualization of such a network is a major challenge especially via the web. There are a number of network visualization components including Cytoscape [31], Graphiz [32] and JGraph [33]. Cytoscape is optimized to visualize biological connections.

1.8 Biomedical Web Servers

The most important challenge in developing a biomedical web server is related to technical limitations that exist in any hardware platform. A typical web server application contains three abstract layers: the database layer, the application logic layer and the presentation layer. A well developed web server conforms to Model View Controller (MVC) [34] architecture. PHP [35]is a popular and mature web development language. PHP supports object oriented programming (OOP) andservice oriented architecture (SOA) [36].

In the following section we explain the functions and challenges for each layer.

1.8.1 Database Layer

The database layer is the primary place for storage and retrieval of any biological data. Commonly used relational databases can be open source like MY SQL and PostgreSQL[37]or commercial software like Microsoft SQL Server and Oracle. As we know biological datasets are relatively large. A typical biological database can store billions of annotation data related to a species genome. So efficient storage and retrieval of annotation data for a wide range of species in one database is a very difficult task. An efficient database application uses stored database procedures and indexing techniques on all searchable fields to improve performance.

1.8.2 Updating Databases

In order to have the latest data annotation we need to synchronize our dedicated database with public databases. Public databases are growing very fast. For example the volume of protein annotation data supplied by Uniprot.org has increased from 86 Gigabytes to 150 Gigabytes in 9 months. This accounts for millions of newly discovered proteins and their annotations. Processing such a file involves inserting and updating millions of records. So the updating process can take several days. During the update process, the database contains partial data and is not available for public searches. A good updating policy is to use a mirror

database and while one database is serving to the public, the mirror database is updated. Once updating is finished one can exchange the roles of the two databases.

1.8.3 Application Logic Layer

This layer is responsible for performing all the analysis and contains all the algorithmsthat consume web server allocated resources including RAM and CPU. This layer also interacts with the database layer. As the number of users connected to a web server can increase unexpectedly, the resources allocated to users also increase accordingly. Utilizing cache technology can help to reduce the overload of a web server significantly. One good practice is to perform a long and resource intensive job outside of web server space in a separate multithreaded space of the operating system.

1.8.4 Presentation Layer

This layer is responsible for the graphical user interface. The data analysis results are presented as diagrams, graphs and tables here. As biological reports contain dense annotation data, it is nearly impossible to show all of them in one page. An efficient biological report classifies annotation data from the most general to the most detailedlevels. Such a report gives navigational access to all levels of annotated data, so a user has the option of viewing any required detailed information.

As data analysis for large biological samples can be time consuming, an efficient web server provides progress indicators to inform users about the estimated time remaining to finish an analysis. These applications utilize AJAX and JavaScript technologies to implement this functionality.

1.9 Summary and Conclusion

As we explained in previous sections there are a number of non-commercial text mining tools available. However, most of them are not specifically designed to parse and analyse biomedical texts. Therefore, there is a need to develop new biomedical text mining tools and to evaluate and compare their performance with existing tools.

In the past decade, emerging low cost high-throughput sequencing technology has driven a large increase in the number of RNA expression profiles in biological experiments. Interpreting these data to understand the underlying biological mechanisms still remains a challenge. As GO annotation data discovery has been growing rapidly in recent years and GO analysis has gained more popularity in systems biology, the design and construction of a comprehensive gene and protein database associated with GO annotations can provide a valuable resource for further GO analysis.

GO analysis of gene expression profiles is particularly important in order to discover underlying biological pathways and detecting central genes. Therefore, the development of an efficient web server to produce novel and meaningful biological reports based on a comprehensive GO database is an important need of the biological research community.

The major aim of this study was to improve different stages of biological literature mining from beginning to end. To reach this goal we divided our objectives into four different categories:

1- Develop new methods and implement them to improve the following types of existing biomedical text mining tools: POS tagger, phrase detector and biological entity (Gene, Protein) recognizer.
2- Develop a novel biological database to maintain up to date gene, protein and taxonomy information along with GO annotations data.
3- Using the above biological database, devise new methods and tools implemented in an efficient web server to produce novel and meaningful biological reports based on gene expression profiles from biological experiments.
4- Design case studies based on real biological experiments to demonstrate the efficiency of our newly developed methods and compare them with existing tools.

1.10 References

1. Chowdhary R, Zhang J, Liu JS (2009) Bayesian inference of protein–protein interactions from biological literature. Bioinformatics 25: 1536-1542.

2. Chou S-M, Lee T-S, Shao YE, Chen I-F (2004) Mining the breast cancer pattern using artificial neural networks and multivariate adaptive regression splines. Expert Systems with Applications 27: 133-142.

3. He Y, Zhang Y, Zheng G, Wei C (2012) CTF: a CRF-based transcription factor binding sites finding system. BMC genomics 13: S18.
4. Trappey A, Trappey CV, Hsu F-C, Hsiao DW (2009) A fuzzy ontological knowledge document clustering methodology. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 39: 806-814.
5. Cohen KB, Hunter LE (2013) Text Mining for Translational Bioinformatics. PLoS computational biology 9: e1003044.
6. Novichkova S, Egorov S, Daraselia N (2003) MedScan, a natural language processing engine for MEDLINE abstracts. Bioinformatics 19: 1699-1706.
7. Hahn U, Wermter J (2004) Tagging medical documents with high accuracy. PRICAI 2004: Trends in Artificial Intelligence: Springer. pp. 852-861.
8. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77: 257-286.
9. Cutting D, Kupiec J, Pedersen J, Sibun P. A practical part-of-speech tagger; 1992. Association for Computational Linguistics. pp. 133-140.
10. Lafferty J, McCallum A, Pereira FC (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
11. Toutanova K, Klein D, Manning CD, Singer Y. Feature-rich part-of-speech tagging with a cyclic dependency network; 2003. Association for Computational Linguistics. pp. 173-180.
12. Padró M, Padró L (2004) Developing competitive HMM PoS taggers using small training corpora: Springer.
13. Chen SF, Goodman J. An empirical study of smoothing techniques for language modeling; 1996. Association for Computational Linguistics. pp. 310-318.
14. Brants T. TnT: a statistical part-of-speech tagger; 2000. Association for Computational Linguistics. pp. 224-231.
15. Smith LH, Rindflesch TC, Wilbur WJ (2006) The importance of the lexicon in tagging biological text. Natural Language Engineering 12: 335-351.
16. Serrano JI, Araujo L. Evolutionary algorithm for noun phrase detection in natural language processing; 2005. IEEE. pp. 640-647.
17. Takeuchi K, Collier N (2005) Bio-medical entity extraction using support vector machines. Artificial Intelligence in Medicine 33: 125-137.
18. Lee K-J, Hwang Y-S, Kim S, Rim H-C (2004) Biomedical named entity recognition using two-phase model based on SVMs. Journal of Biomedical Informatics 37: 436-447.

19. Dimitris G, Evangelos D (2004) Part-of-speech tagging in molecular biology scientific abstracts using morphological and contextual statistical information. Methods and Applications of Artificial Intelligence: Springer. pp. 371-380.
20. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, et al. (2000) Gene Ontology: tool for the unification of biology. Nature genetics 25: 25-29.
21. Harris M, Clark J, Ireland A, Lomax J, Ashburner M, et al. (2004) The Gene Ontology (GO) database and informatics resource. Nucleic Acids Research 32: D258-261.
22. Chan J, Kishore R, Sternberg P, Van Auken K (2012) The gene ontology: enhancements for 2011. Nucleic Acids Research 40: D559-D564.
23. Dimmer EC, Huntley RP, Alam-Faruque Y, Sawford T, O'Donovan C, et al. (2012) The UniProt-GO annotation database in 2011. Nucleic Acids Research 40: D565-D570.
24. Camon E, Magrane M, Barrell D, Lee V, Dimmer E, et al. (2004) The Gene Ontology annotation (GOA) database: sharing knowledge in Uniprot with Gene Ontology. Nucleic Acids Research 32: D262-D266.
25. Da Wei Huang BTS, Lempicki RA (2008) Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources. Nature protocols 4: 44-57.
26. Zeeberg BR, Feng W, Wang G, Wang MD, Fojo AT, et al. (2003) GoMiner: a resource for biological interpretation of genomic and proteomic data. Genome Biol 4: R28.
27. Al-Shahrour F, Díaz-Uriarte R, Dopazo J (2004) FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. Bioinformatics 20: 578-580.
28. Martin D, Brun C, Remy E, Mouren P, Thieffry D, et al. (2004) GOToolBox: functional analysis of gene datasets based on Gene Ontology. Genome biology 5: R101.
29. Castillo-Davis CI, Hartl DL (2003) GeneMerge—post-genomic analysis, data mining, and hypothesis testing. Bioinformatics 19: 891-892.
30. Cogoni C, Macino G (2000) Post-transcriptional gene silencing across kingdoms. Current opinion in genetics & development 10: 638-643.
31. Saito R, Smoot ME, Ono K, Ruscheinski J, Wang P-L, et al. (2012) A travel guide to Cytoscape plugins. Nature methods 9: 1069-1076.
32. Ellson J, North S (2009) Graphviz-graph visualization software. World Wide Web http://www graphviz org.
33. Bagga J, Heinz A. JGraph—A Java Based System for Drawing Graphs and Running Graph Algorithms; 2002. Springer. pp. 459-460.
34. Leff A, Rayfield JT. Web-application development using the model/view/controller design pattern; 2001. IEEE. pp. 118-127.

35. Gutmans A, Bakken S, Rethans D (2004) PHP 5 Power Programming (Bruce Perens' Open Source Series): Prentice Hall PTR.
36. Josuttis N (2007) SOA in Practice: O'reilly.
37. Douglas K, Douglas SP (2003) PostgreSQL: a comprehensive guide to building, programming, and administering PostgresSQL databases: SAMS publishing.

2 Improved Part-of-Speech Prediction in Suffix Analysis

Mario Fruzangohar[1], Trent A. Kroeger[2], David L. Adelson[1*]

[1]School of Molecular & Biomedical Science, University of Adelaide, SA 5005, Australia

[2]School of Computer Science, University of Adelaide, SA 5005, Australia

[*]david.adelson@adelaide.edu.au

Availability and implementation: Java source code, binaries and setup instructions are freely available at http://genomes.sapac.edu.au/text_mining/pos_tagger.zip

The Supporting Information of this paper is contained in Chapter 6, section 6.1

Statement of Authorship

Title of Paper	Improved Part-of-Speech Prediction in Suffix Analysis
Publication Status	PUBLISHED
Publication Details	PloS one 8: e76042 (2013)

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to thepublication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Principal Author (Candidate)	**MARIO FRUZANGOHAR**	
Contribution to the Paper	Conceived and designed the experiment.Performed the Experiments.Analyzed the data.Contributed reagents/ materials/analysis tool.Wrote the paper	
Signature		Date

Co-Author	**DAVID L. ADELSON**	
Contribution to the Paper	Conceived and designed the experiment.Analyzed the data. Wrote the paper.	
Signature		Date

Co-Author	**TRENT A. KROEGER**	
Contribution to the Paper	Conceived and designed the experiment.Analyzed the data.	
Signature	Deceased	Date

OPEN ACCESS Freely available online

PLOS ONE

Improved Part-of-Speech Prediction in Suffix Analysis

Mario Fruzangohar[1], Trent A. Kroeger[2], David L. Adelson[1]*

1 School of Molecular & Biomedical Science, University of Adelaide, Adelaide, South Australia, Australia, **2** School of Computer Science, University of Adelaide, Adelaide, South Australia, Australia

Abstract

Motivation: Predicting the part of speech (POS) tag of an unknown word in a sentence is a significant challenge. This is particularly difficult in biomedicine, where POS tags serve as an input to training sophisticated literature summarization techniques, such as those based on Hidden Markov Models (HMM). Different approaches have been taken to deal with the POS tagger challenge, but with one exception – the TnT POS tagger - previous publications on POS tagging have omitted details of the suffix analysis used for handling unknown words. The suffix of an English word is a strong predictor of a POS tag for that word. As a pre-requisite for an accurate HMM POS tagger for biomedical publications, we present an efficient suffix prediction method for integration into a POS tagger.

Results: We have implemented a fully functional HMM POS tagger using experimentally optimised suffix based prediction. Our simple suffix analysis method, significantly outperformed the probability interpolation based TnT method. We have also shown how important suffix analysis can be for probability estimation of a known word (in the training corpus) with an unseen POS tag; a common scenario with a small training corpus. We then integrated this simple method in our POS tagger and determined an optimised parameter set for both methods, which can help developers to optimise their current algorithm, based on our results. We also introduce the concept of counting methods in maximum likelihood estimation for the first time and show how counting methods can affect the prediction result. Finally, we describe how machine-learning techniques were applied to identify words, for which prediction of POS tags were always incorrect and propose a method to handle words of this type.

Availability and Implementation: Java source code, binaries and setup instructions are freely available at http://genomes.sapac.edu.au/text_mining/pos_tagger.zip.

Citation: Fruzangohar M, Kroeger TA, Adelson DL (2013) Improved Part-of-Speech Prediction in Suffix Analysis. PLoS ONE 8(10): e76042. doi:10.1371/journal.pone.0076042

Editor: Jérémie Bourdon, Université de Nantes, France

Received June 7, 2012; **Accepted** August 26, 2013; **Published** October 4, 2013

Funding: The authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: david.adelson@adelaide.edu.au

Introduction

Hidden Markov Models (HMM) have been used in Part-Of-Speech (POS) tagging of text for 30 years. HMM and, more recently, Conditional Random Field (CRF) models [1] have been shown to be more accurate compared to other rule based methods such as [2], according to [3].

In the process of tagging articles, one always comes across new words. When training corpora are limited, this problem becomes more acute. Biology in particular, with its proliferation of new words and new gene ontology terms, requires a POS tagger with an efficient method to handle new words. The existence of special characters (capitals, numbers, hyphens or symbols) is the first characteristic used to predict a word tag. If a new word does not contain any special characters, particularly when that word is made of all alphabetic lower case characters, the best method to predict a word tag is to examine the lexical structure of the word, such as the suffix and postfix. In English and some other languages, the suffix is a strong predictive feature for word tagging. In this study we first implemented the TnT POS tagger as a standard machine learning tagger. We then used TnT's suffix analysis method to handle new words. Subsequent testing of TnT system gave an unsatisfactory result for suffix analysis, prompting us to design and implement a novel method, which increased accuracy from 66 to 95 percent.

The problem of handling new words has previously been addressed by manually extending the lexicon by adding new words and all of their possible tags to existing lexicon, as in [4], and while this method seems to be simple and accurate, it requires ongoing effort to identify new biological words and add them to the lexicon. This is particularly problematic in the field of biology, where new chemical, biochemical and genetic terms are emerging in papers every day. So, for this study, we did not consider a lexicon-based method to be appropriate for POS tagging of new biological words. Instead, we focused on improving machine learning techniques for POS tagging, using word lexical features such as special characters and suffixes [5], [6]. We will show how we can achieve better performance by mixing this approach with our proposed machine leaning method.

Hidden Markov Model Theory of POS Tagging

If a sentence of length N, contains words $w_1, w_2 \ldots, w_N$, and POS tags for them are $t_1, t_2 \ldots, t_N$, then according to the topology of the HMM the joint probability of this combination will be:

$$p(t_{1\ldots N},w_{1\ldots N})=\prod_{i=1}^{N}p(t_i|t_{i-1},t_{i-2})p(w_i|t_i) \quad (1)$$

The first term is $p(t_i|t_{i-1},t_{i-2})$, and suggests that each word tag depends on 2 previous tags. This is known as a 3-gram HMM and has been chosen because it has been previously shown that 3-gramsare more accurate than 4-grams [7]. We can estimate this term by counting 3-gram frequencies, and for zero frequency 3-grams we use a previously described efficient smoothing algorithm [8].

The second term is $p(w_i \mid t_i)$, which determines the word probability distribution given a POS tag, and we refer to it from now on as the word conditional probability. This conditional probability shows that the probability of one observation (word) only depends on its current state (tag), not on previous or subsequent states.

To estimate this term, we needed first to process our training corpus. We calculated the frequency with which each word occurs in the corpus and built a *lexicon* database table to store those frequencies. For simplicity we show the *lexicon* database table with its fields defined below:

$$lexicon[word,tag,freq] \quad (2)$$

Subsequently, for each word in our lexicon we determined the maximum likelihood estimation (MLE):

$$p_{mle}(w_i|t_i)=\frac{frequency(w_i,t_i)}{frequency(t_i)} \quad (3)$$

Where the numerator is the number of times word w_i had tag t_i in our ontology database table and the denominator is the number of times that tag t_i was assigned to a word. Both of these were determined using *lexicon* database table.

Of course, our training corpus only contained a limited number of words, whereas our HMM system must be able to deal with text containing many words that do not exist in our *lexicon* database table (unseen words). Thus for unseen words, the P_{mle} will be zero and not applicable for equation 1.

In this situation, the most predictive features of a word's tag in English and some other languages are its suffix and special characters. For example a word ending in '_ing' can have tags VVG, VVJG and VVNG (see Table S1). In this paper we propose a solution to estimate the probability of a word with a particular suffix, having a particular tag. For example, we estimate p(Suffix = ing | Tag = VVG). In other words:

$$p(suffix l_{m-i+1}\ldots l_m|tag) \quad (4)$$

Then we propose a comprehensive character feature analysis and a method to interpolate suffix and character feature probabilities into a single probability to be used in equation (1). Wherever a word does not exist in our lexicon, $P_{mle}(word|tag)$ is zero. We have also used suffix analysis to determine the conditional probabilities of our lexicon words for unseen tags, and we show how efficient suffix information can be used to smooth word probabilities associated with all possible tags, particularly where the conditional probability of a word is based on sparse or unseen POS tags.

In this study, we first explain the previously published TnT method and describe its shortcomings, which led us to propose a simple method for estimating a word's conditional probability. We have evaluated both approaches using real data, and can demonstrate that our method provides a significant improvement in accuracy. We also report optimal parameter settings for both methods.

We have also compared our POS tagger with another state-of-the-art POS tagger that is very well trained based on corpora from different fields including technical terms and these results confirm our POS tagger's efficiency in tagging biological terms such as genes and protein names.

Suffix Prediction Methods

1 TnT/Probability interpolation [7]. The central concept in this method was first used in the original TnT POS-tagger, but some parameters are discussed in [7]. Here we explain these parameters and will show how to set them in our experiments.

If a word of length m ends with a suffix of length i, shown as $l_{m-i+1}\ldots l_m$, then that word also ends in a suffix of length $i-1$, $l_{m-i+2}\ldots l_m$, until the suffix length is 0.We therefore need to *interpolate* probabilities between suffix lengths i and $i-1$, to be able to derive the probability to be used in (4). First we estimated:

$$p(tag|suffix l_{m-i+1}\ldots l_m) \quad (5)$$

To estimate (5), we first determined the frequency of each suffix in our lexicon. In order to do this, we examined all words in *lexicon* database table, and counted the occurrences of each suffix. Finally we made a new database table for suffixes containing suffix, tag and frequency:

$$suffix[suffix,tag,freq_1,freq_n] \quad (6)$$

Here we used two counting methods and stored the results of both (*freq_1* and *freq_n*) in *suffix* database table. *freq_1* is the raw frequency, in that we do not multiply the frequency of the suffix by the frequency of the word itself in *lexicon* database table. For *freq_n*, we multiply the suffix frequency by the word frequency in *lexicon* database table. We know that for each suffix-tag pair:

$$freq\_n(suffix,tag)\geq freq\_1(suffix,tag)$$

The counting method can affect the accuracy of different probability interpolation methods, and we demonstrate this below when applied to real datasets. In fact, our study is the first study to examine the effect of multiplier in counting for MLE estimations of suffix.

It is clear that we can re-state p_{mle} in (5) as:

$$p_{mle}(tag|suffix l_{m-i+1}\ldots l_m)=\frac{frequency(tag,l_{m-i+1}\ldots l_m)}{frequency(l_{m-i+1}\ldots l_m)} \quad (7)$$

p_{mle} can then be estimated using data from *suffix* database table. This probability is a proportion, because, $p_{mle}(tag = VVJ \mid suffix = 'ing')$ is equivalent to the proportion of words with suffix 'ing' that are tagged as VVJ, and we know that:

$$\sum_{j=1}^{K} p(tag = t_j | suffix l_{m-i+1} \ldots l_m) = 1 \quad (8)$$

Where k is the maximum number of POS tags (in Table S1, *k* equals 60).

So for probability interpolation we can state:

$$p_{\text{int}}(tag|l_{m-i+1}\ldots l_m) = \lambda_1 p_{mle}(tag|l_{m-i+1}\ldots l_m) + \lambda_2 p_{\text{int}}(tag|l_{m-i+2}\ldots l_m) \quad (9)$$

Where

$$\lambda_1 + \lambda_2 = 1$$

Equation 9 is recursive and starts from p(tag), meaning that the probability for a suffix with length 1 is interpolated with p(tag).As an initial condition, we assume:

$$p_{\text{int}}(tag) = p_{mle}(tag)$$

We continue interpolating until maximum length 5. [7] has proposed a maximum length of 10, but [4] have argued that well known English suffixes are not more than 4 characters in length and they have proposed using a maximum length of 4. Therefore we concluded that for English, 5 was a reasonable maximum length.

In the TnT/probability interpolation method, the estimation of coefficients λ_1 and λ_2 is based on the standard deviation of $p_{mle}(tag)$ for all tags, which usually yields values between 0.03 and 0.10.

$$\theta = \frac{1}{k-1}\sum_{j=1}^{k}[p_{mle}(t_j) - \overline{p}]^2 \quad (10)$$

Where k is the number of POS tags, and in the example presented in Table S1 is 60.

$$\lambda_1 = \frac{1}{1+\theta}, \lambda_2 = \frac{\theta}{1+\theta} \quad (11)$$

In the TnT method, θ is calculated regardless of context, meaning that coefficients are fixed for all suffixes and tags. In addition, two parameters are not specified. The first parameter is the counting method, the values for which were stored in database table *suffix*, and referred to as *freq_1* and *freq_n*. The second parameter is the interpolation method, and is how suffixes are interpolated in 9. We propose 3 different interpolation methods, which reflect the degree or depth of interpolation. To illustrate these different methods we used a 2 dimensional array to store all of the MLE probabilities needed. For example in the word 'tubulointerstitial', we started from suffix 'l' up to suffix 'itial', we estimated the MLE probability and marked the array cells below if we had an entry for that suffix-tag pair in the '*suffix*' table:

For each column in Table 1 we can interpolate three ways:

1) We interpolate up to the maximum suffix length that has an entry in the *suffix* database table. In this example the suffix is '*tial*' which has a length of 4, so depending on the entries in the Table 1, it could be any value from 1 to 5, in this example the maximum value is in column 'JJ'.
2) We interpolate up to 5 levels regardless of the existence of a corresponding entry in the *suffix* database table.
3) We interpolate until we have an entry for that tag in the *suffix* database table. For example for column tag 'VVI', we interpolate for 2 levels.

After estimating the interpolated probability for each column of Table 1, we used the following Bayesian rule to estimate (4):

$$p(suffix|tag) = \frac{p(suffix)}{p(tag)} p(tag|suffix) \quad (12)$$

For *p(tag)*, we used the sum of the frequencies of all the rows in the *suffix* database table for that tag. For *p(suffix)*, we used the sum of frequencies of all the rows with a suffix of length 1 (the choice of length is arbitrary because it is the relative value for each tag that is important in equation 1). Suffixes of length 1 represent the maximum suffix frequency because suffix counts are cumulative. Therefore, in our example for the word 'tubulointerstitial', to calculate the probability ratio in (12), for a tag 'NN' we would use the following value:

$$\frac{p(suffix)}{p(tag)} = \frac{\sum_{rows-with-suffix=itial} freq}{\sum_{rows-with-tag=NN} freq} \quad (13)$$

This allowed us to calculate the joint probability in (1) using the Viterbi algorithm [9].

2 Maximum Suffix Length (MSL) method. This method differs from the suffix probability interpolation approach because it only requires the probability of the maximum length suffix for each tag. For this, we created a 2-dimensional array based on Table 1 as previous method. For example to estimate p('tubulointerstitial'| tag = JJ), we used Table 1, and for example it was apparent from column JJ that the best option was to use the suffix '*tial*', which we were able to estimate using the *suffix* database table as shown below:

$$p('tubulonterstitial'|tag = JJ) = \frac{\sum_{rows-with-suffix='tial'and-tag=JJ} freq}{\sum_{rows-with-tag=JJ} freq} \quad (14)$$

By estimating (14), we were able to use it directly in (2) by applying the Viterbi algorithm [9]. In the following sections we demonstrate the increased efficiency of MSL over the TnT probability interpolation method.

Materials and Experimental Design

1 Implementing POS tagger. We implemented a trigram HMM and used a linear interpolation method between unigram and bigram [7,10–12] for smoothing. We trained our HMM using

Table 1. suffix versus tags for each suffix in *suffix* database table.

Suffix	POS tags											
	DB	**VVI**	**II**	**CS**	**VM**	**NN**	**RR**	**NNP**	**PND**	**JJ**	**VVB**	**DD**
l	freq(DB,l)	freq(VVI,l)	freq(CS,l)	freq(CS,l)	freq(VM,l)	freq(NN,l)	freq(RR,l)	freq(NNP,l)	freq(PND,l)	freq(JJ, l)	freq(VVB,l)	freq(DD,l)
al		freq(VVI,al)				freq(NN,al)	freq(RR,al)			freq(JJ,al)	freq(VVB,al)	
ial						freq(NN,ial)				freq(JJ,ial)		
tial						freq(NN,tial)				freq(JJ,tial)		
itial												

doi:10.1371/journal.pone.0076042.t001

a tagged corpus available from NCBI [4], using maximum likelihood training for better performance [13].

We built our *lexicon* and *suffix* database tables based on frequencies acquired from the training corpus. Our database tables contained 20,662 *lexicon* word-tags (18,416 distinct words) and 16,004 *suffix* database table suffix-tag pairs. We used the set of POS tags defined in Table S1. We performed our suffix analyses to estimate the smoothed probability of all known words in our lexicon for unseen POS tags (as opposed to using the suffix analyser for unseen words), and updated *lexicon* database table for newly detected tags and their conditional probabilities.

We then analysed all the words in *lexicon* database table, for the occurrence of special characters (punctuation, Greek letters, digits, symbols, etc…) and lower/upper case letters. In order to do this, we assigned a feature string to each word, representing the type and order of characters used in that word. We then estimated the maximum likelihood probability for each feature string and stored that information in *char_feature* database table, as we did for *suffix* database table:

$$char\_feature(feature\_string, tag, freq) \tag{15}$$

To estimate $P_{mle}(word|tag)$, we first looked up each word in *lexicon* database table, for each match we used all the smoothed tag probabilities. If we couldn't find a match, we retrieved the word's character feature conditional probability $p_{char\_feature}$ using *char_feature* database table. This probability was informative if the word contained upper case letters or numbers or other non-letter characters. But if the word was all lower case, it was not very helpful, so for this situation we used the suffix conditional probability p_{suffix} from *suffix* database table. Once this process was complete, we had a value for $P_{mle}(word|tag)$, that might have originated from *lexicon*, *suffix* or *char_feature* database tables, for use in equation (1). Finally, used the Viterbi algorithm [9] to tag the whole sentence.

2 Experiment 1; suffix analysis comparison. In order to compare the MSL method to the probability interpolation method, we designed the following experiment. First, we applied the interpolation method with different values for two parameters: *Counting Method* and *Interpolation Method*, where *Counting Method* values 1 and 2 stand for *freq_1 and freq_n* and *Interpolation Method* has values 1, 2 and 3 as defined in section 2.1. This resulted in 2×3 different parameter combinations.

For the MSL method, only the *Counting Method* parameter changes the tagging result. We used *Counting Method* with values of 1 and 2, resulting in a total of 8 different parameter sets.

To prepare testing data, we downloaded biological articles from the NCBI website [14]. We randomly selected 450 articles from different biological journals, utilising the Viterbi algorithm [9] to tag sentences in these articles.

To determine which words should be tagged based on their suffixes we used the following method. First we checked words in a case independent fashion in the *lexicon* database table. If we found no match, we tested the word for known patterns, such as numbers, number ranges, ordered numbers, etc… We selected lowercase non-matching words that failed to contain known patterns for suffix analysis. We carried out the suffix analysis with our 8 different parameter sets for the two methods. We only recorded POS tag results that were discordant as a function of parameter or method, as we were interested in the relative performance of the two methods.

3 Experiment 2; state-of-art stanford maxent tagger comparison. In the second experiment we compared the overall performance of our POS tagger, with a popular and mature POS tagger. We selected the Maxent POS tagger [5] for 2 reasons: first, this POS tagger has reported a higher accuracy in tagging unknown words and second, this POS tagger has been developed using Java, like our POS tagger. This POS tagger is based on a second order conditioning model and maximum entropy classifiers [6], and uses a cyclic dependency network. This POS tagger comes with different models and we selected the most complete and accurate model called 'english-bidirectional-distsim', which was trained based on Wall Street Journal (WSJ) data, extra English data and technical English data.

MaxentTagger uses The University of Pennsylvania (Penn) Treebank tag-set which consists of 45 POS tags, while our tag-set consists of 60 POS tags (see Table S1). These 2 tag sets have nearly the same POS categories but with different notations and our POS tags are more specific with respect to verbal forms. In order to make a fair comparison, we made a table that maps each tag from one tag-set to its equivalent in another tag-set. We then selected 20 randomly selected articles from NCBI (experiment 1), extracted sentences, and tagged them with both POS taggers.

The first difference we observed was related to the way the two taggers tokenized sentences. Our tokeniser was more accurate in detecting numbers, signs and complex biological names, particularly where a biological name contained special characters like a hyphen, parenthesis, slash, dot or other symbol. In many cases, MaxentTagger tended to split those words, and in some cases combined punctuation characters with the actual word, which led to incorrect results.

We excluded all the tokens that were tokenised differently by the two taggers and only compared the POS tags of similar words. We disregarded words with concordant tags and only logged words with discordant tags (using the mapping table connecting the two tag-sets), as we were interested in the relative performance of the two methods.

Results

1 Experiment 1

After processing 450 biological articles, we tagged a total of 79,791 words based on suffix analysis, of those, 28,895 words with discordant POS tags were identified. We randomly selected a total of 1,500 words in 15×100 word samples, and manually corrected them, (Table S2) with a summary of the results shown in Table 2.

We found that 88.86% of the discordant POS tags were correctly assigned using the maximum length method, compared to7.2% using the interpolation method. Overall (concordant plus discordant), the MSL method was nearly 50% (95.82% vs. 66.39%) more accurate than the interpolation method for suffix prediction. We have shown that the accuracy of suffix based POS tagging can be greater than 95.96% according to line 2 of Table 2.

2 Experiment 2

After processing 20 articles, we found 246 differentially tagged words where MaxentTagger was correct 48% of the time and our tagger was correct 52% of the time. MaxentTagger was better at detecting proper nouns (NNP) like city names, countries and persons, not surprising considering its comprehensive corpus, but our POS tagger was more accurate in tagging biological names. In many cases MaxentTagger incorrectly tagged biological names and symbols as FW (Foreign Word).

Considering the fact that our training corpus was significantly smaller and limited to biological texts, but that it still out performed MaxentTagger, we conclude our POS tagger was more efficient at tagging biological texts.

Discussion

We have shown that the MSL method is much more accurate than the probability interpolation method for POS tagging biological words based on suffix analysis. The MSL method is relatively insensitive to the *Counting Method* parameter, but the *freq_n* multiplier method gave a slightly better result. We also have the optimum parameter selection for the interpolation method where *Counting Method* 1 (multiplier 1) and interpolation method 3, yielded the best result.

In addition to superior accuracy, the MSL method is much faster than the interpolation method, this is because it not only performs fewer calculations in equation (9), but also obviates the need for calculations required by equation 12. Because we stored our *lexicon* and *suffix* tables in a database, the MSL method required significantly less time for database access. Both methods exhibit linear time complexity, so they do not differ in that regard.

Table 3. Words incorrectly tagged with all methods.

Word	Wrong Tag	Correct Tag
breathe	DD	VVI
comply	RR	VVI
kept	II	VVN
obese	DD	JJ
bring	VVG	VVI
kits	PNG	NNS

doi:10.1371/journal.pone.0076042.t003

It should be mentioned that some words were incorrectly POS tagged in all 8 parameter sets. This shows that all machine learning methods failed to POS tag some unknown words. Surprisingly, these words are all common English words and none of them are specifically biological words. Fortunately they account for a very low percentage of all unknown words in biological articles (less than 1% in our experiment). These errors occurred because these common English words were similar to known suffixes in our lexicon. In Table 3, we have listed the problematic words we detected in our dataset.

These unknown common English words are not actually unknown, but they were unknown to our lexicon, that was constructed based on our training corpus, so it makes sense to manually add them along with their POS tags to our lexicon as suggested in [4]. The work required to add new common English words is significantly less than for new biological words, since unknown common English words accounted only one percent of all of the unknown words encountered. For example in the case of irregular verbs (that don't exist in the *lexicon* database table), of which there are about 190, we could simply add them to the *lexicon* database table. However, according to our results, predicting the POS tag of a new biological word is fairly accurate (more than 95.96% based on our results) using our machine learning method.

An alternative and more complicated machine learning method would be to use noun and verb phrases, based on grammar rules. For example, we could try to parse our sentences based on tags, allowing us to detect phrases that violate grammar rules, based on incorrect POS tags. We could then replace incorrect tags with more appropriate ones. This method is non-trivial and needs more research, but one possible approach might bet to use dynamic CRF [15] with tag and phrase information to train the CRF based

Table 2. Statistics in all 15 samples for each parameter set.

Main Method	Counting Method	Interpolation Method	Samples Mean (number of correct tags)	Total correct tags	Overall Accuracy
MSL method	Freq_1	N.A.	88.46	1327	95.82%
MSL method	Freq_n	N.A.	88.86	1333	95.96%
Probability Interpolation Method	Freq_1	Method 1	3.4	51	65%
Probability Interpolation Method	Freq_1	Method 2	3.33	50	64.99%
Probability Interpolation Method	Freq_1	Method 3	7.2	108	66.39%
Probability Interpolation Method	Freq_n	Method 1	2.66	40	64.74%
Probability Interpolation Method	Freq_n	Method 2	2.4	36	64.65%
Probability Interpolation Method	Freq_n	Method 3	5.14	77	65.64%

doi:10.1371/journal.pone.0076042.t002

on 2 features. We expect we could further reduce the number of POS tag errors significantly in this fashion.

Based on comparison of our POS tagger with MaxentTagger, we conclude that our tokenising method tokenised sentences much better than MaxentTagger's tokeniser. Even though MaxentTagger was more accurate tagging common English words and proper nouns, our tagger was better at unknown biological names and gene ontology, due to combined MSL suffix and character feature analysis. Finally, we also showed the importance of suffix probabilities for smoothing the conditional probabilities of unseen POS tags based on known words from our lexicon.

References

1. Lafferty JD, McCallum A, Pereira FCN (2001) Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proceedings of the Eighteenth International Conference on Machine Learning: Morgan Kaufmann Publishers Inc. 282–289.
2. Brill E (1992) A simple rule-based part of speech tagger. Proceedings of the third conference on Applied natural language processing. Trento, Italy: Association for Computational Linguistics. 152–155.
3. Hahn U, Wermter J (2004) Tagging medical documents with high accuracy. PRICAI 2004: Trends in Artificial Intelligence: 852–861.
4. Smith LH, Rindflesch TC, Wilbur WJ (2006) The importance of the lexicon in tagging biological text. Natural Language Engineering 12: 335–351.
5. Toutanova K, Klein D, Manning CD, Singer Y (2003) Feature-rich part-of-speech tagging with a cyclic dependency network. Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. Edmonton, Canada: Association for Computational Linguistics. 173–180.
6. Ratnaparkhi A (1996) A maximum entropy model for part-of-speech tagging. Proceedings of the conference on empirical methods in natural language processing. 133–142.
7. Brants T (2000) TnT: a statistical part-of-speech tagger. Proceedings of the Sixth Conference on Applied Natural Language Processing. Seattle, Washington, USA: Association for Computational Linguistics. 224–231.
8. Chen SF, Goodman J (1999) An empirical study of smoothing techniques for language modeling. Computer Speech & Language 13: 359–393.
9. Forney Jr GD (1973) The viterbi algorithm. Proceedings of the IEEE 61: 268–278.
10. Cutting D, Kupiec J, Pedersen J, Sibun P (1992) A practical part-of-speech tagger. Association for Computational Linguistics. 133–140.
11. Padró M, Padró L (2004) Developing competitive HMM PoS taggers using small training corpora. Advances in Natural Language Processing: 127–136.
12. Dimitris G, Evangelos D (2004) Part-of-speech tagging in molecular biology scientific abstracts using morphological and contextual statistical information. Methods and Applications of Artificial Intelligence: 371–380.
13. Merialdo B (1994) Tagging English text with a probabilistic model. Computational linguistics 20: 155–171.
14. NCBI (2011) PUBMED Journals. NCBI. pp. ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/articles.tar.gz. Accessed Jan. 4 2012.
15. Sutton C, Rohanimanesh K, McCallum A (2004) Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. ACM. 99.

Supporting Information

Table S1 Table of POS tags used in our experiment.
(DOC)

Table S2 Table of 15 manually corrected word samples.
(DOC)

Author Contributions

Conceived and designed the experiments: MF TK DLA. Performed the experiments: MF. Analyzed the data: MF TK DLA. Contributed reagents/materials/analysis tools: MF. Wrote the paper: MF DLA.

3 Comparative GO: A Web Application for Comparative Gene Ontology and Gene Ontology-Based Gene Selection in Bacteria

Mario Fruzangohar[1], Esmaeil Ebrahimie[1,2], Abiodun D. Ogunniyi[2], Layla K. Mahdi[2], James C. Paton[2], David L. Adelson[1]*

[1]Centre for Bioinformatics and Computational Genetics, and [2]Research Centre for Infectious Diseases, School of Molecular and Biomedical Science, The University of Adelaide, South Australia 5005, Australia.

*E-mail: david.adelson@adelaide.edu.au

The Supporting Information of this paper is contained in Chapter 6, section 6.2

Statement of Authorship

Title of Paper	Comparative GO: A Web Application for Comparative Gene Ontology and Gene Ontology-Based Gene Selection in Bacteria
Publication Status	PUBLISHED
Publication Details	PloS one 8: e58759(2013)

Author Contributions By signing the Statement of Authorship, each author certifies that their stated contribution to thepublication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Principal Author (Candidate)	**MARIO FRUZANGOHAR**	
Contribution to the Paper	Conceived and designed the experiments. Performed the experiments. Contributed reagents/materials/analysis tools. Wrote the paper.	
Signature		Date
Co-Author	**ESMAEIL EBRAHIMIE**	
Contribution to the Paper	Conceived and designed the experiments. Performed the experiments. Contributed reagents/materials/analysis tools. Wrote the paper.	
Signature		Date
Co-Author	**ABIODUN D. OGUNNIYI**	
Contribution to the Paper	Conceived and designed the experiments. Performed the experiments. Wrote the paper.	
Signature		Date
Co-Author	**LAYLA K. MAHDI**	
Contribution to the Paper	Conceived and designed the experiments. Performed the experiments.	
Signature		Date
Co-Author	**JAMES C. PATON**	
Contribution to the Paper	Conceived and designed the experiments.	
Signature		Date
Co-Author	**DAVID L. ADELSON**	
Contribution to the Paper	Conceived and designed the experiments. Contributed reagents/materials/analysis tools. Wrote the paper.	
Signature		Date

OPEN ACCESS Freely available online

PLOS | ONE

Comparative GO: A Web Application for Comparative Gene Ontology and Gene Ontology-Based Gene Selection in Bacteria

Mario Fruzangohar[1], Esmaeil Ebrahimie[1,2], Abiodun D. Ogunniyi[2], Layla K. Mahdi[2], James C. Paton[2], David L. Adelson[1]*

1 Centre for Bioinformatics and Computational Genetics, School of Molecular and Biomedical Science, The University of Adelaide, Adelaide, South Australia, Australia, **2** Research Centre for Infectious Diseases, School of Molecular and Biomedical Science, The University of Adelaide, Adelaide, South Australia, Australia

Abstract

The primary means of classifying new functions for genes and proteins relies on Gene Ontology (GO), which defines genes/proteins using a controlled vocabulary in terms of their Molecular Function, Biological Process and Cellular Component. The challenge is to present this information to researchers to compare and discover patterns in multiple datasets using visually comprehensible and user-friendly statistical reports. Importantly, while there are many GO resources available for eukaryotes, there are none suitable for simultaneous, graphical and statistical comparison between multiple datasets. In addition, none of them supports comprehensive resources for bacteria. By using *Streptococcus pneumoniae* as a model, we identified and collected GO resources including genes, proteins, taxonomy and GO relationships from NCBI, UniProt and GO organisations. Then, we designed database tables in PostgreSQL database server and developed a Java application to extract data from source files and loaded into database automatically. We developed a PHP web application based on Model-View-Control architecture, used a specific data structure as well as current and novel algorithms to estimate GO graphs parameters. We designed different navigation and visualization methods on the graphs and integrated these into graphical reports. This tool is particularly significant when comparing GO groups between multiple samples (including those of pathogenic bacteria) from different sources simultaneously. Comparing GO protein distribution among up- or down-regulated genes from different samples can improve understanding of biological pathways, and mechanism(s) of infection. It can also aid in the discovery of genes associated with specific function(s) for investigation as a novel vaccine or therapeutic targets.

Availability: http://turing.ersa.edu.au/BacteriaGO.

Citation: Fruzangohar M, Ebrahimie E, Ogunniyi AD, Mahdi LK, Paton JC, et al. (2013) Comparative GO: A Web Application for Comparative Gene Ontology and Gene Ontology-Based Gene Selection in Bacteria. PLoS ONE 8(3): e58759. doi:10.1371/journal.pone.0058759

Editor: Randen Lee Patterson, UC Davis School of Medicine, United States of America

Received January 7, 2013; **Accepted** February 6, 2013; **Published** March 11, 2013

Funding: This work was supported by the National Health and Medical Research Council of Australia (NHMRC) Project Grant 627142 to JCP and ADO, and NHMRC Program Grant 565526 to JCP. MF is a recipient of School Divisional PhD Scholarship from The University of Adelaide. JCP is a NHMRC Australia Fellow. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: david.adelson@adelaide.edu.au

Introduction

Thousands of papers describing new functions for genes and proteins are published every year, and integrating these results into a useful knowledgebase is an ongoing challenge. The primary means of classifying these results relies on Gene Ontology (GO), which was initially invented to unify the representation of gene and gene product attributes across all eukaryotes using a set of structured, controlled vocabularies [1–3]. The main goal of GO is to develop ontologies to support biologically meaningful annotation of genes and their products in terms of their Molecular Function (MF), Biological Process (BP) and Cellular Component (CC) [1–3]. A list of GO terms can be easily used to build a graph describing the relationship between said terms. Alternatively, text-mining tools using entity recognition methods, combined with manual curation, can be used to extract GO terms associated with a list of genes or proteins. Moreover, the concept of GO network interaction in addition to gene network interaction has recently been developed in model eukaryotes including human, mouse, and Arabidopsis using advanced web applications such as COXPRESdb (http://coxpresdb.jp/) and ATTED-II (http://atted.jp/). This new concept has provided more comprehensive analytical approach in systems biology.

While quality-based gene selection strategies such as GO are established in eukaryotes [4,5], the common approach of gene selection in bacteria is based on level of gene expression (quantity-based gene selection). However, the quantity of expression can not be assumed as a sole index of gene significance as some genes with common lower amount of gene expression (such as transcription factors) play a prominent role in bacterial systems biology. Therefore, the use of quality-based metrics such as promoter architecture, GO classification, and network analysis in conjunction with quantity-based gene selection criteria provides a more robust approach for elucidating key bacterial genes and unraveling bacterial systems biology. The challenge is to present this

information to researchers to compare and discover patterns in multiple datasets using user-friendly visual statistical reports. Furthermore, reliable non-parametric statistical tests need to be integrated into GO web applications in order to compare GO distribution of multiple samples.

To fill these needs, we have designed a web server to compare GO protein distributions from gene expression data, using *Streptococcus pneumoniae* as a model. This organism serves as a paradigm for bacterial pathogens that colonize mucosal surfaces (such as the nose and throat) without causing symptoms, prior to invasion of deeper host tissues, such as the lungs, blood and brain [6,7]. For the first time, we have implemented non-parametric (Kolmogorov–Smirnov [K–S] and Wilcoxon Rank Sum) tests [8–10] to compare GO distribution of multiple samples and Goodness-of-Fit (Chi-square and K–S) tests to compare one sample against its expected reference genome distribution. This application is particularly significant when comparing GO distribution between samples from different sources, such as gene expression patterns *in vitro* vs *in vivo*, or between one anatomic niche and another, for example, gene expression patterns between bacteria harvested from an initial site of infection (such as the nose) and expression patterns during translocation into deeper host tissues, such as lungs, blood, or brain [11,12]. Comparing GO protein distribution among list of up- or down-regulated bacterial genes from different samples can help to understand biological pathways, and mechanism(s) of pathogenesis. It can also help to detect a gene that has been associated with a specific function, and investigate this as a novel vaccine or therapeutic target.

To our knowledge, while there are many GO resources available on the web [13–17], none are suitable for comparison of multiple datasets and gene selection and none contain bacterial data. Our web server is able to rapidly compare large lists of genes/proteins with respect to their GO protein distributions and is regularly updated with the latest gene/protein and GO data.

Materials and Methods

Web Application Architectural Design

In order to obtain a user-friendly and statistically meaningful web application to compare and discover patterns in multiple gene lists, we built a web application based on advanced technological standards. The overall schematic component diagram of the application is shown in Figure 1. In the lower part of Figure 1, there is a process of updating database table. This process ensures that latest protein, gene and GO data exists in the main database system. The main part of system is a web application that is hosted under apache web server. The web application consists of 3 major parts: Model, View and Controller (MVC). Model part contains all database and table query operations, and business logic. It is also responsible to interact with R statistical engine. View part contains all visual components and client side logic including Ajax, JavaScript and HTML. View, with the help of Model, can generate required report to be sent to Controller that interacts with user. Controller part contains all the logic regarding handling user HTTP requests and sending back response to user and also it orchestrates Model and View operations. In other words, it makes instances of objects from View and Model and calls their methods in turn, to send output to user.

Data Collection and Sources

We collected and classified all data needed for the system as:

Gene Ids, gene class name (Primary, Synonym, ordered-locus, ORF) and protein names beside protein accession numbers. Collected from uniprot.org ftp server, we processed manually curated file (uniprot_sprot) and automatically generated file (uniprot_trembl).

Gene ontology Ids and descriptions beside GO relationships (is_a, has_part, part_of, regulates, occurs_in, positively_regulates, negatively_regulates). Collected from geneontology.org ftp server [18].

Protein-GO relationships. Collected from uniprot.org ftp server [19].

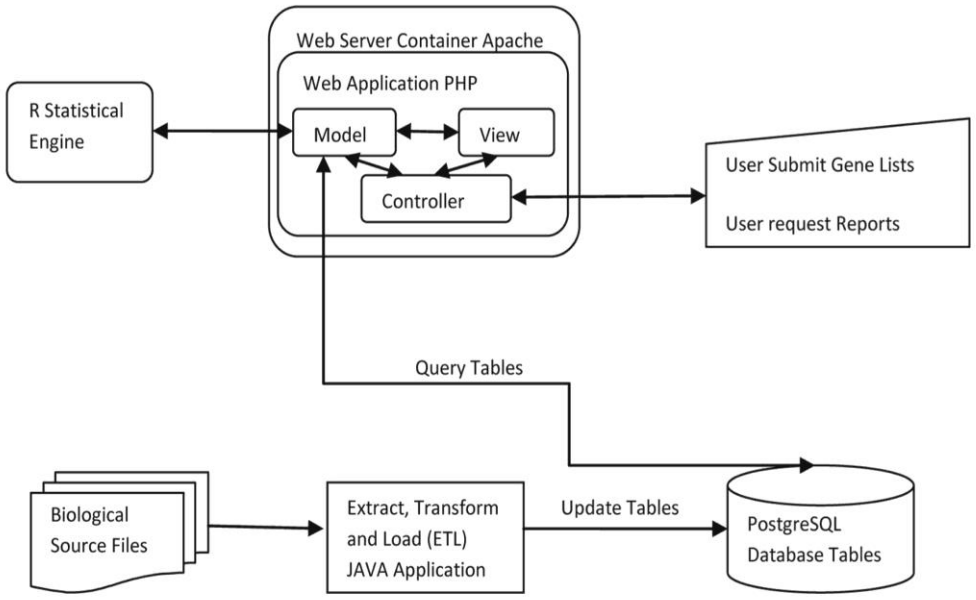

Figure 1. Schematic component diagram of the application. PostgreSQL database is in the centre of system. Lower part of diagram illustrates updating database, and upper part shows how web application uses database.
doi:10.1371/journal.pone.0058759.g001

Taxonomy Ids and descriptions. Collected from ncbi.nlm.nih.gov ftp server.

Data Storage and Update

We stored all the data collected, in a PostGreSQL database, in 6 main tables in normalized form, depicted as an ER (Entity-Relationship) diagram, in Figure 2. For better performance, we have created multiple indexes on all searchable fields. We have used joint table queries as much as possible to improve database performance, and cut down number of queries. We developed application in Java to download flat files from mentioned sources and update tables every 2 weeks automatically.

Data Structures and Processing Logic

In order to prepare useful and user friendly reports, we developed logic and data structures in PHP and integrated into Model part of the web application. The major data structure was directed acyclic graph (or tree, if each node does not have more than one parent node) made from gene ontology and GO relationships. The graph was implemented using linked lists. This data structure is rather large, so we imposed strict PHP memory management to minimize the memory used by the process. Nodes of the graph contain gene ontology and related gene, Protein Ids and other useful information. Root of the graph is one of 3 name spaces MF, BP or CC. Navigation across this graph can be done in multiple ways to produce different reports, which results in proper biological inference. This organization allows for novel visualization of GO graph. In Figure 3, we illustrate how the user can observe nodes of graph and how to navigate through the graph. First, we assign a level to each node of graph starting from root node (level 0), and nodes next to it level 1 and nodes by 2 edges distant level 2, and so forth. The leaves of the graph (nodes that have no children) represent most detail GOs. According to the leveling method, leaves of the graph can be located in multiple levels, not essentially in deepest level (Figure 3B).

In Figure 3A, graph is navigated from root to leaves and vice versa. If current node is in level i, the children nodes of current node (which are located in level i+1), are visualized. Arrows and grey nodes explain logic of navigation. We will explain how informative this navigation method could be in comparing multiple GO graphs in the form of pie charts. In Figure 3B, leaves of the graph are shown as grey nodes. These leaves bear the most detailed GO information. This navigation is supported in all of the visualization reports. In Figure 3C, graph is navigated from root to leaves and vice versa, and at each level, all the nodes in that level is visualized. We will demonstrate how helpful this navigation could be in gene selection mechanism. In this application, all the hypothesis testing and statistical processing is performed using R statistical package. R is externally called by PHP web application. We developed a parameterized R script that can be externally executed and passed by parameters to do statistical analysis.

Genome Wide Comparison and Reference Genome Size Estimation

In order to perform comparison between a gene list and its genome, we used hyper-geometric distribution. This comparison reveals whether a particular GO in a gene list is over-represented or under-represented. For a better user experience, we estimated whole reference genome automatically from database using a novel method, so unlike other web applications user does not need to submit reference genome manually. Other GO web applications prepare 2 by 2 contingency table for each GO group at a time and perform Fisher exact test and report significant GOs based on *P*-value of the test. Instead, we have presented all the common GO groups between sample and its reference genome in a novel bar chart as observed protein numbers next to the expected protein number. Eventually, K–S test is used to compare all GO groups at once between sample and reference genome. Expected protein number of each GO group in sample i, represented by $E(GO_i)$, is mean of hyper-geometric distribution [20]:

$$E(GO_i) = \frac{sample\_size}{genome\_size} \times GO_i$$

To estimate genome size of a taxonomy, we developed a novel method. We first counted number of gene Ids and classified it based on class name (Primary, Synonym, ordered-locus, ORF). We picked the class name with highest number of counts. This number very likely represents actual number of genes in genome. For example we performed this method in *S. pneumoniae*. Estimated genome size 2115 genes pertaining to Ordered-Locus name class, where this number is very close to actual numbers.

Normalization of Protein Numbers of GOs in Multiple Samples

When samples have different number of genes, in order to compare protein numbers of one GO in all samples, we need to adjust protein number based on sample size. We used a simple method. In this method we estimate a coefficient for each sample. Instead of actual protein number we consider product of coefficient of the sample by actual protein number. To estimate coefficients, we order samples based on their size as $S_1 \ldots\ldots S_n$ with the lengths of $l_1 \ldots\ldots l_n$. We assign 1 to coefficient of biggest sample (S_1), then for the rest of samples we have:

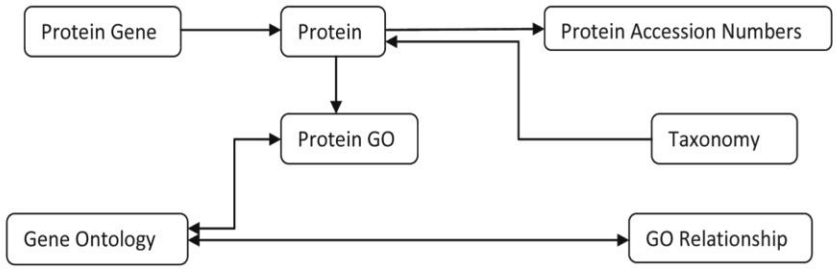

Figure 2. Entity Relationship Diagram. Each entity represents a database table in the system, arrows between entities represent type and multiplicity of relationship between them.
doi:10.1371/journal.pone.0058759.g002

$$coeff(S_1) = 1$$

$$\forall i > 1, coeff(S_i) = \frac{l_1}{l_i}$$

Data Presentation and Visualization

This part of the application implements 'View' part of MVC framework. We have used open source PHP components to produce graphs and charts mainly in Jpg image format. According to our experience, using image-rendered graphs is not only faster than Flash and other plug-ins, but also demands much less memory and CPU on the web browser. Besides, all of plug-ins impose dependency, whereas Jpg images are supported in all browsers and there is no need for manual installation of a plug-in. For a better user experience, we used Ajax as much as possible. Specifically, wherever comparisons are performed among multiple gene lists, data related to each gene list is visualized separately in its own HTML division (div) element, where each division is built and updated separately by one Ajax script. At waiting times, when the application is doing a long running job, an Ajax progress bar component is used.

Results

Unlike other GO tools, our application is specifically designed to generate novel reports to compare multiple gene lists. These reports enable researchers have better understanding of biological pathways, and mechanism(s) of pathogenesis. In addition, it can also help to detect a gene that has been associated with a specific function, and investigate this as a novel vaccine or therapeutic target. To demonstrate the usefulness of this application, we have compared RNA expression of *S. pneumoniae* harvested from the nose lungs, blood, and brain of infected mice. Example data is available on the web application home page to reproduce the reports. We prepared multiple lists of up- and down-regulated pneumococcal genes from various niches and analyzed these lists in the web application using a selection of reports described below.

(A) Pie Chart Comparing Multiple Samples GO Distribution

One of the better methods to visualize change of a specific GO in multiple gene lists is to present percentage of protein distribution among gene lists in pie chart. In the example data provided (Figure 4), we used this novel method to investigate protein distribution involved in "metabolic process" (equivalent to Figure 3A, level 1) between three gene lists from the three comparisons. The results show that the proteins involved in metabolic process constituted 53%, 30%, and 46% of all proteins in the lungs, blood, and brain, respectively. This suggests that pneumococcal genes involved in metabolic process are under-represented in blood during pathogenesis. This report enables user to navigate and observe GO graphs according to Figure 3A. The report also shows related genes in each GO item of the pie chart.

(B) Graph comparing sample versus genome GO distribution

As we mentioned in the Materials and Methods under Genome Wide Comparison and Reference Genome Size Estimation section, comparing gene lists with their expected genome-wide protein distribution can give insight into potential biological significance, especially when this comparison is confirmed by statistical hypothesis testing. Figure 5 shows an example of this capability using bar chart. Here, in the lungs vs nose comparison, under "Molecular Function" ATP binding' molecular function is substantially less than its expected genome distribution. Goodness

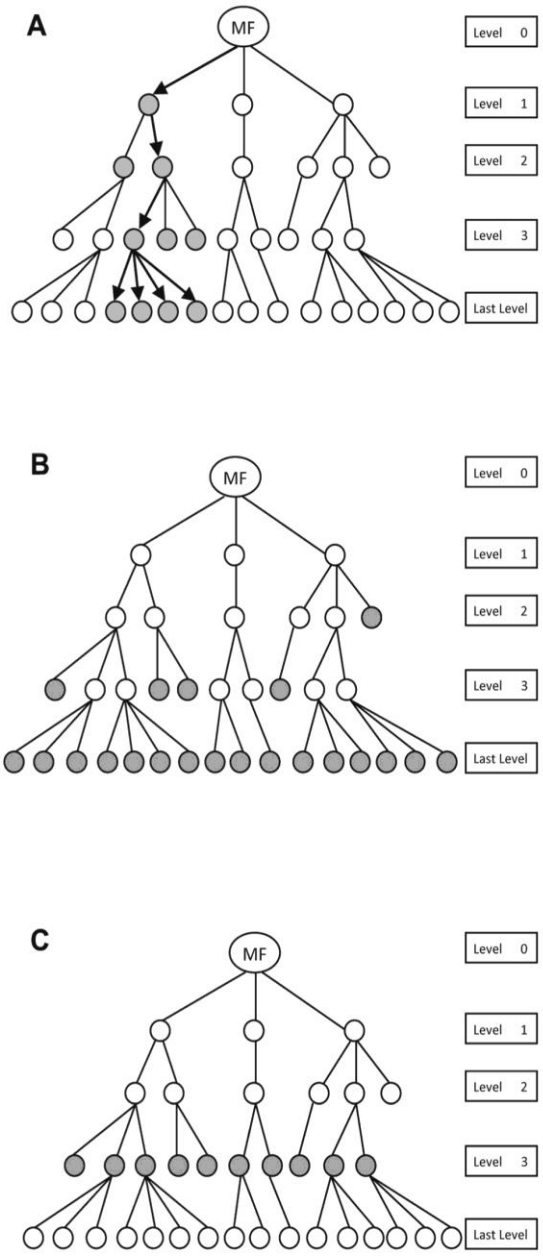

Figure 3. GO Graphs, Navigation and Visualization. A graph structure of molecular function (MF) is built in the memory for one sample. Navigation is done from root (level 0) to the leaves (last level) and vice versa. Visualization can be done in three ways: (A). From each node in a specific level, children of that node are visualized. (B). Leaves of the graph or the most details GO terms are visualized (C). At each level of a GO graph, all the nodes at that level are visualized.
doi:10.1371/journal.pone.0058759.g003

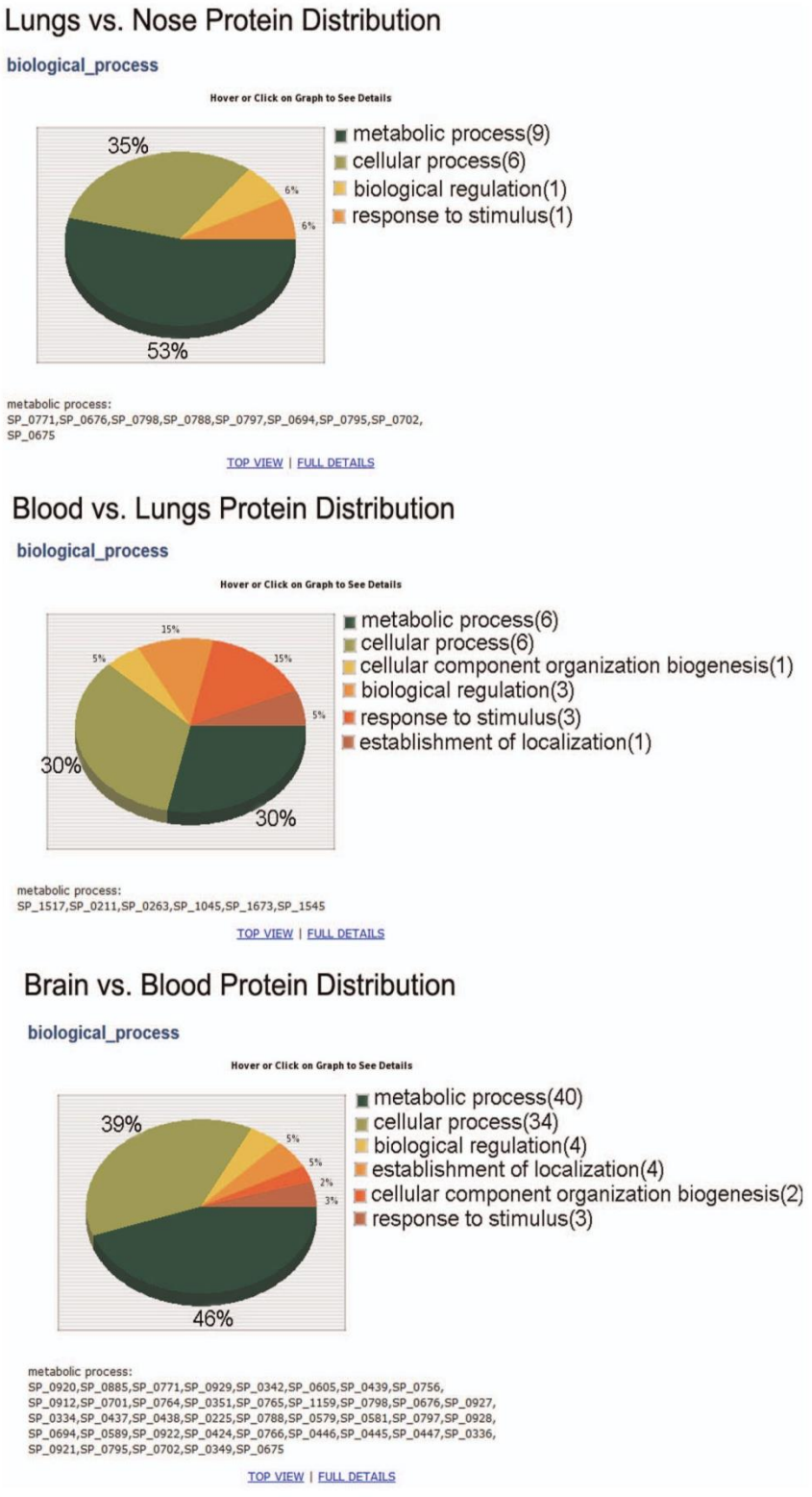

Figure 4. Pie Chart illustrating multiple samples protein distribution. Change of 'metabolic process' protein distribution (percentage) can explain level of bacterial activity in each tissue.
doi:10.1371/journal.pone.0058759.g004

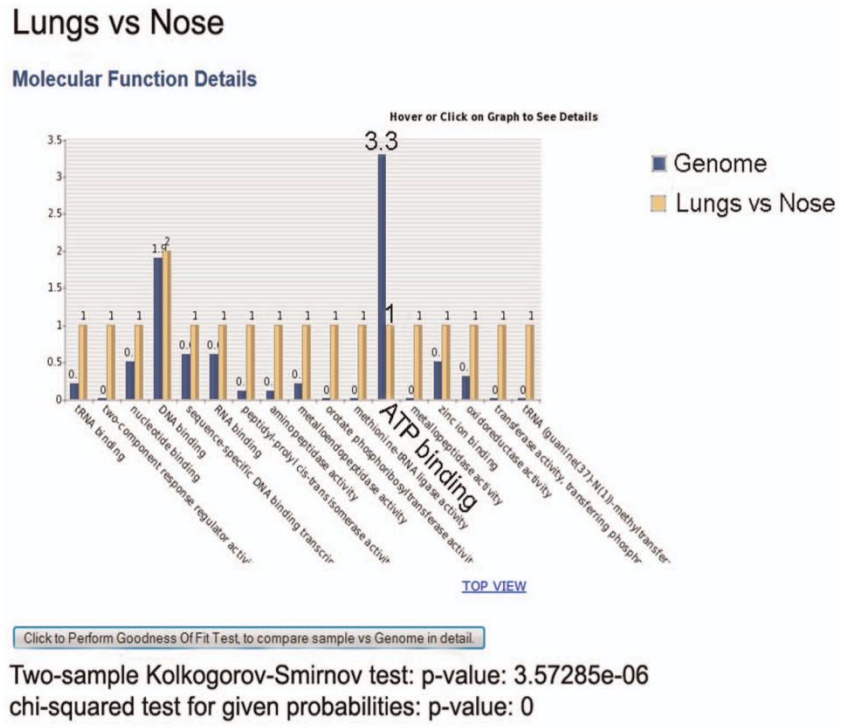

Figure 5. Bar Chart comparing a sample versus its Genome protein distribution. 'ATP binding' protein level of sample is substantially less than its expected number based on whole Genome.
doi:10.1371/journal.pone.0058759.g005

of fit statistical tests (K–S and Chi-square tests) for all GO items at the Molecular Function level, are also reported in the figure. Unlike Pie chart, user can only see number of proteins, instead of percentage of proteins, in each GO.

(C) Tabular gene ontology visualization for gene selection

In our first trial of this system, it was confirmed that this novel report is very effective in selecting important genes. Navigation through the GO graph in this report is based on Figure 3C. In other words, unlike Pie chart user can observe protein distribution of all the GO nodes at a given level of the tree. User can also navigate to leaves of the graph to observe the most detail information as shown in Figure 3B. In this report, each GO item at a given level is shown across selected gene lists (extracted from multiple biological samples) in a line. So, one can observe the rate of change in one GO (by time or location of sample). If there is a significant rate of change in one GO, it is selected for further investigation. Along with rate of change, common genes (intersection) and all the genes (union) are also reported. Common genes can be particularly important, because those genes are over-represented in all gene lists for a given GO. One example of this report is depicted in Figure 6, where lungs vs nose and brain vs blood gene lists were compared. Here, we can observe that 'Sequence-Specific DNA binding transcription factor activity' (arrowed) has been significantly reduced with 0.53 rate and the gene responsible for this is SP_0676. Another example is 'ATP binding' (arrowed) which increased 2.39 times, and the common gene responsible for that is SP_0788.

Discussion

GO analysis provides a new avenue for deeper understanding of gene expression and function, which can be exploited in the context of quality-based gene selection strategy. To achieve this goal, comparative statistically based comparison of GO groups and enriched database are crucial. Current GO web applications are mostly employed in eukaryotic genomes, and lack of reliable comparative statistical analytical approaches hinder the application of GO concept in bacteria. To fill this need, we have designed a user-friendly web application to compare GO protein distributions from gene expression data, using *S. pneumoniae* as a model.

For the first time, we present a dynamic pie chart that illustrates different GO groups as well as the genes involved in each group. This approach allows the user to have a clear, visual comparative understanding of GO distribution in all levels of GO graphs. This can unravel the underlying differential biological pathways, metabolic activation groups, and regulatory networks. Such comparative GO assignments can significantly increase our knowledge of functional genome arrangement and shift during pathogenesis, and provides an avenue for predicting possible activated functional GOs of future virulent strains.

Other GO web resources are able to compare one sample against another reference sample with respect to one GO group at a time, and report the result of enriched GOs based on *P*-values (using Fisher exact test and Chi-square). Instead, our application is able to compare multiple samples visually and statistically using pie chart and solid non-parametric statistical tests (K–S test and Wilcoxon Rank Sum Test) to compare whole samples against each other considering all GO groups. We also managed to facilitate the process of data entry and submitting gene lists. In addition, the Goodness-of-Fit test compares the distribution of GO groups

Ontology	lungs-nose	brain-blood	Rate of Change	Common Genes	All Genes	Unique Genes
tRNA binding	3.76	3	0.8	SP_0788	SP_0788,SP_0579,SP_0581	SP_0579,SP_0581
two-component response regulator activity	3.76	1	0.27	SP_0798	SP_0798	
nucleotide binding	7.52	13	1.73	SP_0675,SP_0788	SP_0675,SP_0788,SP_0919, SP_0756,SP_0581,SP_0437, SP_0438,SP_0579,SP_0885, SP_0912,SP_0349,SP_0439, SP_0445	SP_0919,SP_0756,SP_0581, SP_0437,SP_0438,SP_0579, SP_0885,SP_0912,SP_0349, SP_0439,SP_0445
protein peptidyl-prolyl isomerization	3.76	1	0.27	SP_0771	SP_0771	
DNA binding	7.52	6	0.8	SP_0798,SP_0676	SP_0798,SP_0676,SP_0927, SP_0765,SP_1159,SP_0603	SP_0927,SP_0765,SP_1159, SP_0603
sequence-specific DNA binding transcription factor activity	3.76	2	0.53	SP_0676	SP_0676,SP_0927	SP_0927
RNA binding	7.52	8	1.06	SP_0779,SP_0788	SP_0779,SP_0788,SP_0929, SP_0579,SP_0581,SP_0755, SP_0439,SP_0225	SP_0929,SP_0579,SP_0581, SP_0755,SP_0439,SP_0225
peptidyl-prolyl cis-trans isomerase activity	3.76	1	0.27	SP_0771	SP_0771	
aminopeptidase activity	3.76	1	0.27	SP_0797	SP_0797	
metalloendopeptidase activity	3.76	1	0.27	SP_0694	SP_0694	
orotate phosphoribosyltransferase activity	3.76	1	0.27	SP_0702	SP_0702	
methionine-tRNA ligase activity	3.76	1	0.27	SP_0788	SP_0788	
ATP binding	3.76	9	2.39	SP_0788	SP_0788,SP_0756,SP_0581, SP_0437,SP_0438,SP_0579, SP_0885,SP_0912,SP_0349	SP_0756,SP_0581,SP_0437, SP_0438,SP_0579,SP_0885, SP_0912,SP_0349
transcription, DNA-						

Figure 6. Tabular Report Used in Selecting Important Genes and Important GO. Genes located under 'common genes' column are the most important genes in each GO item. On the other hand, GO item with higher 'Rate of Change' represents more significant biological trend.
doi:10.1371/journal.pone.0058759.g006

between any given sample versus the reference genome. This test provides another piece of information for finding over- or under-represented GO groups, relative to the entire genome. Such discovered GO groups offer a route map for prevention and/or treatment of bacterial pathogenesis and virulence through inactivation of the GO group.

In this work, we also present tabular GO visualization for gene selection. This simple approach offers the advantage of finding genes that are common to samples from different sources (for example, genes that are central to a pathogenic process). Such genes would serve as targets for controlling the movement of pathogens from one tissue to another. The tabular data also presents the rate of change in number of genes/proteins between samples, which has significant implication in deciding which functional GO group is more enriched between given samples. For example, functional groups involved in two-component sensor activity, DNA binding, and antioxidant activity, are central to *S. pneumoniae* functional genomics. These groups are excellent targets for monitoring bacterial evolution and pathogenesis and provide valuable clues for predicting the possible activated GOs of emerging virulent strains.

In conclusion, we present a novel, user-friendly web application that compares GO protein distributions across samples from different sources. This tool is particularly useful in understanding biological pathways, mechanism(s) of infection and discovery of genes associated with specific function(s) for investigation as a novel vaccine or therapeutic targets. The application can also be expanded to integrate gene expression levels (quantity-based) with the current quality-based GO approach, which would result in more accurate selection of important genes from diverse biological sources. It also has the potential to include eukaryotic information to study diseases such as human cancer and other biological phenomena.

Author Contributions

Conceived and designed the experiments: MF EE ADO LKM JCP DLA. Performed the experiments: MF EE ADO LKM. Analyzed the data: MF EE ADO LKM JCP DLA. Contributed reagents/materials/analysis tools: MF EE DLA. Wrote the paper: MF EE ADO DLA.

References

1. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, et al. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat Genet 25: 25–29.
2. Consortium TGO (2008) The Gene Ontology project in 2008. Nucleic Acids Res 36: D440–444.
3. Consortium GO (2012) The Gene Ontology: enhancements for 2011. Nucleic Acids Res 40: D559–564.
4. Huang da W, Sherman BT, Tan Q, Collins JR, Alvord WG, et al. (2007) The DAVID Gene Functional Classification Tool: a novel biological module-centric algorithm to functionally analyze large gene lists. Genome Biol 8: R183.
5. Huang da W, Sherman BT, Lempicki RA (2008) Systematic and Integrative Analysis of Large Gene Lists Using DAVID Bioinformatics Resources. Nat Protoc 4: 44–57.
6. Bogaert D, De Groot R, Hermans PW (2004) *Streptococcus pneumoniae* colonisation: the key to pneumococcal disease. Lancet Infect Dis 4: 144–154.

7. O'Brien KL, Wolfson LJ, Watt JP, Henkle E, Deloria-Knoll M, et al. (2009) Burden of disease caused by *Streptococcus pneumoniae* in children younger than 5 years: global estimates. Lancet 374: 893–902.
8. Stephens MA (1974) Edf Statistics for Goodness of Fit and Some Comparisons. J Am Stat Assoc 69: 730–737.
9. Wilcoxon F (1945) Individual Comparisons by Ranking Methods. Biometrics Bull 1: 80–83.
10. Mann HB, Whitney DR (1947) On a Test of Whether One of 2 Random Variables Is Stochastically Larger Than the Other. Ann Math Stat 18: 50–60.
11. Mahdi LK, Wang H, Van der Hoek MB, Paton JC, Ogunniyi AD (2012) Identification of a novel pneumococcal vaccine antigen preferentially expressed during meningitis in mice. J Clin Invest 122: 2208–2220.
12. Ogunniyi AD, Mahdi LK, Trappetti C, Verhoeven N, Mermans D, et al. (2012) Identification of genes that contribute to the pathogenesis of invasive pneumococcal disease by *in vivo* transcriptomic analysis. Infect Immun 80: 3268–3278.
13. Al-Shahrour F, Diaz-Uriarte R, Dopazo J (2004) FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. Bioinformatics 20: 578–580.
14. Boyle EI, Weng SA, Gollub J, Jin H, Botstein D, et al. (2004) GO: TermFinder – open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. Bioinformatics 20: 3710–3715.
15. Martin D, Brun C, Remy E, Mouren P, Thieffry D, et al. (2004) GOToolBox: functional analysis of gene datasets based on Gene Ontology. Genome Biol 5: R101.
16. Beissbarth T, Speed TP (2004) GOstat: find statistically overrepresented Gene Ontologies within a group of genes. Bioinformatics 20: 1464–1465.
17. Dennis G, Sherman BT, Hosack DA, Yang J, Gao W, et al. (2003) DAVID: Database for annotation, visualization, and integrated discovery. Genome Biol 4: R60.
18. Harris MA, Clark J, Ireland A, Lomax J, Ashburner M, et al. (2004) The Gene Ontology (GO) database and informatics resource. Nucleic Acids Res 32: D258–D261.
19. Camon E, Magrane M, Barrell D, V L, Dimmer E, et al. (2004) The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. Nucleic Acids Res 32: D262–D266.
20. Castillo-Davis CI, Hartl DL (2003) GeneMerge – post-genomic analysis, data mining, and hypothesis testing. Bioinformatics 19: 891–892.

4 Application of Global Transcriptome Data in Gene Ontology Classification and Construction Of A Gene Ontology Interaction Network

Mario Fruzangohar[1], EsmaeilEbrahimie[1], David L. Adelson[1]*

[1]School of Molecular and Biomedical Science, The University of Adelaide, South Australia 5005, Australia.

mario.fruzangohar@adelaide.edu.au

esmaeil.ebrahimie@adelaide.edu.au

*david.adelson@adelaide.edu.au

The Supporting Information of this paper is contained in Chapter 6, section 6.3

Statement of Authorship

Title of Paper	Application of Global Transcriptome data in Gene Ontology Classification and Gene Ontology Interaction Network
Publication Status	Manuscript Prepared
Publication Details	N.A.

Author Contributions

By signing the Statement of Authorship, each author certifies that their stated contribution to thepublication is accurate and that permission is granted for the publication to be included in the candidate's thesis.

Principal Author (Candidate)	**MARIO FRUZANGOHAR**	
Contribution to the Paper	Conceived and designed the methodology.Performed the Experiments.Analyzed the data.Contributed reagents/ materials/analysis tool.Wrote the paper	
Signature		Date

Co-Author	**EsmaeilEbrahimie**	
Contribution to the Paper	Conceived and designed the methodology.Performed the Experiments.Analyzed the data.Contributed reagents/ materials/analysis tool.Wrote the paper	
Signature		Date

Co-Author	**DAVID L. ADELSON**	
Contribution to the Paper	Conceived and designed the methodology. Analyzed the data.Contributed reagents/materials/analysis tool.Wrote the paper	
Signature		Date

Abstract

Background

Gene Ontology (GO) classification of statistically significant over/under expressed genes is a common method for interpreting transcriptomics data as a first step in functional genomic analysis. In this approach, all significant genes contribute equally to the final GO classification regardless of their actual expression levels. However, the original level of gene expression can significantly affect protein production and consequently GO termenrichment. Furthermore, even genes with low expression levels can participate in the final GO enrichment through cumulative effects.

GO terms have regulatory relationships allowing the construction of a regulatory directed network combined with gene expression levels to study biological mechanisms and select important genes for functional studies.

Results

In this report, we have used gene expression levels in bacteria to determine GO term enrichments. This approach provided the opportunity to enrich GO terms in across the entire transcriptome (instead of a subset of differentially expressed genes) and enabled us to compare transcriptomes across multiple biological conditions. As a case study for whole transcriptome GO analysis, we have shown that during the infection course of different host tissues by *streptococcus pneumonia*, Biological Process and Molecular Functions' GO term protein enrichment proportions changed significantly as opposed tothose for Cellular Components. In the second case study, we compared *Salmonella enteritidis* transcriptomes between low and high pathogenic strains and showed that GO protein enrichment proportions remained unchanged in contrast to a previous case study.

In the second part of this study we show for the first time a dynamically developed enriched interaction network between Biological Process GO terms for any gene samples. This type of network presents regulatory relationships between GO terms and their genes. Furthermore, the network topology highlights the centrally located genes in thenetwork which can be used for network based gene selection. As a case study, GO regulatory networks of *streptococcus pneumonia* and *Salmonella enteritidis* were constructed and studied.

Conclusion

In both *Streptococcus pneumonia* and *Salmonella enteritidis*, the pathways related to GO terms "Environmental Information Processing", "Signal transduction" and "two-component

system”were associated with increasing pathogenicity, breaching host barriers and the generation of new strains.

This study demonstrates a comprehensive GO enrichment based on whole transcriptome data, along with a novel method for developing a GO regulatory network showing overview of central and marginal GOs that can contribute to efficient gene selection.

Background

The functional genomic changes in bacterial pathogens during disease progression or in emerging highly pathogenic strains are poorly understood. Classifying genes into distinct functional groups through Gene Ontology (GO) is a commonly used and powerful tool for understanding the functional genomics and underlying molecular pathways. However, GO protein enrichment is related to the amount and number of proteins described in that GO, and in eukaryotes mRNA levels are often poorly correlated with protein expression. Bacteria are attractive organisms for GO analysis since they have less Post-transcriptional gene silencing compared to animals and plants [1] with gene expression levels moderately correlated with protein levels [2].

Because of the lack of specific resources for GO analysis in bacteria, we recently developed Comparative GO, a PHP based web application for statistical comparative GO and GO-based gene selection in bacteria [3]. Comparative GO has the potential to provide a comprehensive view of bacterial functional genomics by categorizing genes into a limited number of annotated GO groups [3, 4].

Another major advantage of GO analysis is developing quality-based gene selection strategies compared to the common approach of gene selection in bacteria which is solely based on the level of gene expression (quantity based gene selection) [3, 4]. It should be noted that expression level alone cannot be used as a sole index of gene significance because some genes with lower expression levels (such as transcription factors) play a prominent role in bacterial systems biology [3, 4]. An integrative approach, combining quality-based metrics such as GO classification, promoter analysis, and network construction in conjunction with quantity-based gene selection criteria provides a more robust approach for identifying key bacterial genes and describing bacterial systems biology. Such an approach can contribute to the discovery of genes associated with specific function(s) for investigation as novel vaccine candidates or pathways for pharmacological targeting.

Biological process GO terms are analogous to genes because they have regulatory relationships with each other that can be used to construct a directed acyclic network. Compared to common gene networks, GO regulatory networks can identify key functional genomics based interactions in a broader sense. Classifying a large number of genes in a small number of GO classes and visualising the GO networks can significantly decrease the

network complexity and, more importantly, offers a new approach for gene selection by considering the genes which contribute to central nodes in GO networks. To our knowledge there is no tool and methodology currently available to dynamically construct GO regulatory networks.

The common approach in transcriptome experiments is that GO analysis is carried out on a short list of genes with statistically significant differential expression (up/down regulated) [5-7]. In this approach, all selected genes contribute equally in the final GO classification regardless of their actual expression levels.

The major drawback to this approach is that the original levels of gene expression can significantly affect protein production and consequently actual GO term enrichment. In addition, even genes with low or statistically non-significant expression levels can participate in final GO enrichment through cumulative effects.

In this report we show for the first time how gene expression levels in bacteria can be used to determine GO term enrichments. By using gene expression levels as coefficients, we also took into account the impact of non-significantly expressed genes in GO enrichment. This approach provided the opportunity to enrich GO terms in the entire transcriptome genome (instead of samples of a short list of genes) and enabled us to compare GO terms of transcriptomes across multiple biological conditions. In order to achieve this, we enhanced our recently developed web server, Comparative GO [3, 8]. To enable analysis of very large gene sets such as from a whole genome, we implemented cache technology to improve web server performance. We also integrated robust non-parametric chi-square based tests into our web application to test if there is a significant difference between genome scale GO enrichment levels of 2 biological conditions. The ability to bin a sample's GO enrichment levels makes the 2-sample chi-square test a suitable test to compare such data sets, particularly where background data distribution is unknown [9].

We applied our new methods to two important bacterial pathogens, *streptococcus pneumonia* and *Salmonella enteritidis* in order to unravel the global, transcriptome based, GO pattern of *streptococcus pneumonia* during infection of host tissues and breaching of tissue barriers as well as the comparison of low and highly pathogenic *Salmonella enteritidis* strains [10].

In the second part of this study we describe the implementation of GO based gene selection and GO network discovery. We show for the first time a dynamically constructed interaction network between Biological Process GO terms for any given bacterial gene sample. To this

end, GO relationships were extracted from Gene Ontology database [11-13], and used to build a directed acyclic graph (DAG). To visualise the final DAG, we used the Cytoscape web browser plug-in [14]. We used our *streptococcus pneumonia* and *Salmonella enteritidis* data sets as case studies for this method.

Material and Methods

Incorporation of gene expression levels into GO analysis

Normalization of Expression Levels

The system accepts any type of expression level such as microarray fold-change data and RPKM counts of RNA-Seq data. In all cases, for each gene, one normalized coefficient is estimated based on its expression level within the sample or within the genome. If we want to perform comparative GO analysis on a sample of *n* genes, and the expression level of gene *i* in sample *j* is e_{ij} and also given that the smallest expression level across n samples is denoted by e_{min}, then the coefficient of gene *i* in sample *j* (C_{ij}) is estimated as :

$$C_{ij} = e_{ij} \Big/ e_{\min} \text{ where } e_{\min} > 0$$

If a trait of interest is measured for each sample, then C_{ij} can be replaced by the correlation of gene *i* with a trait as suggested in [15]. But in our study we have not measured any phenotypic traits, so we use normalized expression levels as coefficients.

GO Enrichment Methodology and Significant Gene Set Detection

Furthermore, if GO term t in sample s is associated with genes $G_{1s}...G_{ms}$, then the protein enrichment level of GO term t in sample s (PE_{ts}) is estimated as:

$$PE_{t,s} = \sum_{i=1}^{m} C_{is} \qquad (1)$$

As we know each GO term is associated with multiple genes (or proteins). And a set of genes that are part of the same biological pathway, are related to a common GO term. Therefore, the problem of finding the most significant gene set across multiple samples (biological conditions) is reduced to finding the most significant GO terms.

To detect the most important GO term we define and estimate a metric for each GO term. The GO term associated with the maximum value for this metric is the most important GO term and genes associated with it are the desired gene set. We formulate the process as below.

Suppose we have 2 expression profiles of all genes ($G_{1...}G_m$) from 2 samples s1 and s2.

Then the most significant GO term is the term that maximises or minimizes equation 2.

$$PE_{t,s1} \Big/ PE_{t,s2} \qquad (2)$$

that is the equivalent of maximizing:

$$|\mathrm{Log}(PE_{t,s1}) - \mathrm{Log}(PE_{t,s1})| \qquad (3)$$

where PE is estimated by equation 1.

In the case where we have more than 2 samples (n>2), we use the geometric average across all samples as the metric. So we select GO term t that maximizes:

$$\sqrt[n-1]{\prod_{i=1}^{n-1} PE_{t,si+1} \Big/ PE_{t,si}} \qquad (4)$$

These methods have been implemented on our web server [8]. Particularly, in the tabular report, a user can compare enrichments of GOs for any number of samples and detect highly variable GO terms. These comparisons can be made at any level of a hierarchical GO tree. Comparison of GO enrichments at higher levels of the GO tree is particularly important such as when we move from leaves of the GO tree (detailed GO terms) to the higher levels (more general GO terms), subtle variations can be accumulated and significant changes can be observed in GO terms located at higher levels.

Hypothesis Testing Tool

We implemented a tool to test the hypothesis of a significant difference between 2 genomes/samples GO term distributions. Specifically, we implemented a Chi-Square test for 2 samples and we compared it with the Kolmogorov–Smirnov test using the R-statistical package [16]. These two methods are both non-parametric and are suitable for comparing 2 lists of paired numbers like GO term enrichment values for 2 samples.

GO regulatory Network Construction

Regulatory relationships (up/down regulation) were extracted between Biological Process terms from the Gene Ontology database [11, 12]. We stored these relationships in our internal database [3]. For any given gene sample, our application builds a GO DAG (Directed Acyclic Graph) network, based on regulatory relationships.

In order to infer new relationships from available relationships we expanded initial GO network to include parental nodes; then, new relationships were inferred from relationships between parental GO nodes to the nodes in the network. Figure 1 depicts a simple GO regulatory network, where grey nodes represent the GO terms related to the sample, and the relationships between GO terms are depicted by green arrows. As we can see at the top of the graph, there is a relationship between parental GO terms 2 and 3. Accordingly, we inferred 3 new relationships between nodes 4, 5, 6 and node 7, depicted as green dotted arrows. The final enriched network can describe novel regulatory relationships between GO terms and consequently between their associated genes.

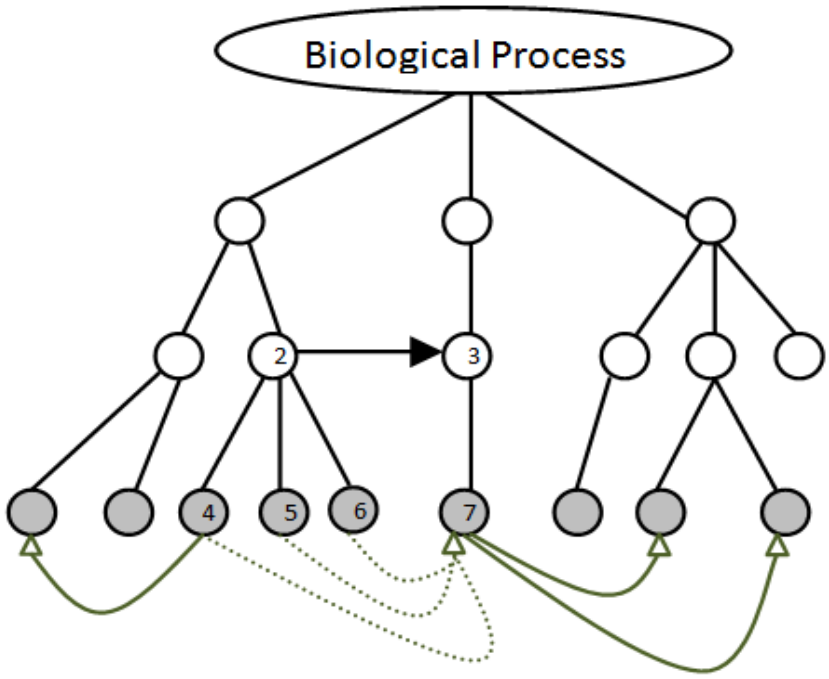

Figure 1: Schematic presentation of a simple GO regulatory network. Grey nodes represent GO terms related to the sample and the relationships between GO terms are depicted as green arrows. Parental GO nodes 2 and 3 have a relationship that can be extended to children GO nodes 4, 5, 6 and node 7, depicted as green dotted arrows.

Web Application Enhancements

Methods and algorithms were implemented in our web application [3] using PHP 5 and PostgreSQL. Because of the additional functionality to analyse the GO distribution of all expressed genes within a genome (global transcriptomics), significant memory and processing resources were required by the Apache web server. To enhance performance and husband system resources we implemented file based caching technology to cache the whole genome GO graphs. When a GO graph was built for the first time, subsequent references to that GO graph, even by other users, was instantaneous. For a better user experience in web application pages where long running tasks were performed, we used Ajax technology to implement task progress bars.

Visualising the GO interaction network

In order to visualize the enriched GO regulatory network, we used the Cytoscape [14] flash player plug-in for web. We initialized and used this component through JavaScript.

Cytoscape contains advanced dynamic network customization options such as zooming, network filtering, node re-locating, node and edge re-sizing, and colour scheming. These user-friendly options allow developers and users to dynamically change the look and feel of the network.

Case study data

To demonstrate the biological application of our new methods in global transcriptome GO analysis and GO network analysis, data from 2 previous gene expression experiments were used. *Streptococcus pneumonia* and *Salmonella enteritidis* were selected since they are responsible for high morbidity, mortality, and infection worldwide and have been well studied.

The first data set [4] was two colour microarray data from *Streptococcus pneumoniain vivo* derived RNA samples, where the relative expression of each gene in one niche was calculated in comparison to expression in the previous niche when bacteria moved from nose ⟶ lungs ⟶ blood ⟶ brain. The relative expression of all 2236 genes of *Streptococcus pneumonia* during the course of infection are presented in Additional File 1 (lung versus nose), Additional File 2 (blood versus lung), and Additional File 3 (brain versus blood). Additional files are in MS Excell worksheet format.

The second case study [10] was RNA-Seq global transcriptome data from 6 strains of *Salmonella enteritidis*, where 3 highly pathogenic strains and 3 low pathogenic strains were compared. The average whole genome expression of (4402) genes of the 3 highly pathogenic strains is presented in Additional File 4. While Additional File 5 contains the average expression of the 3 low pathogenic strains. The goal of this analysis was to unravel significantly different GO terms between highly and low pathogenic strains of bacteria using *Salmonella enteritidis* as a model.

For GO network analysis, in case study 1, the 30 highest over expressed genes in *Streptococcus pneumonia* during infection in lung versus, blood versus lung and brain versus blood were used (Additional File 6). Also, in case study 2, 18 genes with the highest fold change in expression levels between highly pathogenic strains versus low pathogenic strains are presented in Additional File 6.

Results

Introduction of gene expression levels into GO analysis

Addition of expression level data with GO term data provided the opportunity of (1) quantifying exact GO enrichments, (2) extending analysis coverage from sample-wide to genome-wide, and (3) developing statistical tests for comparison of GO distributions across transcriptomes. Considering the influences of all expressed genes in functional genomics, even those with low levels of expression, could possibly increase the accuracy of the analysis in prokaryotes.

GO regulatory network

GO regulatory networks for a sample of genes initially present three types of information: regulatory relationships between GO terms and their associated genes depicted by directed edges of the graph, enrichment levels of GO terms that are proportional to nodes' diameter of graph and finally, the genes associated with each GO term.
Furthermore, network topology revealed GO groups and their genes that had the highest number of interactions with other groups. Specifically, genes located in centre of the network were selected as good candidates for further experiments and gene discovery. In addition, the enrichment levels of GO terms that were proportional to the size of the nodes in the graph were in accordance with the regulatory relationships between GO terms.

Case studies

As case studies, we used publicly available two colour microarray and global transcriptomics data of two important bacterial pathogens, *Streptococcus pneumonia* and *Salmonella enteritidis* respectively. For each bacterium, 2 types of analysis were carried out: transcriptome based GO enrichment and GO network discovery. In *Streptococcus pneumonia*, all expressed genes were subjected to GO analysis in order to characterise functional changes in *Streptococcus pneumonia* during the course of infection. Then, using a selection of significantly up-regulated genes during infection in each tissue, GO networks were constructed to identify the central GO node and the key genes associated with the central GO node. In the *Salmonella enteritidis* case study, we first compared transcriptome GO enrichment levels between highly pathogenic and low pathogenic

strains to highlight GO functional groups correlated with pathogenicity. We then constructed the GO network using the genes which were significantly more highly expressed in pathogenic strains

Case Study 1: Changes in the transcriptome GO during *Streptococcus pneumonia* from nose ⟶ lungs ⟶ blood ⟶ brain

After downloading microarray data [4] from the NCBI GEO database for *Streptococcus pneumonia*, we selected data of strain WCH43 after 72 hours infection across 4 different tissues. We estimated the geometric means of the fold-change for each gene in the genome. The result was 3 genome-wide lists (Nose vs. Lung, Lung vs. Blood and Blood vs. Brain) each containing 2236 genes along with their mean fold-changes (Additional File 1, 2 and 3). These lists were submitted to the web server.

First, we used the pie chart visualisation to determine GO term proportions (protein enrichment distribution percentage) at different levels of the GO tree. GO term proportions of some GO groups didn't change across multiple tissues. Hence, the GO term proportions of 3 genome-wide lists were mutually compared by Kolmogorov–Smirnov test and the calculated p-values are presented in Table 1.

Table 1: Comparison of genome-wide GO enrichment levels by Kolmogorov–Smirnov test during the infection course of *Streptococcus pneumonia* from nose ⟶ lungs ⟶ blood⟶ brain

	Biological Process	**Molecular Function**	**Cellular Components**
(Lung vs. Nose ~ Blood vs. Lung)	P=0.01	P=0.01	Not significant
(Blood vs. Lung ~ Brain vs. Blood)	Not Significant	P=0.01	Not significant

Table 1 suggests that Cellular Components GO enrichment proportions did not change during the course of infection at all. Interestingly, when bacteria moved from blood to its final destination (brain), the overall proportions of Biological Process GO terms did not change.

We then produced a tabular report of the last level (most detailed) of the GO tree. From a large list of GO terms, this report highlighted GO terms that were consistently up/down regulated. Surprisingly, in this study only identified a few such GO terms (Figure 2).GO terms with upward or downward arrows had consistent up/down expression patterns. The continuously up regulated GOs were “barrier septum assembly” and tryptophan synthase activity which are involved in propagation of *Streptococcus pneumonia*. This result confirmed a known, experimentally verified mechanism in this organism [4]. The list of genes in each GO is also presented to assist with GO based gene selection. GOs such as “histidine biosynthesis process” and “amidase activity” were down regulated. This report also highlights GO terms with more than 4 fold average fold-change.

tRNA binding	37.67	4.44	213.28		2.379	SP_1910,SP_0579,SP_0631,SP_0118,SP_1554,SP_0788, SP_0881,SP_0221,SP_0216,SP_0581,SP_0768,SP_2069, SP_0208,SP_0271,SP_0234,SP_0272,SP_1383,SP_2042,	SP_1910,SP_0579,SP_0631,SP_0118,SP_1554,SP_0788, SP_0881,SP_0221,SP_0216,SP_0581,SP_0768,SP_2069, SP_0208,SP_0271,SP_0234,SP_0272,SP_1383,SP_2042,
histidine biosynthetic process	1.2	0.91	0.65	↓	0.736	SP_0825,	SP_0825,
6,7-dimethyl-8-ribityllumazine synthase activity	1.59	0.1	17.03		3.273	SP_0175,	SP_0175,
barrier septum assembly	5.98	6.15	11.21	↑	1.369	SP_1664,SP_1666,SP_0807,SP_1568,SP_1732,	SP_1664,SP_1666,SP_0807,SP_1568,SP_1732,
amidase activity	1.12	0.68	0.67	↓	0.773	SP_0965,	SP_0965,
dihydroorotate oxidase activity	4.38	0.2	81.34		4.309✓	SP_0764,SP_0964,	SP_0764,SP_0964,
isopentenyl-diphosphate delta-isomerase activity	1.43	0.15	23.24		4.031✓	SP_0384,	SP_0384,
methenyltetrahydrofolate cyclohydrolase activity	1.2	0.91	0.65	↓	0.736	SP_0825,	SP_0825,
methionine adenosyltransferase activity	1.11	0.13	18.86		4.122✓	SP_0762,	SP_0762,
methionyl-tRNA formyltransferase activity	1.91	0.28	7.25		1.948	SP_1735,	SP_1735,
methylenetetrahydrofolate dehydrogenase (NADP+) activity	1.2	0.91	0.65	↓	0.736	SP_0825,	SP_0825,
ribonucleoside-diphosphate reductase activity, thioredoxin disulfide as acceptor	3.57	1.81	1.01	↓	0.532	SP_1179,SP_1180,	SP_1179,SP_1180,
superoxide dismutase activity	2.17	0.12	47.09		4.658✓	SP_0766,	SP_0766,
thymidylate synthase activity	1.54	0.11	25.87		4.099✓	SP_0669,	SP_0669,
tryptophan synthase activity	1.29	3.03	4.39	↑	1.845	SP_1811,SP_1812,	SP_1811,SP_1812,
ribonucleoside-diphosphate reductase complex	3.57	1.81	1.01	↓	0.532	SP_1179,SP_1180,	SP_1179,SP_1180,
alcohol metabolic process	1.19	1.13	1.08	↓	0.953	SP_2026,	SP_2026,
purine nucleotide biosynthetic process	1.2	0.91	0.65	↓	0.736	SP_0825,	SP_0825,
dTMP biosynthetic process	1.54	0.11	25.87		4.099✓	SP_0669,	SP_0669,
S-adenosylmethionine biosynthetic process	1.11	0.13	18.86		4.122✓	SP_0762,	SP_0762,
polyamine biosynthetic process	5.74	0.05	146.49		5.052✓	SP_0922,	SP_0922,
superoxide metabolic process	2.17	0.12	47.09		4.658✓	SP_0766,	SP_0766,
amino acid transport	2.53	0.07	52.94		4.574✓	SP_0749,	SP_0749,
acetaldehyde dehydrogenase (acetylating) activity	1.19	1.13	1.08	↓	0.953	SP_2026,	SP_2026,
phosphoenolpyruvate-protein phosphotransferase activity	1.5	0.11	24.73		4.06✓	SP_1176,	SP_1176,
serine-type D-Ala-D-Ala carboxypeptidase activity	1.13	0.15	20.48		4.257✓	SP_0872,	SP_0872,
tagatose-bisphosphate aldolase activity	0.84	0.14	14.94		4.217✓	SP_1190,	SP_1190,
deoxyribonucleoside diphosphate metabolic process	1.1	0.86	0.36	↓	0.572	SP_1180,	SP_1180,
deoxyribonucleotide biosynthetic process	1.1	0.86	0.36	↓	0.572	SP_1180,	SP_1180,
putrescine biosynthetic process	4.06	0.07	66.98		4.062✓	SP_0921,	SP_0921,
carbon utilization	1.19	1.13	1.08	↓	0.953	SP_2026,	SP_2026,
5-formyltetrahydrofolate cyclo-ligase activity	0.92	0.15	17.38		4.346✓	SP_2095,	SP_2095,
small molecule binding	1.51	0.13	25.08		4.075✓	SP_1234,	SP_1234,
antibiotic transport	2.64	0.06	72.49		5.24✓	SP_0905,	SP_0905,
cell wall macromolecule metabolic process	1.12	0.68	0.67	↓	0.773	SP_0965,	SP_0965,

Figure 2: Amended "Table report" which lists consistently up and down regulated GO terms and also GO terms with more than 4 times change in protein enrichment.

The GO regulatory network during *Streptococcus pneumonia* infection from nose ⟶ lungs ⟶ blood ⟶ brain

The GO network during movement of *Streptococcus pneumonia* from nose to lung is presented in Figure 3A. Upon inspection, regulation of transcription (Gene Ontology ID: 6355) is a central node in the network. SP_0798 is the only component of this GO network. Interestingly, the GO group (regulation of transcription) governed by SP_0798 plays a key role in breaching the brain-blood barrier and infection of brain tissue. We previously demonstrated that the SP_0798 transcription factor positively regulates the Sp-0927 transcription factor and activates a sub network through interaction with proteins such as SP_0797, SP_0084, SP_2083, SP_1226, and SP_0799 [4]. The SP_0798 sub network is one of the key sub networks conferring high virulence to *Streptococcus pneumonia [4]*.

When comparing lung-nose niche expression patterns, the SP_0798 governed GO has interactions with GOs such as: “phosphorylation”, “fatty acid biosynthesis process”, “establishment of competence for transformation” and “oxidation-reduction process”. The “establishment of competence for transformation” GO (SP_0798 gene) can play a significant role in the translocation of *Streptococcus pneumonia* from nose to lung.

Figure 3C showed that the SP_0798 governed GO (Gene Ontology ID: 6355) had a considerable number of regulatory effects in the brain-blood comparison. The brain is the final destination of *Streptococcus pneumonia* WCH43 where it causes meningitis. SP_0798 activated different GO groups such as “metabolic process”, “establishment of competence for transformation”, “phosphorylation” and “antibiotic transport” while reaching and infecting the brain. Activation of “antibiotic transport process” helps *Streptococcus pneumonia* resist antibiotics.

It was previously [4] known that in meningitis-inducing strains of *Streptococcus pneumonia* such as WCH43, relative global gene expression significantly decreased in blood compared to the previous niche (lung) or the subsequent niche (brain). Interestingly, the GO network shown in Figure 3B helps illustrate the underlying mechanism of this global down regulation and shows that Gene Ontology ID 45892 (“negative regulation of transcription, DNA-dependent”) governed by SP_1713 transcriptional repressor NrdR is central to this relative decrease in expression. Gene Ontology ID 45892 has interactions with “CTP/GTP biosynthesis process”, “barrier septum assembly” (involved in propagation), “cytokinesis binary fission”, and “tryptophan biosynthesis process” (Figure 3B). The SP_1664 protein is involved in barrier septum assembly. SP_1813, SP_1814 and SP_1815 proteins participate in tryptophan biosynthesis process.

Discovery of the Gene Ontology ID 45892 (“negative regulation of transcription, DNA-dependent”) governed by SP_1713 and its considerable influence in suppression of genes opens a new avenue for the treatment of blood stream-based diseases such as Bacteremia and Sepsis.

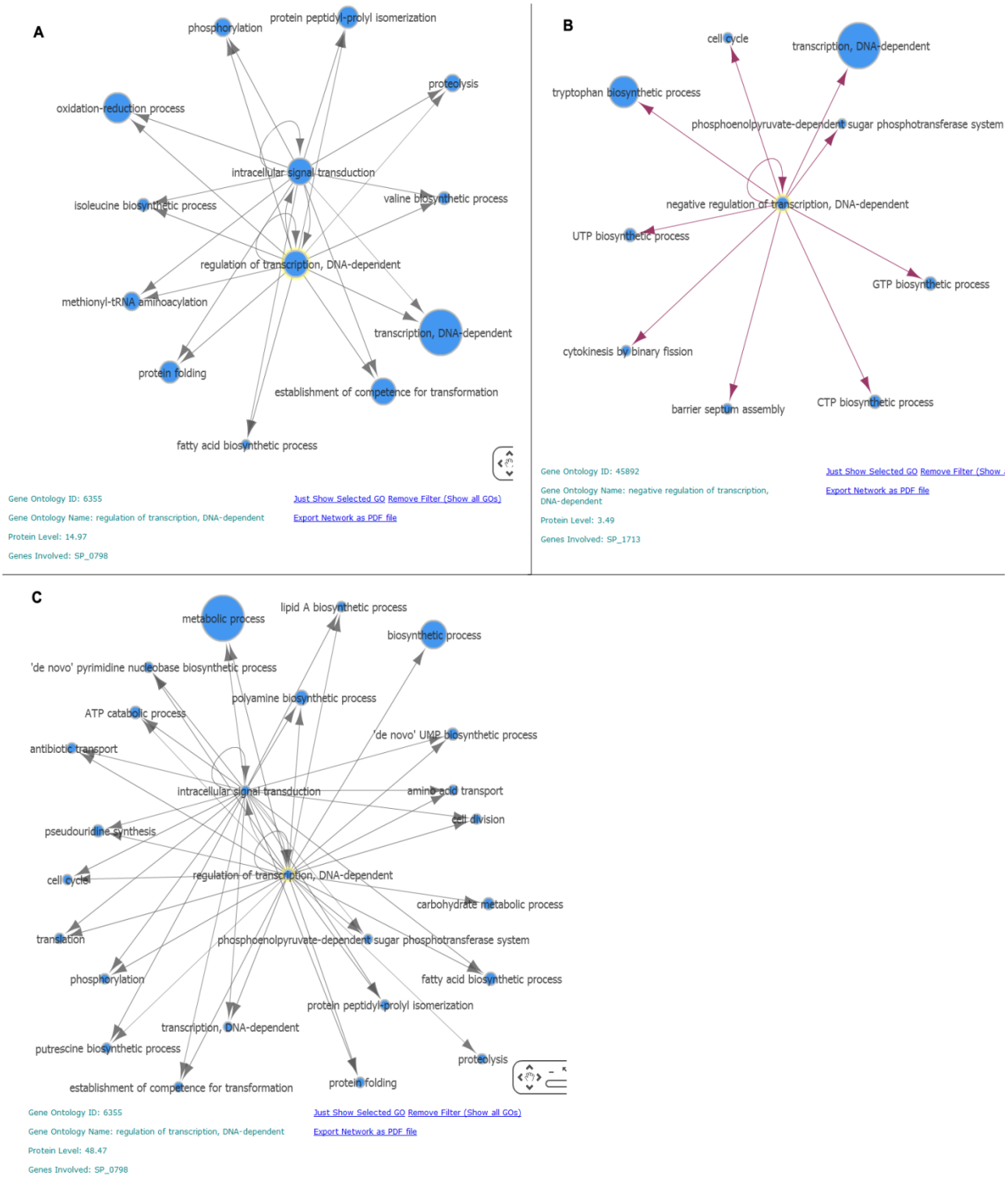

Figure 3: GO regulatory network constructed based on differentially expressed *Streptococcus pneumonia* genes in (A) Lung versus Nose (B) Blood versus Lung (C) Brain versus Blood.

Case Study 2: Comparison of whole transcriptome based GO enrichment between low and highly pathogenic *Salmonella enteritidis*

We collected RNA-Seq data for 6 strains of low and high pathogenic *Salmonella enteritidis* [10] including 3 low pathogenic strains and 3 highly pathogenic ones. We averaged the RPKM counts for each gene of the 3 low pathogenic strains and created a single list of genome expression levels. We did the same for the 3 highly pathogenic strains (Additional File 4 and 5). After submission of both gene lists (4402 genes for each one) to the web server, we used the pie chart to visualise the GO term proportions and navigate the GO term tree. The comparison revealed very similar GO proportions at nearly all levels of the GO tree. This encouraged us to perform hypothesis tests to compare the GO enrichment proportions between low and highly pathogenic strains. Table 2 shows the result of this comparison for Biological Process, Molecular Function, and Cellular Components.

Table 2: Comparison of genome wide GO enrichment levels of low pathogenic strains of *Salmonella enteritidis* versus high pathogenic strains by Kolmogorov–Smirnov test

	Biological Process	***Molecular Function***	***Cellular Components***
Low Pathogenic strains Vs. HighPathogenic strains	P value =0.86	P value = 0.34	0.7590978

Based on a 0.05 level of significance for our tests, Table 2 indicates that there is probably no significant difference in GO protein enrichment proportions between low and highly pathogenic strains of *Salmonella enteritidis* bacteria. This suggests that the change from low pathogenic strain to highly pathogenic strain is not associated with a global shift in GO term proportions. To verify this idea further, one can perform equivalence tests from more samples. In general, a non-significant difference must not be considered as significant homogeneity [17]. However, as seen below, a shift in a subset of GO terms can be associated with higher pathogenicity.

GO regulatory network changes between high and low pathogenic strains of *Salmonella enteritidis*

A list of the most differentially expressed genes - with greater than 10 fold change - were submitted to the Web server (Additional File 6), including fljB, SEN1084, motA, flgK, cheA, invF, invA, invG, ,fliD, prgH, osmY, , ipB, sipC, yeaG, sipA, dps, yjbJ, and bfr.
The resulting GO network is presented in Figure 4.
Interestingly, the GO term "signal transduction by phosphorylation" (Gene Ontology ID: 23014) is central in the overrepresented GO expression network of highly pathogenic *Salmonella enteritidis* strains. The protein kinase encoded by cheA is the sole component of "signal transduction by phosphorylation process". This shows that higher pathogenicity in *Salmonella enteritidis* appears to be associated with increased signal transduction and phosphorylation. We speculate that up regulating GO "Signal transduction by phosphorylation" may allow *Salmonella enteritidis* to more rapidly sense environmental changes and activate more genes through stronger phosphorylation activity. "Response to stress", "iron ion transport" (bfr gene), "pathogenesis", "transcription DNA dependent", "protein phosphorylation" (yeaG gene) and "chemotaxis" are the other GO terms which are differentially expressed in highly pathogenic strain.

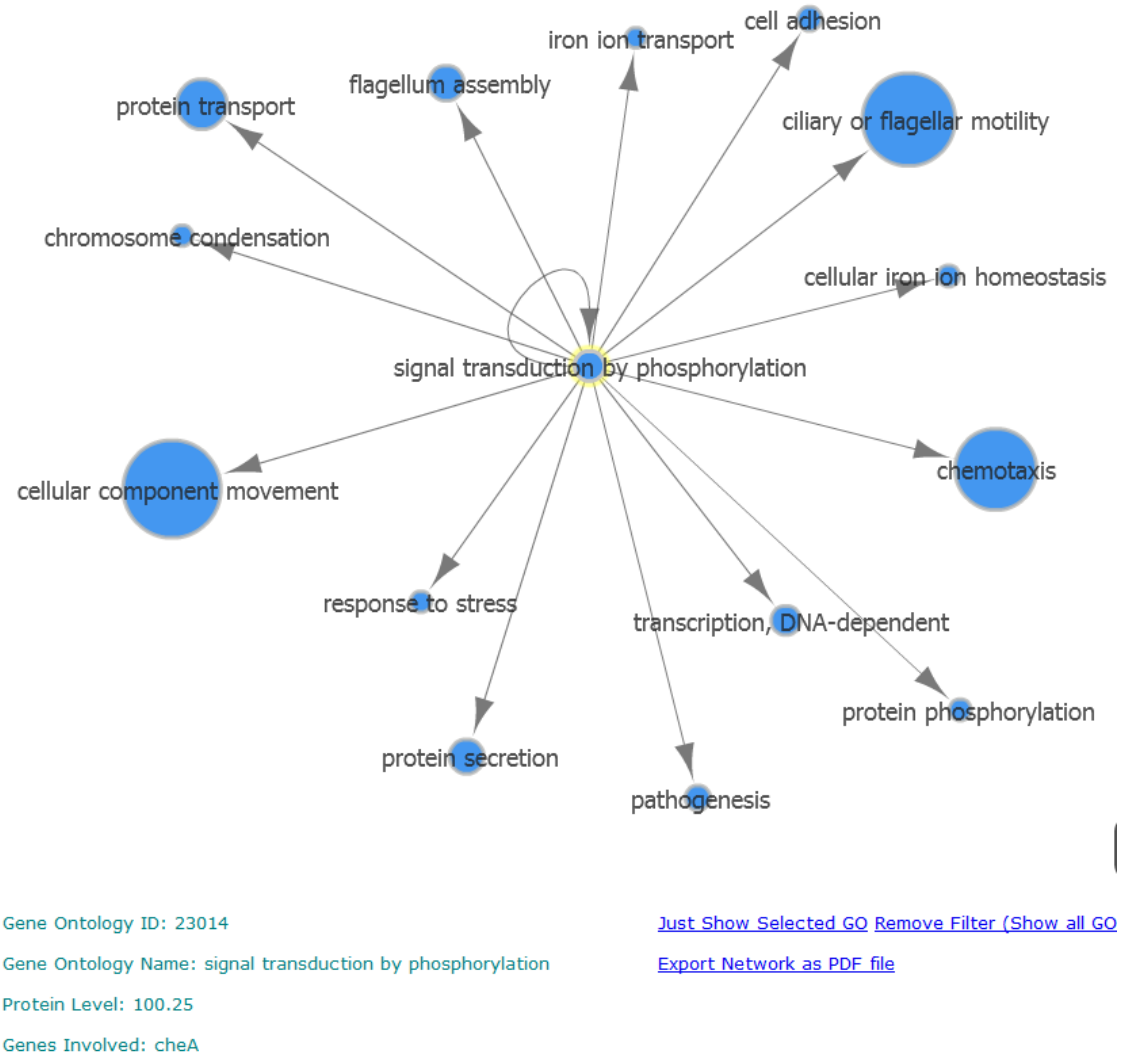

Figure 4: GO regulatory network based on 18 genes with significant differential expression levels in highly pathogenic versus low pathogenic *Salmonella enteritidis* strains.

Commonality between GO Regulatory Networks of Case Studies

Selection of *Streptococcus pneumonia*during the course of infection in nose, blood, and brain of host allowed us to apply whole genome based GO enrichment and GO in study of tissue-based pathogenesis and breaking host barriers by pathogen. In addition, comparative study of GO enrichment and GO network between highly pathogenic and low pathogenic strains of *Salmonella* provided to investigate mechanisms involved in generation of highly pathogenic strains using GO concept.

Go network analysis in *Streptococcus pneumonia* and *Salmonella enteritidis* resulted in detection of new biological results and genes that were not reported in original works.

Furthermore, central roles of GO classes of “regulation of transcription” and “signal transduction by phosphorylation” governed by SP_0798 and cheA in induction of pathogenesis were unravelled. Phosphorylation, performed by kinases, is one of the main pathways of rapid signal response and gene activation. Interestingly, even in plants, protein kinases are the central compartment of inducing high stress resistance and evolution [18].

cheA(chemotaxis protein CheA)is a sensor histidine kinase and a member of two-component system. cheA is majorly involves in “Environmental Information Processing” and “Signal transduction” (KEGG database [19]). According to Pfam database [20], cheA contains the following domains: PF01584(CheW-like domain), PF01627 (Hpt domain), PF02518(Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase), PF02895 (Signal transducing histidine kinase, homodimeric domain), PF09078(CheY binding), and PF13589(Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase).

SP_0798 is a DNA-binding response regulator CiaR and a member of two-component system. According to Pfam database [20], SP_0978 contains PF0072 (response regulator receiver domain) and PF00486 (Transcriptional regulatory protein, C terminal). Similar to cheA, SP_0798 is also involved in “Environmental Information Processing”, “Signal transduction” and two-component system (KEGG database [19]). It can be concluded that SP_0798 and cheA are ortholog to each other.

Based on the above discussion and the observed similar observed mechanism between *Streptococcus pneumonia* and *Salmonella enteritidis*, it can be suggested that “Environmental Information Processing” which carries ON by “Signal transduction” and two-component system pathways are methods of choice by bacterial pathogens in increasing pathogenicity, host barrier breaking and generation of new strains. In fact, successful pathogens such as *Streptococcus pneumonia* and *Salmonella enteritidis* are developing expert systems to recognise faster external environment and also react more promptly by a more efficient signal transduction system. Two-component system is a head-tail pathway which one member sits outside the cell and other member inside the cell and informs the bacteria about environmental signals/changes. Rapid recognition of environmental alterations such as antibiotic stress and nutrient change allows bacteria to act more rapidly and increase the chance of surviving. Two-component system has a confirmed role in bacterial virulence [21, 22].

On the other hand, SP_1713 is the major player of negative regulation in blood infection of *Streptococcus pneumonia*. The fact that SP_1713 has the ability to regulate a large number of other gene ontology terms and dramatically decreases the global transcriptome expression levels in blood, offers a new possibility for treatment of blood-based infections such as Bacteremia and Sepsis. This example shows how GO network construction can be employed for discovery of key GO groups and GO based gene selection.

Discussion

GO analysis provides a new avenue for a deeper understanding of gene expression and function, which can be exploited in the context of quality-based gene selection strategies [3, 4]. While other GO web servers [7, 23] support gene annotation in model eukaryotes via user submitted gene lists that must match the single source of annotation used by the server, our web server supports all sequenced prokaryotes and viruses and automatically recognizes gene names from all annotation sources.

In contrast to other web servers, our web server provides interactive visual navigation of the hierarchical tree structure of GO groups weighted according to gene expression values at all levels. Furthermore, our server provides dynamic visual reports (using AJAX technology) such as pie charts (to visualize GO group proportions) and bar charts (to compare GO term enrichments versus reference genome based on hyper-geometric distribution), whereas other web servers present this information in text format or rely on visualization capacity provided by other websites [24].

The most significant analytical advantage provided by our web server is the ability to compare GO terms across multiple gene samples (or whole genomes) from multiple biological conditions. At present other web servers [7, 23] can only compare one sample against a reference genome. Comparative GO analysis is particularly important as a means to identify the underlying biological pathways recruited under different biological conditions. This is an essential method if one wishes to identify important genes for perturbation experiments.

Unlike other GO web servers that compare one GO term compared to a reference genome at a time (using the Fisher Exact test), our web server can compare all the GO term enrichments from two or more samples (or whole genomes) simultaneously by using

robust non-parametric statistical tests. This enables detection of any global significant shift in GO enrichment levels as experimental conditions change.

Finally, our comparative table report takes into account protein enrichment to detect GO terms with special enrichment patterns or with specific enrichment fold-change across multiple samples. This helps identify key GO terms and their associated genes because their expression prevalence. At present, this is a unique analytical approach that is not found elsewhere.

Global transcriptome based GO analysis was achieved by integrating gene expression levels with GO classifications. This allowed us to compare GO enrichment that better reflected the biological reality of the experiments across multiple samples by taking into account the abundance of gene products. This type of comparison was not previously possible, most likely because the prevalence of eukaryotic GO databases and web servers [7] would not have benefited from such an analysis. Current GO web applications are mostly developed in eukaryotic genomes [5-7] where protein abundance levels are poorly correlated with gene expression levels, making the need for transcript abundance weighting less relevant.

In this report we have presented a method to build GO regulatory networks using public Gene Ontology data [11]. GO regulatory networks from differentially expressed genes can reveal underlying biological pathways [25]. In particular the topology of such networks can highlight highly connected/central GO terms and their associated genes, supporting the discovery of candidate genes.

Furthermore, by looking at networks from different bacterial species we can elucidate common biological pathways. Even though we have only implemented GO regulatory networks for bacteria, this type of network could be very effective for eukaryotes as well, particularly for proteomics data. To our knowledge, no current GO web server provides this capability.

We have also demonstrated how to combine a GO regulatory network with gene expression data. The resultant network can be used to study regulatory effects of genes and GOs on each other. For example, by comparing and overlapping multiple GO regulatory networks for the same genes across multiple biological conditions, we can detect areas of the network that confirm or contradict expected regulatory relationships. This can be used as a mean to support or question the validity of original transcriptomic data or indicate the existence of any unknown environmental effects in the experiment.

Moreover, by replacing the GO regulatory network's nodes with their associated genes one can generate a GO-based gene regulatory network (GRN).

Finally, combining GO-based gene regulatory networks with other types of gene regulatory networks [25] (those that are reverse engineered from transcriptome data) such as co-expression networks [26, 27] can lead to the discovery of unknown biological entities or biological mechanisms, particularly where such results contradict one another.

Together, the global transcriptomics based GO enrichment and GO regulatory network, developed in the present investigation and implemented in Comparative GO Web application [3, 8] can significantly increase the knowledge of bacterial regulatory mechanisms of pathogenesis as well as functional genomics arrangements which result in emerging new highly pathogenic strains.

Conclusion

We applied whole transcriptome data and gene expression levels to GO classification analysis leading to new meaningful biological reports. We have also developed a method to dynamically construct GO regulatory networks for any given sample. Finally, we have demonstrated the efficiency of our developed methods and tools through case studies on two types of bacteria. The results of these analyses either identified new candidate genes and GO terms that were not reported in the original work or confirmed the functionality of known genes.

Availability of supporting data

The data sets supporting the results of this article are included within the article and its additional files

List of abbreviations

GO: Gene Ontology; DAG: Directed acyclic Graph; GRN: Gene Regulatory Network; RPKM: Reads Per Kilo Base Per Million;

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

Conceived and designed the methodology and experiments: MF, EE, DLA

Performed the experiments: MF, EE

Data Analysis: MF, EE, DLA

Wrote the paper: MF, EE, DLA

All authors read and approved the final manuscript.

Acknowledgments

- We would like to greatly thank Dr.Abiodun Ogunniyi, Dr.Layla Mahdi and Prof. James Paton from the Research Centre for Infectious Diseases of The University of Adelaide for their comments and help. We would also like thank Dr. Dan Kortschak for his helpful comments.

References

1. Cogoni C, Macino G: **Post-transcriptional gene silencing across kingdoms**. *Current opinion in genetics & development* 2000, **10**(6):638-643.
2. Taniguchi Y, Choi PJ, Li G-W, Chen H, Babu M, Hearn J, Emili A, Xie XS: **Quantifying E. coli proteome and transcriptome with single-molecule sensitivity in single cells**. *Science* 2010, **329**(5991):533-538.
3. Fruzangohar M, Ebrahimie E, Ogunniyi AD, Mahdi LK, Paton JC, Adelson DL: **Comparative GO: A Web Application for Comparative Gene Ontology and Gene Ontology-Based Gene Selection in Bacteria**. *PloS one* 2013, **8**(3):e58759.
4. Mahdi LK, Ebrahimie E, Adelson DL, Paton JC, Ogunniyi AD: **A transcription factor contributes to pathogenesis and virulence in Streptococcus pneumoniae**. *PloS one* 2013, **8**(8):e70862.
5. Conesa A, Götz S: **Blast2GO: A comprehensive suite for functional analysis in plant genomics**. *International journal of plant genomics* 2008, **2008**.
6. Conesa A, Götz S, García-Gómez JM, Terol J, Talón M, Robles M: **Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research**. *Bioinformatics* 2005, **21**(18):3674-3676.

7. Da Wei Huang BTS, Lempicki RA: **Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources**. *Nature protocols* 2008, **4**(1):44-57.
8. **Comparative GO: A Web Application for Comparative Gene Ontology Analysis** [genomes.ersa.edu.au/BacteriaGO/]
9. Press WH, Teukolsky SA, Vetterling WT, Flannery BP: **Numerical Recipes: The art of scientific computing (Cambridge**. In.: Cambridge Univ. Press; 1992.
10. Shah DH: **RNA-Seq reveals differences in the global transcriptome between high-and low-pathogenic Salmonella Enteritidis strains**. *Applied and environmental microbiology* 2013:AEM. 02740-02713.
11. **Gene Ontology Database** [http://www.geneontology.org/]
12. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT: **Gene Ontology: tool for the unification of biology**. *Nature genetics* 2000, **25**(1):25-29.
13. Consortium GO: **The gene ontology: enhancements for 2011**. *Nucleic acids research* 2012, **40**(D1):D559-D564.
14. Saito R, Smoot ME, Ono K, Ruscheinski J, Wang P-L, Lotia S, Pico AR, Bader GD, Ideker T: **A travel guide to Cytoscape plugins**. *Nature methods* 2012, **9**(11):1069-1076.
15. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES: **Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles**. *Proceedings of the National Academy of Sciences of the United States of America* 2005, **102**(43):15545-15550.
16. Team RC: **R: A language and environment for statistical computing**. *R foundation for Statistical Computing* 2005.
17. Wellek S: **Testing statistical hypotheses of equivalence and noninferiority**: CRC Press; 2010.
18. Alimohammadi A, Shiran B, Martínez-Gómez P, Ebrahimie E: **Identification of water-deficit resistance genes in wild almond< i> Prunus scoparia</i> using cDNA-AFLP**. *Scientia Horticulturae* 2013, **159**:19-28.
19. **Kyoto Encyclopedia of Genes and Genomes** [http://www.genome.jp/kegg/]
20. **Pfam Protein Database** [http://pfam.sanger.ac.uk/]
21. Miller SI, Kukral AM, Mekalanos JJ: **A two-component regulatory system (phoP phoQ) controls Salmonella typhimurium virulence**. *Proceedings of the National Academy of Sciences* 1989, **86**(13):5054-5058.
22. Stibftz S, Aaronson W, Monackt D, Falkowt S: **Phase variation in Bordetella pertussis by frameshift mutation in a gene for a novel two-component system**. 1989.
23. Al-Shahrour F, Díaz-Uriarte R, Dopazo J: **FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes**. *Bioinformatics* 2004, **20**(4):578-580.
24. **The European Bioinformatics Institute** [http://www.ebi.ac.uk/]
25. Zinman GE, Zhong S, Bar-Joseph Z: **Biological interaction networks are conserved at the module level**. *BMC systems biology* 2011, **5**(1):134.
26. Langfelder P, Horvath S: **WGCNA: an R package for weighted correlation network analysis**. *BMC bioinformatics* 2008, **9**(1):559.
27. Liu LZ, Wu FX, Zhang WJ: **Reverse engineering of gene regulatory networks from biological data**. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2012, **2**(5):365-385.

5 Summary and Conclusion

As the number of published biomedical articles has grown dramatically, the task of manually reading and extracting biological facts from them has become nearly impossible. Hence, automating the task of extracting biological relationships is a crucial need within the biomedical research community.

Literature mining refers to whole process of scanning and analysing text, extracting biological relationships, storing in database and finally classify and present them through meaningful biological reports.

In general, extracted biological relationships are stored in public biological databases in order to be shared by researchers all around the world. There are a wide range of biological databases that store annotations related to genes and proteins, including interactions, biological functions, alleles, SNPs (single nucleotide polymorphism), diseases and drugs. Of these databases, the Gene Ontology database is a key database that connects other biological entities through a standard controlled vocabulary.

In this thesis I developed new methods and tools for all stages of literature mining in order to improve current methods and tools. I also compared the efficiency of my methods with those currently in use. I have divided my work into three parts and that are discussed in three sections.

In the first part of my thesis I developed a set of methods and tools for scanning and analysing biomedical texts that include: an article reader, a sentence detector, a sentence tokenizer, a POS tagger and finally a noun and verb phrase detector. I used the Java programming language and the PostgreSQL database to implement these tools. The POS tagger was the most sophisticated of these tools to implement, as predicting the POS tag of an unknown biomedical word is a very challenging problem.

I implemented a fully functional HMM POS tagger and I devised a method to predict the POS tag of an unknown word based on its suffix and other character features including capital letters, numbers and special characters. I compared my method with the only other published method and showed how my method significantly outperformed the other method. I also introduced the concept of counting methods in MLE parameter estimation

(based on including or excluding coefficients), and I showed how the counting method affects the accuracy of suffix based POS tagging.

Some common (non-biologically specific) English words were incorrectly POS tagged in all the methods (e.g. breathe, bring, obese). This was because of their similarity with common known suffixes. As the number of such common English words is limited (less than 1 percent of all unknown words), I proposed to manually add them to my machine's internal lexicon. However, this solution is not suitable if you want to tag an unknown biological word, because these accounted for more than 99% of all unknown words in my dataset. Fortunately I was able to show that using my proposed suffix and character feature-based method correctly tagged 95% of unknown biological words.

In order to evaluate the overall performance of my tools, I compared them with a well-established POS tagger called Maxent Tagger. My sentence tokenising method tokenised biomedical sentences much better than the Maxent tokeniser. While the Maxent POS tagger was better at tagging common English words, perhaps due to a larger internal lexicon, my POS tagger out-performed Maxent tagger when tagging unknown biomedical words as a result of its efficient suffix and character feature analysis. Finally, I showed how one can use suffix analysis to estimate the conditional probability of known lexicon words for unseen POS tags.

As previously mentioned, GO analysis is critical if one is to understand biological pathways and their associated genes. Current GO web servers primarily work with eukaryotic genomes and they lack visual and comparative statistical analysis, which limits them and specifically makes them unsuitable for bacterial studies.

In the second part of this thesis, I first built a comprehensive up to date database (using PostgreSQL) for genes, proteins, taxonomy and GO annotations of bacterial and viral species. I then designed a user-friendly web application (using PHP technology) to visualise GO term analysis and compare GO term enrichments of multiple gene samples. As GO terms build a directed acyclic graph, I provided navigational access to all levels of the GO graph in all of the visual reports using AJAX and JavaScript web technologies.

I implemented a pie chart visualisation tool to compare GO enrichments across multiple samples' at any level of the GO graph. This report visually revealed the GO enrichment shifts across multiple samples and was able to identify the specific genes involved.

Moreover, I also implemented a bar chart comparing a selected sample's GO distribution compared to a reference genome based on the hyper-geometric distribution. While this report can help researchers visualise any over/under represented GO group, other web servers only provide a user un-friendly text table with p-values from the Fisher Exact test.

While my web server is able to compare any number of gene samples from different biological sources simultaneously, other web servers only compare one sample to its reference genome. This critical feature of my web server makes it suitable to study biological pathways and pathogenesis of bacteria across multiple host tissues or biological conditions.

Furthermore, I implemented a multi level tabular report to compare GO enrichment from multiple numbers of samples in one place. This report is specifically designed for GO-based candidate gene selection. In this report, each GO term shows the overall fold change of estimated GO enrichment values and reports common genes associated with these terms. In addition, this report highlights any GO terms that have been continuously up or down regulated. The genes suggested by this report can be good candidates for further experiments. As a case study, *Streptococcus pneumonia* was used as a model. Bacterial samples harvested from four tissues (nose, lung, blood and brain) underwent microarray gene expression profiling. From this, 3 lists of differentially expressed genes were prepared and were submitted to the web server. Most of the observed results either confirmed our current knowledge about this bacterium or provided more insight into the molecular machinery of pathogenesis, suggesting novel genes for further experiments. For instance, the pie chart report revealed that 'Metabolic Process' protein levels were significantly reduced in blood compared to lung, but increased again in brain. Furthermore, the bar chart based on the hyper geometric distribution showed that 'ATP binding' was significantly over-represented in lung compared to the reference genome and implicated and associated gene SP_0788. Finally, the comparative tabular report detected SP_0676 as the only gene in all samples that participated in 'Sequence Specific DNA Binding Transcription'.

In the third part of this thesis, I integrated gene expression levels with GO enrichment analysis. This type of analysis is eminently applicable to bacterial species, because gene expression levels are known to be directly proportional to protein expression in bacteria.

As the result of this integration, I produced more meaningful biological reports. For instance, in the case study of *S.Pneumonia,* the comparative table report revealed that protein levels for "barrier septum assembly" were continuously up-regulated in during pathogenesis. Finally, I have shown that the influence of all expressed genes, even those with low levels of expression, most likely increased the accuracy of this analysis in prokaryotes.

Another advantage of this integration was the ability to enrich GO terms in the entire transcriptome (instead of samples of a short list of genes), enablingme to compare GO terms of transcriptomes across multiple biological conditions. In another case study, I compared *Salmonella enteritidis* high versus low pathogenic strains. Non-parametric statistical tests revealed that GO term proportions across the entire genome did not change between high and low pathogenic strains. However, the same analysis for *S.Pneumonia,*showed that "Biological Process" and "Molecular Function" proportions changed significantly (but not "Cellular Component") during pathogenesis.

In this thesis, I devised a method to construct a dynamic GO regulatory network for any given sample of genes. Taking advantage of the fact that "Biological Process" groups imply regulatory relationships I could show that the GO regulatory network for differentially expressed genes between 2 biological conditions revealed an underlying biological pathway in those conditions. In particular, the topology of such a network highlights central GO term groups and their associated genes, allowing them to be used to discover novel candidate genes.

As a case study I compared the GO regulatory networks of *S.Pneumonia* across host tissues. From the networks I produced I discovered that SP_0798 plays a key role in breaching the blood-brain barrier and subsequent infection of brain tissue. In addition, discovery of the significant "negative regulation of transcription, DNA-dependent" GO term associated with SP_1713, and its considerable influence in suppression of genes has opened a new avenue for the treatment of blood stream-based diseases such as Bacteremia and Sepsis.

Replacing the nodes of a GO regulatory network with their associated genes resulted in a GO-based gene regulatory network (GRN). Merging GO-based gene regulatory networks with other types of gene regulatory networks such as co-expression networks could

increase the accuracy of subsequent biological interpretation and provide better candidate gene selection.

In conclusion, the global transcriptome based GO enrichment and GO regulatory networks, developed in this thesis and implemented in the Comparative GO Web application can significantly increaseour knowledge of bacterial regulatory mechanisms governing pathogenesis as well as functional genomic changes resulting in emerging new highly pathogenic strains.

6 Supporting Information

6.1Supporting Information for chapter 2

Table S1: Table of POS tags used in our experiment:

CC coordinating conjunction	NNS plural noun	VBN participle been	VVD past tense
CS subordinating conjunction	PN pronoun	VBZ 3rd present	VVG present part
CSN comparative conjunction	PND determiner as pronoun	VDB base do	VVI infinitive lexical verb
CST complementizer	PNG genitive pronoun	VDD past did	VVN past participle
DB predeterminer	PNR relative pronoun	VDG participle doing	VVZ 3rd present
DD determiner	RR adverb	VDI infinitive do	VVNJ prenominal past part.
EX existential	RRR comparative adverb	VDN participle done	VVGJ pronominal present part.
GE genitive marker	RRT superlative adverb	VDZ 3rd present	VVGN nominal gerund
II preposition	SYM symbol	VHB base have	(left parenthesis
JJ adjective	TO infinitive marker	VHD past had	) right parenthesis
JJR comparative adjective	VM modal	VHG participle having	, comma
JJT superlative adjective	VBB base be	VHI infinitive have	. end of sentence
MC number	VBD past was, were	VHN participle had	: colons
NN noun	VBG participle being	VHZ 3rd present	`` left quote
NNP proper noun	VBI infinitive be	VVB base form lexical verb	'' right quote

Table S2

Sample	MSL, Freq_1	MSL, Freq_n	PIM, Freq1,Int1	PIM, Freq1, Int2	PIM, Freq1, Int3	PIM, Freq_n, Int1	PIM, Freq_n, Int2	PIM, Freq_n, Int3
1	90	90	4	5	9	3	3	5
2	84	84	8	7	8	4	4	7
3	89	90	4	4	8	2	1	2
4	91	92	2	1	4	2	2	3
5	86	86	6	5	9	4	3	7
6	91	94	2	1	4	0	0	4
7	91	92	1	1	5	3	3	3
8	84	85	5	5	9	4	3	7
9	85	84	3	3	5	3	3	5
10	89	91	3	3	9	3	3	7
11	92	91	2	2	9	1	0	6
12	94	96	2	2	5	2	2	5
13	88	86	1	1	12	5	5	10
14	84	85	5	5	8	2	2	4
15	89	87	6	5	4	2	2	2

MSL: Maximum Suffix Length Method*PIM:*ProbabilityInterpolation Method

6.2 Supporting Information for chapter 3

Appendix: List of differentially expressed genes in *Streptococcus pneumonia*

Lung vs. Nose	Blood vs. Lung	Brain vs. Blood
SP_0432	SP_0211	SP_0133
SP_0440	SP_0263	SP_0225
SP_0675	SP_0538	SP_0325
SP_0676	SP_1044	SP_0326
SP_0677	SP_1045	SP_0327
SP_0678	SP_1109	SP_0328
SP_0683	SP_1329	SP_0333
SP_0684	SP_1430	SP_0334
SP_0685	SP_1517	SP_0335
SP_0686	SP_1545	SP_0336
SP_0692	SP_1673	SP_0341
SP_0693	SP_1752	SP_0342
SP_0694	SP_1860	SP_0343
SP_0699	SP_2074	SP_0344
SP_0702	SP_2182	SP_0349
SP_0771	SP_2237	SP_0350
SP_0772		SP_0351
SP_0773		SP_0352
SP_0774		SP_0421
SP_0779		SP_0422
SP_0780		SP_0423
SP_0781		SP_0424
SP_0782		SP_0429
SP_0787		SP_0430
SP_0788		SP_0431
SP_0789		SP_0432
SP_0790		SP_0437
SP_0795		SP_0438
SP_0796		SP_0439
SP_0797		SP_0440
SP_0798		SP_0445
		SP_0446
		SP_0447
		SP_0448
		SP_0579
		SP_0580
		SP_0581
		SP_0582
		SP_0587
		SP_0589
		SP_0590
		SP_0595
		SP_0596
		SP_0597
		SP_0603
		SP_0604
		SP_0605
		SP_0606
		SP_0675
		SP_0676
		SP_0677
		SP_0678
		SP_0683
		SP_0684
		SP_0685
		SP_0686
		SP_0691
		SP_0692
		SP_0693
		SP_0694

		SP_0699 SP_0700 SP_0701 SP_0702 SP_0739 SP_0740 SP_0741 SP_0742 SP_0747 SP_0748 SP_0749 SP_0750 SP_0755 SP_0756 SP_0757 SP_0758 SP_0763 SP_0764 SP_0765 SP_0766 SP_0771 SP_0772 SP_0773 SP_0774 SP_0779 SP_0780 SP_0781 SP_0782 SP_0787 SP_0788 SP_0789 SP_0790 SP_0795 SP_0796 SP_0797 SP_0798 SP_0885 SP_0903 SP_0904 SP_0905 SP_0906 SP_0911 SP_0912 SP_0913 SP_0914 SP_0919 SP_0920 SP_0921 SP_0922 SP_0927 SP_0928 SP_0929 SP_0930 SP_1159 SP_1324 SP_1605 SP_2111

6.3 Supporting Information for chapter 4

File Name	File Format	Title of Data	Description of Data
AdditioanlFile1.xlsx	Xlsx	transcriptome	*streptococcus pneumonia* Whole transcriptome Relative Expression Levels Lung vs. Nose (2 Colour Microarray)
AdditioanlFile2.xlsx	Xlsx	transcriptome	*streptococcus pneumonia* Whole transcriptome Relative Expression Levels Blood vs. Lung (2 Colour Microarray)
AdditioanlFile3.xlsx	Xlsx	transcriptome	*streptococcus pneumonia* Whole transcriptome Relative Expression Brain vs. Blood (2 Colour Microarray)
AdditioanlFile4.xlsx	Xlsx	transcriptome	*Salmonella enteritidis* global transcriptome high pathogenic (average of 3 strains) RPKM Counts
AdditioanlFile5.xlsx	Xlsx	transcriptome	*Salmonella enteritidis* global transcriptome low pathogenic (average of 3 strains) RPKM Counts
AdditioanlFile6.xlsx	Xlsx	samples	List of Differentially Expressed Genes in Case Study 1 and Study 2 for GO Network Analysis