PHD

A computational approach to the identification of lineage-specific bacterial genes and a determination of their biological significance

Wilson, Gareth Anthony

Award date:
2007

Awarding institution:
University of Bath

A Computational Approach to the Identification of Lineage-Specific Bacterial Genes and a Determination of their Biological Significance

Gareth Anthony Wilson

A thesis submitted for the degree of Doctor of Philosophy

University of Bath

Department of Biology and Biochemistry

May 2007

Contents

List of Tables

List of Figures

List of Abbreviations

ANOVA	Analysis of Variance
BLAST	Basic Local Alignment Search Tool
BLOSUM	Blocks Substitution Matrix
CDS	Coding Sequences
CEH	Centre for Ecology and Hydrology
DNA	Deoxyribonucleic Acid
EBI	European Bioinformatics Institute
E-value	Expect Value
EMBL	European Molecular Biology Laboratory
FAQ	Frequently Asked Questions
FIG	Fellowship for Interpretation of Genomes
FTP	File Transfer Protocol
GeneRIF	Gene Reference into Function
GFF	Generic Feature Format
GOS	Global Ocean Sampling
GSC	Genomic Standards Consortium
GUI	Graphical User Interface
HOPs	Heterogeneous Occurrence in Prokaryotes
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IIO	Isolation Index of Organism
IMG	Integrated Microbial Genomes
IMM	Interpolated Markov Model
INSDC	International Nucleotide Sequence Database Collaboration
JGI	Joint Genome Institute
JNLP	Java Network Launching Protocol

mRNA	Messenger RNA
NCBI	National Center for Biotechnology Information
NCBI-nr	NCBI's non-redundant sequence database
ND	Neighbourhood Distribution
NEBC	NERC Environmental Bioinformatics Centre
NERC	Natural Environment Research Council
ORF	Open Reading Frame
PAM	Percent Accepted Mutations
PDB	Protein Data Bank
QIPP	Quality Index for Predicted Proteins
RNA	Ribonucleic Acid
RefSeq	NCBI Reference Sequence Project
RDBMS	Relational Database Management System
rRNA	Ribosomal RNA
RT-PCR	Reverse Transcription Polymerase Chain Reaction
SQL	Standard Query Language
TCP	Toxin Co-regulated Pili
TRG	Taxonomically Restricted Genes

Acknowledgments

I would like to thank NERC and CEH for providing the funding and resources needed to perform my research. I would also like to take this opportunity to thank the people who gave me the support necessary to complete this thesis. Firstly, I thank Dr Dawn Field, my supervisor at CEH Oxford, for her encouragement, ideas and enthusiasm. My supervisor at the University of Bath, Dr Ed Feil, should also be thanked for providing direction and advice when required. I would also like to express my gratitude towards members of CEH, CCS and NEBC for their assistance and technical expertise, particularly Nicolas Bertrand, Seb Adams and Chimdi Ekeke. I have also appreciated the input from Sarah Turner, both as a user of the QuickMine software and for reading sections of the thesis.

I have also appreciated the input and critical reading of Andrew Spiers, Rob Edwards, Dave Ussery, Chris Taylor, Peter Sterk, Mike Allen and Jack Gilbert. Keith Keller should also be thanked for providing the mapping of unique identifiers, for use in the MicrobesOnline database.

Finally, I would like to thank my family, friends and colleagues both from CEH Oxford and from around the country, particularly Paul Swift and Ian Robinson, for making the working day that bit more enjoyable, my parents for providing a listening ear if required and of course Rebecca Chanda for giving me all the support and encouragement I could have wished for.

Abstract

Lineage-specific genes, especially those which are species- and strain-specific, are of special interest because they are expected to play a role in defining exclusive ecological adaptations to particular niches. Despite this, they are relatively poorly studied and little understood, in large part because many are still unique to a particular isolate (termed orphan genes), or only possess homologues in very closely related isolates. This lack of homology confounds attempts to establish the likelihood that a hypothetical gene is expressed and, if so, to determine the putative function of the protein.

The QuickMine software package and OrphanMine database were written to enable the identification and exploration of lineage-specific genes in bacterial and archaeal genomes. Analysis of this data indicates that, despite expectations to the contrary, the number of orphan genes in our collection of complete bacterial genome sequences is continuing to increase as more genomes are sequenced.

Additionally, it was found that genes restricted to a small number of isolates tend to have certain sequence properties that differentiate them from more conserved coding regions. The index, 'Quality Index for Predicted Proteins' (QIPP), was created for assessing the quality of a predicted protein, based on the combined features of its coding sequence (length, percentage low complexity, G+C content, amino acid cost, and neighbourhood distribution). These five criteria were selected for their ability to detect purifying selection and therefore, provide a means to gauge the probability that the sequence encodes a functional protein. This index can be used to prioritise genes for further experimental characterisation. The QIPP score can also provide an indication of the likely degree of conservation of a particular sequence. Additionally, the score correlates well with functional categories and can be used to estimate the amount of functional information available for a sequence.

The challenge of understanding orphan and poorly characterised genes will not be solved by simply generating additional sequence data. Instead, new methods need to be developed to help characterise proteins. QIPP, in the absence of homology, provides an important step forward in the standardisation and automation of identifying biologically important genes.

CHAPTER 1

Orphan Genes – A Figment of our Annotation or Visitors from an Unknown World?

1.1 Overview

In 1995, the sequencing of the bacterium *Haemophilus influenzae* represented the first step into the genomic era. Only twelve years later, there are now hundreds of Bacterial and Archaeal genomes publicly available. As this genome collection continues to grow, it presents an unparalleled opportunity to investigate the molecular basis of ecological adaptation through the use of computational analyses, combined with experimental investigation.

Most predicted genes in a newly sequenced organism encode proteins belonging to homologous families conserved in a number of organisms. However, there are also many families which display lower levels of conservation. In fact, a large number of families still contain just a single representative member, an orphan gene. As these genes are found in isolated lineages, it is plausible that they are responsible for niche specific traits.

It has been said that the number of orphan genes discovered in complete genome sequences is one of the biggest surprises of the genomic era (Doolittle, 2002). Prior to this discovery, it was generally accepted in the fields of biochemistry and genetics that science had succeeded in identifying most (approximately 80%) of the genes required for the normal life of a model organism, such as *E. coli* (Moxon & Higgins, 1997). The discovery of such unexpected genetic diversity has many implications, and interest in the subject is increasing. During this chapter, several explanations for the presence of orphan genes in bacterial genomes will be proposed, their biological significance will be described and the bioinformatics challenges that lie ahead if, as a community, we are to systematically study these poorly understood sequences will be discussed. In addition, the aims and objectives of this thesis will be introduced.

1.2 How many orphans are there?

It is important to quantify the scope of the orphan phenomenon before attempting to explain why the orphan genes exist. A useful way of estimating the current number of orphans is to determine the number of orphan genes in the complete bacterial and archaeal genomes. In terms of raw orphan numbers, the taxonomic uniqueness (how distant the closest complete genome is) of the genome being sampled, will be a key factor in the number of orphans found within a given genome. For example, if a genome from a new taxonomic division was sequenced, it would be expected that this genome would contain more orphan genes than a genome that was a member of a species that had already had several strains sequenced, presuming the genomes were of similar size and the species inhabited a similar ecological niche. To provide a measure of the taxonomic uniqueness of an organism, Fukuchi & Nishikawa (2004) introduced the 'Isolation Index of Organisms' (IIO). The index, based on the average of the logarithm of the best hit E-values collected over all the query sequences within a genome, was found to be proportionally related to the number of orphan genes in a genome (Fukuchi & Nishikawa, 2004). This relationship suggests that as more genomes are sequenced, the orphan number could plateau (Siew & Fischer, 2003a). Therefore the orphan genes could be the result of a lack of sequencing to a sufficient depth (Unger, Uliel & Havlin, 2003). It is known that selection of genomes for sequencing is highly biased (this situation is not unique to genome sequences, the American Type Culture Collection is similarly biased (Floyd *et al.*, 2005)), for example the over representation of pathogenic species (Wilson *et al.*, 2005).

A more recent study investigated the accumulation of bacterial orphan genes using the proteomes of the first 122 published bacterial species (Wilson *et al.*, 2005). The data was generated by comparison of each proteome to every other proteome using BLASTP (Altschul *et al.*, 1990) with a cut-off of 10^{-03}. The study found that the number of orphan bacterial genes was continuing to rise in a roughly linear fashion, despite the large number of genomes sequenced. After 122 proteomes of different bacterial species, the percentage of orphans as a total of predicted proteins was 12%. Of the 122 species, 7 species represented the only isolate from a division. These taxonomically unique species provided approximately 13% of the total orphans. This finding reflected the limited nature of our sampling of bacterial diversity (although projects now exist that aim to increase the diversity of the genome collection (Eisen & Fraser, 2003)), but also suggested that orphans were a widespread occurrence in bacterial taxa, with the exception of endosymbionts and intracellular parasites, both of which possess very small genomes.

Comparative analysis of eight pathogenic isolates of *Streptococcus agalactiae* found that even after eight genomes, each new strain continued to add new genes. Mathematical extrapolation of these results predicted that new genes will continue to be found even if hundreds of strains are sequenced (Tettelin *et al.*, 2005). This analysis led to the term 'pan-genome'. The pan-genome includes a core genome, containing genes present in all strains and hence defining the species, and a dispensable genome comprised of a halo of genes that may be absent in some strains and genes that are unique to a given strain.

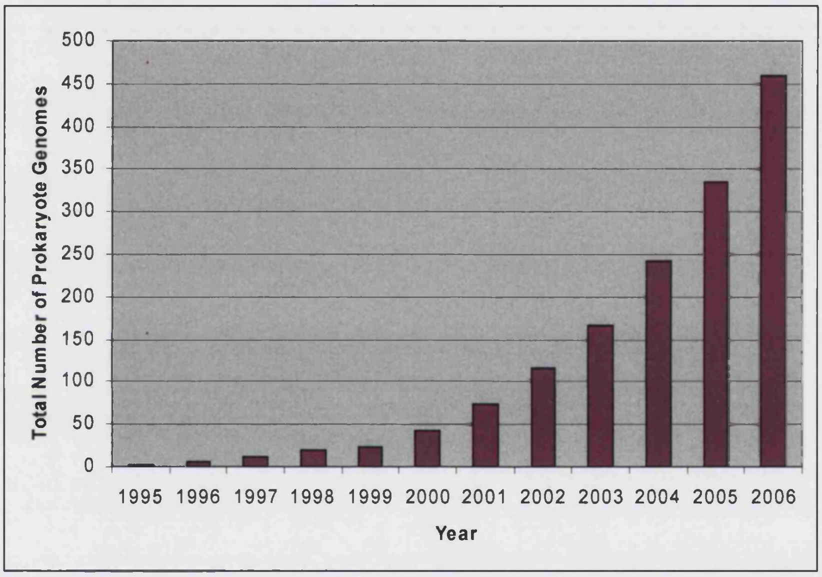
The structure of a species' pan-genome will depend on factors such as the environmental niche occupied, the level of genetic exchange and the population size (Holden, Rajandream & Bentley, 2005). In a study to investigate genes subject to positive selection in uropathogenic strains of *E. coli*, the size of the *E. coli* core genome was predicted to be 2865 genes. This is a relatively small total when it is estimated that each new *E. coli* genome will contribute 441 new genes (Chen *et al.*, 2006). Thus the dispensable genome in the case of the *E. coli* pan-genome is far larger than the core. It is thought that the dispensable genome may contain genes that are not essential for bacterial growth, but which confer selective advantages that may allow colonisation of a new niche (Medini *et al.*, 2005). The power of a species' pan-genome is indicated in *Vibrio cholerae*. Previously undetected toxin-like genes were discovered when a number of environmental isolates were analysed (Purdy *et al.*, 2005). These findings supported the discovery that environmental strains lacking the *ctxA* and *tcpA* genes (typically responsible for the pathogenicity of *V. cholerae*) were still capable of causing disease in mammalian models (Faruque *et al.*, 2004).

A similar method of referring to the different sections of the genome was proposed by Chiapello *et al.* (2005) in which they refer to a species backbone and strain specific loops. Investigation of the dispensable genes in *Stretococcus agalactiae* revealed that the majority were accounted for by hypothetical, phage and transposon related genes (Tettelin *et al.*, 2005). Analyses of 5 strains of *Streptococcus pyogenes* revealed similar patterns (Medini *et al*, 2005).

As the number of complete bacterial genomes continues to increase exponentially (see Figure 1.1), so to will the number of genes of unknown function. In addition to the complete bacterial genome collection, metagenomic techniques are also discovering vast numbers of previously unknown genes. For example, 1.2 million previously unknown genes were obtained from the Sargasso Sea (Venter *et al.*, 2004). More recently the results of the Sorcerer II Global Ocean Sampling Expedition (GOS) were released. This extensive dataset yielded 7.7 million sequencing reads with 6.12 million predicted proteins (Yooseph *et al.*, 2007). Of the assembled sequence, 85% was found to be unique using a sequence identity cut-off of 98%, indicating the great diversity within the dataset (Rusch *et al.*, 2007). When analysing the protein families, a linear trend in the discovery of new protein clusters was found. In addition, Yooseph *et al.* (2007) investigated the effect of the new dataset on orphan numbers. They obtained 84911 orphans from the NCBI-nr database and found that they were able to home 6044 of these orphans when compared to the GOS dataset. This implies that there are likely to be many more protein families remaining to be discovered (Yooseph *et al.*, 2007) and environmental sampling of this type will be able to place significant numbers of orphans into protein clusters.

Such studies indicate that the orphan gene phenomenon is not a self-solving puzzle. Instead, it is necessary to look at the issue more closely to try and determine the source of these genes. Several explanations have been suggested for the existence of orphan genes in microbial genomes. These will be discussed below, beginning with the possibility that they may not be genes at all.

Figure 1.1. Accumulation of complete archaeal and bacterial genome sequences. Data was obtained from GOLD v2.0 (Liolios *et al.*, 2006) and plotted by year.



1.3 Genes or Junk?

The first explanation for orphan genes is that they are not real protein coding genes. Instead they are random sequences of DNA that have been mis-annotated during the annotation process. Bacterial genome annotation has become, largely, an automated process. The most reliable method for identifying genes is through homology to a known gene. In the absence of such evidence, genes can only be identified *de novo* on the basis of structural features. These include the length of an open reading frame, the presence of a ribosome binding site in close proximity to the start codon and codon usage that is consistent with other genes in the genome (Pevsner, 2003).

Programmes such as Glimmer (Salzberg *et al.*, 1998) and GeneMark (Borodovsky *et al.*, 1995) are used to find genes in the raw DNA sequence. Such gene finding applications use a variety of Markov models to predict where the genes are located. For example, Glimmer identifies coding regions using Interpolated Markov Models (IMM) (Delcher *et al.*, 1999). The genome sequence is searched for all open reading frames (ORFs) above a threshold length. Glimmer then scores each of these ORFs using an IMM; if the score reaches a threshold value the sequence is judged to be a gene. Interpolated Markov Models are based on Markov chain models of the type used in programmes such as GeneMark.

The original GeneMark (more recent versions include GeneMarkS (Besemer, Lomsadze & Borodovsky, 2001) used a 5th-order model. A 5th-order model predicts a base by using the previous five bases. However, such a model can only perform accurately when there is sufficient training data, i.e., enough data to accurately estimate the probability of each base occurring after every possible combination of five preceding bases. Glimmer's IMM model overcomes this problem by only using oligomers for which sufficient training data is available, ranging from 1 to 8 bases in length. This works on the principle that in a typical microbial genome, some 5mers will occur infrequently and not provide reliable probability estimates, whilst some 8mers may occur frequently enough to give very reliable estimates (Salzberg *et al.*, 1998).

Whilst these systems work with a high level of accuracy, they are not perfect. The main issue is that of over-annotation. This is where the gene prediction programmes predict the presence of more coding regions than are actually present in the sequence. This results in a number of non-coding random DNA sequences being annotated as real genes. Such sequences will not find a match in sequence databases and so would be deemed, incorrectly, to be orphan genes. Data obtained from the Glimmer website (http://www.cbcb.umd.edu/software/glimmer/g3.table4.jun01.shtml) shows the accuracy of Glimmer3.0 in comparison with the NCBI RefSeq genome sequences. 30 microbial genomes were used in the analysis. The NCBI RefSeq annotations produce 84865 predicted coding regions; of these Glimmer3.0 predicts 81320 (95.82%). However in addition, it predicts 7938 coding regions not found in the RefSeq annotations. Some of these regions may be coding and therefore reflect errors in the RefSeq annotation, however many will be non-coding. It is also important to realise that many RefSeq annotations will be based on the output from gene prediction programmes and therefore, there may be inaccuracies in this data as well.

In gene prediction programmes, such as Glimmer3.0, there is a trade off between correctly identifying all coding regions but falsely predicting a number of extra genes, or reducing the number of extra genes predicted but in doing so increasing the risk of missing real genes. Additionally, with no formal annotation guidelines or procedure, different annotation groups may choose to use different length thresholds in their analyses, resulting in different levels of accuracy in different projects.

The NCBI and EBI provide sequence data to much of the biological community. I performed a comparison of the number of proteins predicted in the first 122 sequenced bacterial species. The data for this analysis was obtained in October 2005. Of the 122 genomes included in the comparison, only 7 were predicted to have the same number of proteins by both the NCBI and the EBI. Whilst these public resources are of massive value to the research world, it is clear that annotation errors persist in both these databases.

An example of annotation error can be seen in the genome of *Agrobacterium tumefaciens* C58. In this case, an identical strain (C58) was sequenced and annotated independently by two separate groups, Cereon (Goodner *et al.*, 2001) and Dupont (Wood *et al.*, 2001). The results of the two annotation efforts were published back to back in the same issue of the journal *Science*. In a perfect world these two sequences would be identical and neither would contain orphans. However this is not the case. *A. tumefaciens* C58 Cereon is predicted by the NCBI to contain 4554 proteins whilst *A. tumefaciens* C58 Dupont is predicted to contain 4661 proteins. The EBI echoes this discrepancy by predicting 4565 proteins in the Cereon sequence and 4662 proteins in the Dupont sequence. In addition, comparing the two proteomes resulted in over 100 orphans in each sequence. Performing a tBLASTn comparison (a similarity search of a DNA sequence database using a protein query) of these orphans against the DNA sequence of each genome, homes all orphans. Therefore the apparent differences in the sequences were down to discrepancies in the annotation.

The average size in amino acids of the orphans was much less than the average size of other genes within the *A. tumefaciens* C58 genomes. This size differential is echoed when the orphan gene phenomenon is looked at as a whole. The discriminatory power of methods such as codon usage becomes less reliable for shorter ORFs. This, coupled with the large number of short random ORFs, could potentially lead to an over prediction of short genes. For example, *E. coli K12* is believed to have approximately 4300 genes, but it is claimed by Skovgaard *et al.* (2001) that a more likely estimate would be in the region of 3800 genes.

The problem of over prediction is likely to be increasingly prevalent as the GC content of the organism increases. This is due to stop codons being AT rich, hence an increase in the likelihood of an ORF, by chance, reaching the threshold size acceptable as a gene. The length distribution of orphans and non-orphans has been described in several papers (Charlebois *et al.*, 2003, Siew & Fischer, 2003b, Skovgaard *et al.*, 2001) and has been used to suggest that the majority of orphans are annotation errors.

Annotation problems are amplified by the lack of standard protocols. Different significance, size or overlapping threshold can be applied and likewise the level of human supervision also varies between different genome annotation projects (Alimi *et al.*, 2000). Genomes are clearly annotated to different levels of quality. For example, of the first 150 bacterial genomes, 10% do not have their rRNA gene sequences annotated in their GenBank files (Ussery & Hallin, 2004). Further, once the initial annotation is completed, the predicted genes and their sequences are released into the public domain where any errors may potentially be perpetuated throughout the community. This process has been termed 'error percolation' (Gilks *et al.*, 2005).

Thus, bacterial genome annotation is not a trivial exercise and it seems unlikely that all regions annotated as coding for an expressed protein are in fact genuine genes. Novel annotation methods are being developed that may assist in the identification of real genes. Examples include 'genomic context' methods (Doerks *et al.*, 2004, Enault, Suhre & Claverie, 2005) and the systematic use of genomic data and scientific literature to associate genes to phenotypes (Korbel *et al.*, 2005). Genomic context methods predict functional associations between protein coding genes, such as physical interactions, co-membership in pathways or other cellular processes (Doerks *et al.,* 2004). Characterising protein function using this technique is not able to provide information about the exact function of a protein. A subsystems approach to genome annotation has been launched by FIG (Fellowship for Interpretation of Genomes) (Overbeek *et al.*, 2005). This approach involves experts in a particular subsystem (a generalisation of the term 'pathway') annotating that subsystem over the complete collection of genomes, rather than having an annotation expert attempting to annotate all genes in a single genome. One outcome of this method was the discovery that genes that appeared to be missing from a subsystem in a particular organism, were in fact found to be present. However, the relevant ORF had originally been missed by the gene prediction programme (Overbeek *et al.*, 2005).

Techniques to prioritise genes for further experimental characterisation are much needed and, in the future, with concerted community effort may help to improve the current annotation situation. Unfortunately at the present time there is no straight forward way to determine which of the predicted genes are real and which are not.

1.4 On the Brink of Extinction or a Long Lost Relative?

If an orphan gene is not a result of annotation error, how can they be explained using traditional evolutionary theory? One possibility is that the orphan gene is the last remaining member of an otherwise extinct gene family. Alternatively, the orphan gene could be a lost member of a known gene family that has diverged beyond recognition.

Firstly I shall look at the possibility that an orphan gene represents a gene family on the brink of extinction, due to gene loss and genome degradation. It has been claimed that lineage-specific gene loss accounts for the majority of the differences in gene repertoires between genomes (Krylov *et al.,* 2003). Gene loss is particularly common when bacterial lineages make the transition from a free-living or facultative parasitic life cycle to permanent associations with hosts (Moran *et al.*, 2002). Such gene loss has been seen in many species, such as the *Mycoplasma*, *Rickettsia*, *Buchnera aphidicola* and *Borrelia burgdorferi.* Some genes that are lost from reduced genomes are those that are no longer required. Elimination of unnecessary pathways explains a large proportion of gene losses. For example, many genes involved in energy metabolism have been eliminated from *Mycoplasma* species and *Rickettsia* species (Moran *et al.*, 2002). However, it is also found that discarded genes encode products that seem as useful in an obligate pathogen as they would in a free-living organism. Such gene loss could be attributed to genetic drift and the fixation of mutations that inactivate potentially useful, though not essential, genes (Moran *et al.*, 2002). An analysis by Snel, Bork & Huynen (2002) investigated the evolution of archaeal and proteobacterial gene content. They determined that gene loss was quantitatively the most dominant process in shaping the genome.

If this is the case, it is possible that divergence from a common ancestor could lead to an orphan in one genome and a pseudogene in another. Pseudogenes have been rendered non-functional due to frameshifts or premature (in-frame) stop codons that act to truncate full length proteins. In eukaryotes, surveys have indicated that pseudogene formation is more likely in younger, more taxonomically restricted protein families, often linked to the generation of functional diversity (Harrison & Gerstein, 2002).

Historically, prokaryotic genomes have been perceived to be lacking in pseudogenes due to the small genome size and the influx of genetic elements such as bacteriophage (Lawrence, Hendrix & Casjens, 2001). This influx results in high deletion rates in most bacteria thus maintaining the compact genome size and paucity of pseudogenes. Exceptions to this are intracellular parasites such as *Mycobacterium leprae* (Cole *et al.*, 2001) whose sheltered lifestyle removes them from the danger of insertion elements and phage. Therefore, they have a lower deletion rate and higher pseudogene load (Lawrence *et al.*, 2001).

However, this view has been challenged (Liu *et al.*, 2004, Lerat & Ochman, 2004, 2005). An analysis of 64 prokaryotic species resulted in the identification of 6895 candidate pseudogenes. Of these pseudogenes, approximately 2300 overlapped annotated hypothetical genes (Liu *et al.*, 2004). These results, once again, indicate erroneous gene annotations or sequencing errors in bacterial genomes. Work on *E. coli* MG1655, *E. coli* O157:H7, *E. coli* CFT073 and *S. flexneri* 2a identified 98, 142, 98 and 168 new pseudogenes, respectively (Lerat & Ochman, 2004). The genome of *Buchnera aphidicola*, the symbiont of *Acyrthosiphon pisum*, contains four genes that share no sequence similarity to its closest free living relatives. Further analyses led to the conclusion that these unique genes possess traits commonly found in pseudogenes (Mira, Klasson & Andersson, 2002). More recently, a study into the genomes of human pathogens and their close relatives found that all contained substantial numbers of pseudogenes. The data suggested that pseudogenes appear to be more common in the genomes of recent pathogens than in free living or benign relatives (Lerat & Ochman, 2005). The reason for this could be that previously useful genes are rendered useless when relying on nutrients from the host. These superfluous genes are knocked out to become pseudogenes. Another reason could be the reduction in population size on host infection. This would relieve selective pressure and result in an increase in deleterious mutations.

In prokaryotic organisms, pseudogenes are believed to arise from three processes. The first of these is the disablement of a native duplication. Secondly, it could be the result of the decay of a native single copy gene. Finally, it is possible that pseudogenes are a result of failed horizontal transfer events (Liu *et al.*, 2004). It is possible that the decay of a single copy gene to form a pseudogene in one genome could have the effect of creating, what appears to be, an orphan in another genome. The relationship between horizontal transfer and orphan genes will be discussed in detail below.

An analysis of orphans in *Rickettsia conorii,* found that the majority were short remnants of longer genes, present in the ancestor of the modern *Rickettsia* species (Amiri, Davids & Andersson, 2003). The ancestral species gene sequences were reconstructed using data from *R. typhi* and *R. prowazekii* (both members of the typhus group (TG)), and also *R. montana* and *R. rickettsii* (both members of the spotted fever group (SFG)). It was found that members of TG and SFG were both moving towards a similar gene set but at different rates. Therefore, proposed orphans in the SFG corresponded with pseudogenes in the TG, and pseudogenes in the SFG corresponded with extensively degraded gene remnants in the TG (Amiri *et al.*, 2003).

In effect, fragments of genes are retained temporarily and have the appearance of multiple short ORFs. These short ORFs will possess nucleotide composition patterns similar to those of the full length ancestral sequences from which they were derived. However, they no longer code for functional proteins (Amiri *et al.*, 2003). As more genomes are sequenced, the sequences of many closely related organisms will become available. Analysis of these genomes should provide more pseudogenes to compare orphan genes against whilst highlighting errors in the original genome annotations.

An alternative explanation for the presence of orphan genes is that they are members of known gene families that have diverged beyond recognition. In other words, the relationship may have faded to such an extent that our current sequence analysis tools do not possess the statistical power and recognition capabilities to detect it. In such cases, structural studies may be able to shine a light on these distant relationships and allow us to home some of the orphan genes. If orphans are distant members of known protein families, they will have similar functions and hence similar three-dimensional structures, even if the protein sequences have diverged beyond recognition. A study of the 3D structures of orphans found within the PDB (Protein Data Bank) (Berman *et al.*, 2000), identified that the majority of the orphans do possess previously observed folds (Siew & Fischer, 2004). This suggests that the orphans may correspond to distant members of known protein families. Further work has been performed on a family of sequences specific to *Bacillus.* Using methods such as fold recognition, it was possible to identify an α/β hydrolase fold and hypothesise that the orphans may belong to the haloperoxidase family (Siew, Saini & Fischer, 2005).

Several factors responsible for controlling the rates of protein evolution have been suggested, for example gene dispensability (Hirsh & Fraser, 2001, Yang, Gu & Li, 2003), recombination rate (Pal, Papp & Hurst, 2001) and levels of gene expression

(Pal, Papp & Hurst, 2003). In the case of the latter, highly expressed genes are expected to evolve more slowly. Since orphan genes are likely to encode an accessory function, it is speculated that they would be expressed at low levels. As such, orphan genes are candidates for rapid sequence divergence.

The pace of sequence divergence has been tested in *Drosophila* species. Sequencing of narrowly restricted genes shared by *Drosophila* species shows that orphan genes evolve, on average, significantly faster than non-orphan genes (Domazet-Loso & Tautz, 2003). Cai *et al.* (2006) investigated the divergence rates of genes with different degrees of lineage-specificity in the Ascomycota fungi. The results of this analysis also indicate that genes with greater lineage-specificity had accelerated evolutionary rates. This may reflect the influence of selection and adaptive divergence during the emergence of orphan genes (Cai *et al.*, 2006). However, other data from the Domazet-Loso & Tautz (2003) analysis showed that some orphan sequences can have very low divergence rates. Additionally, the processes described may only be applicable in eukaryote species and not be transferable to bacteria.

An alternative evolutionary mechanism has been suggested that could explain some new gene families. This mechanism involves changes in the frames of translation. Research in this area suggested a frame-shifted evolutionary relationship between several hundred domain families (Pellegrini & Yeates, 1999). Whilst this study was focussed on relatively common protein sequence families, there is no reason why an investigation into the orphan genes may not provide similar results.

To solve the problem of homing orphan genes within the correct gene families, new techniques need to be developed, for example using functional domain composition to predict protein function (Cai & Doig, 2004). Methods are required that make use of alternative patterns and mine metadata within the sequence data, in doing so going beyond traditional approaches.

1.5 The New Gene Generators

In the section above, the process by which orphans could be generated through gene duplication and subsequent extreme diversification was described. Horizontal gene transfer provides a mechanism for bacterial isolates to obtain sequences (and traits) from both related and unrelated organisms. As these genes have already been refined by natural selection, the benefit to the organism could be instantaneous (Daubin & Ochman, 2004b). Such benefits have been seen in numerous bacterial lineages, the best publicised is perhaps that of antibiotic resistance, for example in *Salmonella enterica* (Carattoli *et al.*, 2002).

There are several methods by which bacteria can obtain new genes, examples of which include: (i) transformation, which involves genetic material being taken up from the environment, (ii) conjugal transfer between bacterial species and (iii) transduction, when DNA is delivered by a virus i.e. gene insertions by phage (Medini *et al.*, 2005).

The arrival of complete bacterial genome sequences revealed for the first time the importance of the phage-bacterium interaction. It was shown that, in certain bacteria, a substantial amount of bacterial DNA was of phage origin (Casjens *et al.*, 2003). Such data has contributed to a shift in our understanding, from a straight forward host-parasite relationship to a co-evolution of bacterial and viral genomes (Canchaya, Fournous & Brussow, 2004).

A comparative analysis of 18 phage genomes from *Pseudomonas aeruginosa* revealed a high percentage of novel genes (55% were restricted to the phage they were found in), suggesting that phage store a vast reservoir of genetic diversity (Kwan *et al.*, 2006). In another study, 10 mycobacteriophage genomes were sequenced and compared to each other and to 4 previously sequenced mycobacteriophage genomes (Pedulla *et al.*, 2003). A total of 1659 predicted coding regions were identified in the 14 genomes, remarkably in the region of 50% of these were unique when queried against current databases. Of the remaining 50%, three quarters only found matches in other mycobacteriophage genomes (Pedulla *et al*, 2003). The authors suggest that, if the data obtained accurately reflects the bacteriophage population, "bacteriophages perhaps represent the biggest unexplored reservoir of sequence information in the biosphere". This claim is supported by the data obtained from the Global Ocean Sampling Expedition, in which a higher than expected proportion of sequences were of viral origin (Yooseph *et al.*, 2007), reflecting the poor sampling of viral diversity.

Analyses suggest that horizontal gene transfer from phage may be responsible for contributing large numbers of orphan genes to bacterial genomes (Ohnishi, Kurokawa & Hayashi, 2001, Beres *et al.*, 2002, Deng *et al.*, 2002, Smoot *et al.*, 2002 and Hsiao *et al.* 2005). In *Pseudomonas aeruginosa*, genes encoding the tail of two different bacteriophages (P2 phage and lambda phage) have been converted to form bacteriocins (R-type and F-type). These can be used by the bacteria to kill its competitors (Nakayama *et al.*, 2000). These regions within the *P. aeruginosa* genome are found to contain several orphan genes and those that are not orphans are found to have a highly restricted bacterial distribution.

The lack of homology could be the result of the poor sampling of phage genomes. Orphans are significantly shorter than native genes and are A+T rich when contrasted with the rest of the genome (Daubin & Ochman, 2004a). Phage also encode short A+T rich genes (Pedulla *et al.*, 2003), and on average phage are 4% richer in AT than their hosts (Rocha & Danchin, 2002). The dinucleotide frequencies of *E. coli* orphans and of phage known to infect *E. coli* were found to be similarly biased in contrast with the native genes (Daubin & Ochman, 2004a). Research investigating proposed orphans in *E. coli* found that 54% of the orphans and the HOPs (genes with a heterogeneous occurrence in prokaryotes) are found in clusters of two or more genes. In addition, many of the clusters were in the vicinity of regions associated with lateral gene transfer, such as IS elements and prophages (Daubin & Ochman, 2004a).

However, questions remain. Why do phage provide bacterial species with useful genes? One theory is that by providing useful genes, the inevitable parasite host conflict can be avoided, in favour of a mutually beneficial symbiosis, in which the phage and bacterium can co-exist (Daubin & Ochman, 2004b). As Daubin & Ochman (2004b) wrote "one might view bacteriophages as start up entities whose existence is based on creating an innovation that has been overlooked by other organisms".

It is also possible that ORFs transferred in by phage may be non-coding or, alternatively, of no functional use to the bacterial host. Many horizontally acquired genes are likely to cause deleterious effects in the bacterial recipient; therefore these bacteria will be lost from the population (Thomas & Nielsen, 2005). It has been estimated that there are 10^{31} bacteriophage on Earth which infect 10^{24} bacteria per second; it is therefore easy to imagine a constant flow of genetic material (Tettelin *et al.*, 2005). In order to maintain an effective genome size in such conditions, the bacterial population must be able to remove the unwanted sequence from its gene pool. Therefore, our genome sequences could be considered as a snapshot (Daubin, Lerat & Perriere, 2003) of a constantly changing environment.

The hypothesis that phage are responsible for many of the bacterial orphan genes has been questioned by Yin & Fischer (2006). Using an analysis of orphans and non-orphans from 277 microbial genomes, searched against the public viral protein database, they showed only 2.8% of the orphans had viral homologues compared with 7.9% of the non-orphans, suggesting the evidence for the viral origin of orphans is weak (Yin & Fischer, 2006). It is also worth considering that whilst orphan genes and phage genes generally have higher AT content than the host chromosome, this is equally true of intergenic regions of bacterial genomes (Binnewies *et al.*, 2006).

As mentioned previously, horizontal transfer can take various forms. Plasmids could also be a source of orphan genes in bacteria. One example of a plasmid integrating with a bacterial chromosome is found in the *Methanopyrus kandleri* AV19 genome (Jensen *et al.,* 2003). Two large regions within the bacterial chromosome were found to have an AT content significantly different to the rest of the genome. Further investigation led to the conclusion that the regions were formed from the integration of two plasmids into the chromosome. The two regions being investigated were also found to contain a large number of orphan genes. Examples of transfer between different bacterial species are also common. For example, species such as *Thermotoga maritime* and *Aquifex aeolicus* have a substantial number of genes showing greatest similarity to those found in the archaea (Ochman, Lerat & Daubin, 2005). As these are between known bacterial species, they would not be viewed as orphans, but they may still be of interest for their role in niche adaptation.

Another way in which new genes could occur is through the process of de-novo gene creation. De-novo gene formation refers to the idea of non-coding sequence undergoing a change that leads to it coding for something. It can then evolve into a gene. There is very little in the scientific literature discussing this possibility.

1.6 Proof of Function

Despite errors and incomplete sampling, it appears that at least a proportion of the orphans are real. Phylogenetic analysis can be used to test whether taxonomically restricted genes appear to be functional. An analysis of genes restricted to γ-Proteobacterial clades indicated that the majority of the genes were functional proteins (Daubin & Ochman, 2004a). The analysis was performed using the K_a/K_s ratio. In addition to predicting that the genes were functional, it was found that the characteristics of genes restricted in the deeper clades (i.e., those that had been in the lineage longer), were approaching those of the native genes (in terms of base composition and evolutionary rates). In contrast, the younger genes tended to be clustered and adjacent to horizontally transferred regions (Daubin & Ochman, 2004a). Ochman (2002) also utilised the K_a/K_s ratio to predict that the majority of putative genes, including those that are deemed as being short, are genuine protein coding regions. However, the majority of ORFs that appeared likely to be mis-annotations were short and of unknown function. It has since been claimed that the method used by Ochman could exclude legitimate annotations, such as leader peptides, in which only a small number of amino acids in the sequence are under selection (Lawrence, 2003). Furthermore, it has been suggested that, due to the use of arbitrary length thresholds for determining when an ORF becomes a predicted coding region, some small genes are not being annotated (Harrison *et al.*, 2003). This is judged to be a manageably low number.

Increasingly, a range of experimental methods are also providing evidence for real orphans. One example validation of the pathogen-defining potential of orphan genes and their relationship with phage is found in *Vibrio cholerae.* The genome sequence of *V. cholerae* Tor N16961 revealed a single copy of the cholera toxin (CT) genes, *ctxAB* (Heidelberg *et al.*, 2000). These genes are localised within the integrated genome of CTX$^{\Phi}$, a temperate filamentous phage (Waldor & Mekalanos, 1996). The receptor for the entry of the CTX$^{\Phi}$ phage into the bacterial cell is thought to be the toxin-coregulated pili (TCP). The TCP represent the critical intestinal colonisation factor of *V. cholerae* (Manning, 1997), allowing the cells to clump together and stick to the intestinal walls.

The genes involved in assembly of TCP are part of a pathogenicity island that includes genes sharing homology with bacteriophage proteins (Heidelberg *et al.*, 2000). The majority of the genes located in the TCP cluster were classed as orphans until the sequence of *V. fischeri* (Ruby *et al.*, 2005), a symbiotic bacterium of squid, was completed and orthologs were found. This surprising finding is made more intriguing by the suggestion that this region is native to *V. fisheri* but was acquired recently by *V. cholerae,* perhaps through phage mediated transfer. In addition, the *ctxAB* genes closest homologue is found in a pathogenic strain of *E. coli* and is therefore an example of a gene with heterogeneous occurrence in prokaryotes (HOP).

This example illustrates several points. Firstly, it shows that genes classed as orphans may be encoding proteins responsible for important biological phenotypes that play a major role in the lifestyle of an organism. Secondly, it indicates how gene delivery via a phage may be an important method for transferring these dispensable genes to different organisms. Finally, by sequencing more closely related genomes it will be possible to gain greater understanding of the evolution of these organisms and the factors that lead to their differentiation, through the homing of taxonomically restricted but phenotypically relevant genes into gene families (Field, Feil & Wilson, 2005a).

In contrast to the example above of an unusual 'orphaned locus' with an obvious and critically important phenotype, there has been little in the way of experimental characterisation of orphans of unknown function. Such work is expensive and time consuming. An exception to this is the work of Alimi *et al.* (2000), who have provided reproducible evidence of transcription in 19 proposed orphan genes (25 orphan genes were conservatively selected for the experiment) from the *E. coli* K12 MG1655 genome. This high rate of mRNA detection suggests that a large majority of predicted genes of unknown function are of biological relevance. In the study, it was found that 86% of the predicted 4290 *E. coli* genes exhibit detectable mRNA levels. Of the 4290 predicted genes, 1352 were classified as hypothetical. mRNA was detected for 80% of these hypothetical genes. As previously stated 19 of the 25 strictly orphan genes (76%) expressed mRNA. Hence, hypothetical genes, both orphan and conserved, do not appear to be significantly less likely to be transcribed than known annotated genes (Alimi *et al.*, 2000).

However, obtaining transcribed RNA does not confirm that the gene codes for a functional protein (Amiri *et al.*, 2003). This has been demonstrated by Taoka *et al* (2004), who found that horizontally transferred genes on the chromosome of *E. coli* rarely produced a protein product, despite the majority appearing to be transcribed to RNAs as efficiently as the native bacterial genes. Hence, confirmation of the expression of orphans remains speculative until evidence of protein products is given.

In a second project, genes unique to the halophilic archaea, *Halobacterium sp. NRC-1*, were investigated by RT-PCR (Shmuely *et al.*, 2004). 39 novel predicted genes were used in the analysis, each of which had at least one homologue within the genome but no detectable homologues in other organisms. The 39 predicted genes represented 14 paralogous families. RT-PCR identified mRNA from 30 of the 39 predicted genes, corresponding to members of 13 of the 14 paralogous families. Of the 9 targets which failed to yield evidence for expression, only 2 corresponded to proteins shorter than 150 amino acids. Therefore, in this analysis, there was no indication that shorter predicted genes are less likely to be expressed (Shmuely, *et al.*, 2004). However, as in the work of Amiri *et al.*, (2003), further work is required to determine evidence of a protein product. Preliminary work from computational analyses, such as fold recognition methods, suggested that 8 of the 14 paralogous families may correspond to distant members of known families (Shmuely *et al.*, 2004). Similar work has been performed on hypothetical genes in *Haemophilus influenzae* (Kolker *et al.*, 2004) and *Shewanella oneidensis* (Kolker *et al.*, 2005, Elias *et al.*, 2006). In both studies, mRNA expression was found for the majority of these genes.

Genomotyping (whole genome comparisons of microbes using microarrays) has been used on several bacterial species. Such studies provide a method for identifying genes associated with particular phenotypes, such as virulence. These candidates could include lineage-specific and orphan genes. Examples of genomotyping experiments include work on the pathogen *Campylobacter jejuni* (Champion *et al.*, 2005) and *Neisseria gonorrhoeae* (Snyder, Davies & Saunders, 2004). Another genomotyping study was performed on 15 *Helicobacter pylori* clinical isolates. It was found that 22% of the *H. pylori* genes are dispensable in one or more strains (Salama *et al.*, 2000). This number is expected to be an underestimate, as the array could only contain genes present in one of the sequenced strains. Distinct patterns of strain-specific gene distribution along the chromosome were found, this may be explained by mechanisms of gene acquisition and gene loss. In addition, candidate virulence genes from the strain-specific genes were identified and can now undergo further characterisation experimentally (Salama *et al.*, 2000).

1.7 <u>Do Orphans have a Future?</u>

In the Roberts report (2004) for the American Academy of Microbiology, the need for a prioritised list of genes of unknown function was highlighted. The list should include all uncharacterised species- and strain-level taxonomically restricted genes. The need for such a list has been elevated by the recent recognition of the pan-genome concept and the realisation that genetic diversity has been vastly underestimated. As an increasing number of metagenomic projects report back their findings, it is becoming clear that we are still far from discovering all protein families in nature. A list of the top 10 conserved hypothetical genes was created in an attempt to encourage the experimental characterisation of these genes (Galperin & Koonin, 2004). The list was based on numerous criteria, the primary being phyletic spread, and illustrated how the bioinformatics community could interact with experimentalists to systematically tackle key issues in genomics. Lists of orphan genes have been produced previously; examples include the Orfanage (Siew, Azaria & Fischer, 2004) and CUPID (Mazumder *et al.*, 2005). Both these examples are online databases that enable the user to generate lists of taxonomically restricted genes. However, in order for such resources to keep track of both changes in annotation and new genome sequences, substantial investment in time and capital is required.

The Roberts report also influenced the development of the Gene Trek in Prokaryote Space (GTPS) project (Kosuge *et al.*, 2006), which aims to assign a degree of reliability to all predicted protein-coding genes in bacterial and archaeal genomes held by the INSDC (International Nucleotide Sequence Database Collaboration). Predicted coding regions are graded for quality according to a number of analyses, including BLAST and InterProScan results. Potential genes range in their grades from AAAA1-D3 (five main categories including orphans), thus providing the user with a means to estimate the quality of a potential coding region and prioritise candidates for further investigation.

More recently, a community call similar to that made by Roberts has been issued by Karp (2004). This proposal focussed on the inverse problem, i.e., functions with no associated sequence. Such proteins have been termed 'orphan enzymes'. An example is shown in *Prochlorococcus marinus* CCMP1378 (MED4) in which there is no recognisable gene sequence for carbonic anhydrase (Fuhrman, 2003). If these proposals were followed, it is likely the work would have significant overlaps, with many of the hypothetical protein sequences being responsible for many of the orphan enzymes.

The ability to distinguish real and artefactual annotations would have several positive outcomes. It would improve the quality of genomic annotations, provide lists of candidate genes for further analysis and answer fundamental questions about the coding capacity of different organisms. For example, genomic islands are clusters of genes in genomes that show evidence of horizontal origins. A study by Hsiao *et al.* (2005) found that, not only do genomic islands contain a disproportionate number of genes of medical, agricultural and environmental importance, but they also contain higher proportions of orphan genes. This suggests that microbes have a larger 'arsenal' of novel genes for niche adaptation than previously anticipated. However, an alternative explanation for this result is the fact that genes in genomic island regions are more likely to be predicted incorrectly by gene prediction software. This is because of the difference in composition properties, such as codon usage between genomic islands and the host chromosome (Hsiao *et al.*, 2005). An index that can be used to accurately characterise the orphaned fraction of complete genomes would be of great use to the community (Wilson *et al.*, 2007).

Galperin & Kolker (2006) recently called for new approaches to deal with the vast diversity of data that projects are uncovering. Indeed, it is clear that novel methods will need to be applied if the orphan problem is to be solved. Such methods could include the improved use of structure (placing orphans in folds) and information on protein interactions. Such data is currently scarce in comparison with the volume of sequence data, thus inference from both direct interactions (physical binding between proteins) and indirect associations (e.g. shared pathway membership) is lacking. Fortunately, headway is being forged in this area by the database STRING (von Mering *et al.*, 2006), which aims to collect, predict and unify both direct and indirect protein-protein interactions.

It is also clear that mechanisms must be put into place to systematically remove errors from current genome annotations. As part of this, it is important that any such resource is driven by the research community through direct contributions. Examples of successful community action include work in the structural genomics initiative (Stevens, 2004). The number of structures, including many hypothetical proteins, solved within this initiative is already in the hundreds (Galperin & Koonin, 2004). Such a resource dedicated to taxonomically restricted genes in prokaryotes could offer an important step forward in the research community's attempts to explore these unique predicted genes.

1.8 Aims and Objectives

I intend to utilise computational methods to investigate and contribute significantly to the analysis of orphan genes. To perform such analyses, it is necessary to design and develop suitable software. Firstly, software responsible for the analysis of the genomic data will be required. Due to the volume of data available, the software must be able to analyse the genomic sequences and produce output in a human readable format. In addition, the completed software should be made available to the research community for further use. Secondly, a database is required to store the data generated in the analyses. The database should have an intuitive interface that allows members of the research community to interact with it and access the data stored within it. It should also allow users to download the data in a standard format so that it can be integrated with data obtained from other projects, hence allowing informed research to proceed efficiently. Developing these tools will form a significant proportion of the work required to produce this thesis.

Initial research will seek to determine whether the number of orphan genes in our complete bacterial genome collection is still rising or whether the number has reached a plateau, as previously predicted. The bacterial genome collection is an important resource for scientists working in microbiology. It is therefore important to understand the nature of the genomes that comprise the collection with particular emphasis on determining the biases present. Such biases will have implications on the analysis of genomic data and, therefore, it is important that these are understood.

As described, there are many different explanations for the presence of orphan genes in prokaryotic genomes. Many of these ideas appear contradictory, hence it is important to realise that no single explanation can account for all the orphan genes independently, but each explanation might be responsible for a percentage of the orphans. Each individual orphan will need to be investigated in order to determine what it represents, for example, is it an annotation error, a pseudogene, a member of an existing family or could it be the result of a horizontal transfer event? Such an analysis is not possible due to the volume of data being produced and the economic and time costs associated. Therefore, it is necessary to prioritise the orphans for further characterisation.

Of the different explanations for the existence of orphans, that of errors in annotation is of the most immediate significance. Whilst not necessarily of interest from a biological perspective, it is a major issue that has limited exposure. This is largely due to the excitement of the possibilities opened up by the influx of genomic data. However, this excitement could become significantly diminished if annotation errors are found to be prevalent and responsible for the majority of the orphans. It is important to determine which of the orphan genes are most likely to be real coding genes and which are likely to be a result of errors in the annotation process. I aim to investigate the possibility of annotation errors and develop a method for ranking orphans according to their 'quality'. The high quality sequences are those most likely to be coding, the low quality are those most likely to be errors. By obtaining expression data from public microarray resources, it will be possible to provide support for the ranking method. A successful ranking system should result in research focussing on the orphans of high quality.

Many genes are restricted in their distribution to a particular taxonomic group and hence can be termed as lineage-specific. Orphans found in taxonomically isolated genomes, may not be species or strain-specific, but instead could be division or family-specific, appearing as orphans due to sampling bias. It would be of use to the wider community to determine which of the orphan genes in isolated genomes are more likely to be found in other species and which are most likely to be unique to a given species. Experimental work to determine the functionality of a gene could be focussed on the genes that are likely to be found in numerous genomes.

It is hoped that the methods developed and the results reported in this thesis will further our understanding of bacterial orphan genes and provide a platform for future analyses to take place.

CHAPTER 2

Orphans as Taxonomically Restricted and Ecologically Important Genes

Gareth A. Wilson, Nicolas Bertrand, Yatin Patel, Jennifer B. Hughes,
Edward J. Feil and Dawn Field
(2005)
Microbiology 151: 2499-2501.

2.1 Overview

The abundance of orphan genes, or genes without known homologues, is amongst the greatest surprises uncovered by the sequencing of a large number of eukaryotic and bacterial genomes. It is therefore important to determine how the number of orphan genes will change as we sample more genomes. There are three possibilities. Firstly, the number of orphans could continue to rise as we sample new genomes. Alternatively, orphan numbers could plateau, despite the sampling of novel taxa, as has been suggested in the past (Siew & Fischer, 2003a). Finally, the number could decrease by improving our annotation methods and the sensitivity of our similarity searching algorithms, thereby finding homes (gene families) for current orphans (Skovgaard *et al.*, 2001).

Here we examine these possibilities using data generated for a set of 122 bacterial species for which we have complete genomes. We use this data to show that orphans are continuing to increase in number, emphasise further the importance of sequencing taxonomically diverse isolates (especially from environmental samples) and suggest that we now classify these predicted proteins as "taxonomically restricted genes" (TRGs), as this concept seems more useful for advancing our knowledge of these sequences and their potential ecological significance.

2.2 Numbers of Orphan Genes in Bacterial Genomes

We examined the accumulation of bacterial orphans using the proteomes of the first 122 published bacterial species (Figure 2.1A). The decline in orphans, over genomes sequenced, as a percentage of total predicted proteins in these proteomes (Figure 2.1B) was also examined. Datasets 'D1' and 'D2' were taken from the OrphanMINE database (www.genomics.ceh.ac.uk/orphan_mine). In order for these analyses to test the hypothesis that orphan number would plateau at 26,000 (Siew & Fischer, 2003a), we defined orphans in the following way, based on the methodology of Siew & Fischer (2003a). The datasets were generated by comparison of each proteome to every other proteome using BLASTP with a cut-off of 10^{-3}. D2 was generated by removing all predicted proteins smaller than 150 amino acids in length or containing any regions of low complexity (>0% calculated by SEG using default settings (Wooton & Federhen, 1993)), from D1. Genomes were added to the analysis in the order in which their sequence was published. These orphans are predicted genes found in only one genome in this set of bacterial genomes and are only orphans with respect to this dataset (a small proportion of these genes do have matches in phage and plasmids and among bacteria without complete genome sequences). Figure 2.1A shows that the number of these orphan bacterial genes is continuing to rise in a roughly linear fashion despite the large number of genomes sequenced, and this trend shows no sign of levelling off. In fact, the last 30 species included in this study, provided 30% of the total orphans in our study (mean=441 ±643 for dataset D1, despite the large standard deviation all species contributed orphans).

With the availability of relatively few bacterial genome sequences, the addition of new species removed a large percentage of orphans (Siew & Fischer, 2003b). However, as new species are added, the fall in the percentage of orphans slows and each new genome contributes very little to the decrease in orphans. In dataset D1, the percentage of orphans fell from 100% to 30% after the inclusion of the first 10 bacterial species, however after 55 species the percentage is only down to 15%, and the percentage drops only 3% further to a value of 12% after 122 species.

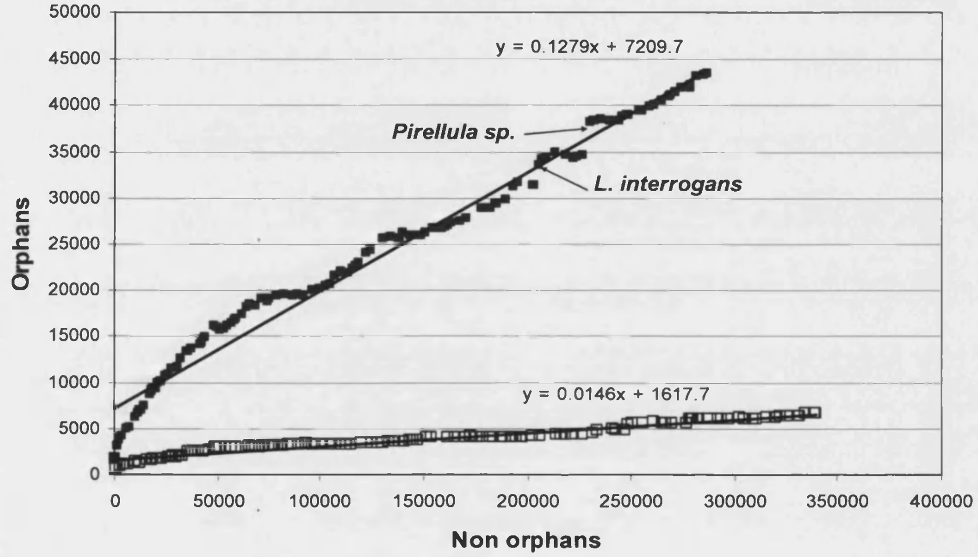
Trend lines were fitted and used to predict orphan gene levels after the sampling of 200 species. For the more conservative dataset D2, the percentage of orphans after the inclusion of 122 species was 1.89% (6696 of 355079 ORFs) and after 200 species, 1.16% (6751 of 582,000 ORFs). Therefore, although the percentage of orphans is falling, the actual number of orphans continues to rise, albeit very slowly. A similar pattern can be seen for D1 where 10% of all predicted coding regions in 200 species are predicted to be orphans. This is a far more significant percentage, but it is possible that this larger dataset contains genes which represent annotation artefacts (Skovgaard *et al.*, 2001). However, it has also been recently shown that A+T rich, short proteins, which look like mis-annotated junk, may actually be derived from phage genomes by horizontal gene transfer (Daubin & Ochman, 2004a).

These trends reveal several interesting points. First, given our current dataset for bacteria, it is not possible to make an estimate of the maximum number of orphans, as orphan growth does not show evidence of reaching a plateau. This conclusion is also supported by examining the rate at which new protein families are discovered (Kunin *et al.*, 2003).

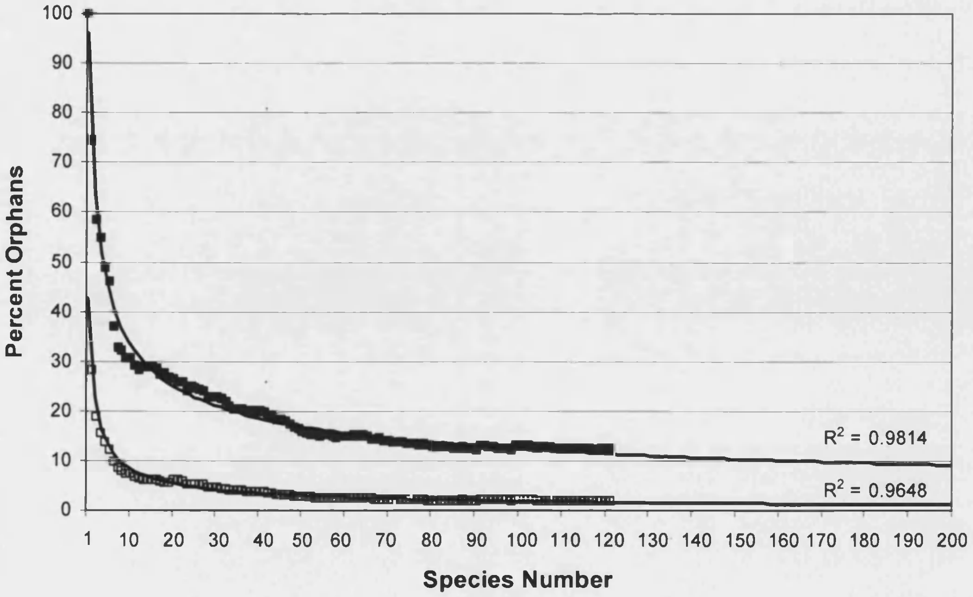
Second, it appears that improved taxonomic sampling of distantly related genomes is continuing to reveal large numbers of orphans. These data suggest that the number of bacterial orphan genes will continue to increase for the foreseeable future, as long as

Figure 2.1 A & B. The accumulation of bacterial orphans. For this analysis, data on the number of orphans in complete bacterial genomes was taken from the 'OrphanMine' database (www.genomics.ceh.ac.uk/orphan_mine). The dataset D1 represents all the orphans found in the bacterial genomes using BLASTP similarity searches and a cutoff threshold of 10^{-03} (corresponds to dataset D3 in database). In addition we created a more conservative dataset (D2) in which all predicted proteins smaller than 150 amino acids in length containing any regions of low complexity were removed (corresponds to dataset D4 in database). **A.** A plot of the cumulative number of orphans versus non-orphans. The number of orphans in datasets D1 (■) and D2 (□) are plotted showing that the number of orphans is continuing to rise in a linear fashion. Each data point represents the addition of a complete genome sequence in chronological order of publication (N=122 species). The two species contributing the largest numbers of orphans are shown. **B.** The decline in the number of orphans in datasets D1 (■) and D2 (□) as a percentage of all predicted proteins. A power curve was fitted and the R^2 value is shown. An extrapolation of this curve is used to predict the percentage orphans after 200 species have been sequenced (shown by the solid black line).

A.



B.



we continue to assay novel branches of the microbial tree of life. Therefore, although improved taxonomic sampling is reducing the overall percentage of orphans, it cannot be used to assign all orphans to known gene families. Further, it is also likely that orphans will continue to be found in lineages that have already been heavily sampled (Hayashi *et al.*, 2001, Perna *et al.*, 2001).

Third, the number of currently known genes is undoubtedly a small proportion of the number of genes yet to be found as we sample more taxonomically and ecologically diverse species. It is well known that our selection of genomes for sequencing is highly biased. For example, nearly half of the species in this dataset are pathogens, and 76 of the 122 species examined here are from only two divisions, Proteobacteria and Firmicutes. Of these 122 species, 7 represent the only isolate from a division. These taxonomically unique species contribute approximately 13% of the total orphans in our dataset. It is therefore expected that our current databases are a significant underestimate of the number of new genes that might be sequenced in the future. Fortunately, there are now projects aimed at maximizing the taxonomic diversity of our current genome collection (Eisen & Fraser, 2003).

The importance of a representative sample of genomes, especially from increased numbers of environmental bacteria, is underscored by the observation that the largest numbers of orphans are contributed by genomes that share one or more of the following characteristics: distant taxonomic relatedness, ecological uniqueness, or large genome size. For example, *Pirellula sp.1*, the first species belonging to the division Planctomycetes to be sequenced, produced 3576 orphan genes (49% of the total genes), despite being the 100th species to be sequenced. *Leptospira interrogans*, the third Spirochaetale to be sequenced and 92nd species, contains 2138 orphan genes (45% of the total genes). This genome contains two chromosomes, and the species can survive as either a saprophyte or as a facultative parasite. It is believed that *L. interrogans* was originally an environmental bacterium that has subsequently emerged as an important human pathogen (Ren *et al.*, 2003). The ability to inhabit two different environments, in addition to its past as an environmental organism, could help to explain the presence of such a large number of orphan genes. The two species described above are the two biggest sources of bacterial orphan genes in this dataset.

2.3 *Classifying Orphans as "Taxonomically Restricted Genes" of Potential Ecological Importance*

The cumulative number of orphans identified in complete bacterial genomes does not appear to be levelling off. This observation reflects both the small proportion of the total bacterial diversity sampled to date and the widespread occurrence of orphans in almost all bacterial taxa, with the exception of the very small genomes of intracellular parasites or endosymbionts. This suggests that, far from being non-coding "junk" DNA, these orphan sequences may be taxon-specific genes that, because of their restricted taxonomic distributions, may play an important role in bacterial adaptation. Databases are continuing to grow in size, and evidence is accumulating that orphans are often real genes (Daubin & Ochman, 2004a) rather than annotation artefacts (Skovgaard *et al.*, 2001). Therefore we should stop referring to orphans as 'mysterious' and start classifying them more appropriately as biologically significant "taxonomically restricted genes" (TRGs).

All genes are taxonomically restricted at some level. For example, any genes found in Eubacteria and not in Archaea or Eukaryotes are TRGs at the domain level. Genes restricted to Firmicutes or Proteobacteria are TRGs at the division level. The orphan genes reported in this study are TRGs at the species level because isolates of 122 different species were included in the analysis. Orphans, defined as species- or strain-level TRGs may be of special interest for their contributions to ecological adaptation. The concept of cataloguing genes that define (are restricted to) a given taxonomic group is already established (for example, Graham *et al.*, 2000), and we believe orphans firmly belong within this framework.

2.4 *Conclusion*

The availability of a large collection of complete prokaryotic genome sequences makes it possible to begin to explore in detail how the evolutionary diversification of gene content reflects the ecological needs and opportunities of different taxa. Surprisingly, few bacterial genes are truly universal (Charlebois & Doolittle, 2004), and many hypothetical coding regions appear to be unique to a given family, genus or species. It is also well known that strains within a species can vary greatly in their shared gene content (Lan & Reeves, 2000). The study of these 'taxonomically restricted' genes could reveal the genotypic basis of exclusive ecological adaptations. Furthermore, once the contributions of under-sampling of bacterial lineages and computational errors in gene prediction and assignment to gene families have been removed from our current estimated numbers of orphans, the number of orphans found in many genomes will likely become experimentally tractable. Therefore orphans, better defined as TRGs restricted to the species- and strain-levels, should be an important target of future study.

CHAPTER 3

QuickMine - A Computational Pipeline for the Analysis of Lineage-specific Bacterial Genes

3.1 Overview

The explosion in the number of complete genomes over the past decade has spawned the new and exciting discipline of comparative genomics. Biologically interesting features, such as pseudogenes and orphan genes, often only become apparent when placed in a comparative genomic context. There are now vast collections of genomes in public databases, however to exploit the full potential of this data requires the development of novel algorithms and software. The ability to compare these genomes brings a series of challenges. Issues of data storage, file formats and computational speed all become more complex and of greater importance (Field, Feil & Wilson, 2005b).

QuickMine is a suite of Perl scripts capable of the analysis of large volumes of genomic data. It has been written to interrogate such data, to find genomic features of interest, with particular emphasis on lineage-specific genes, including orphans.

In this chapter, I introduce the key concepts behind the functionality of the QuickMine pipeline and describe the development of the system. Section 3.2 outlines the aims and requirements of the QuickMine project. The design and implementation of the QuickMine system is described in 3.3, whilst the functionality of QuickMine is described in 3.4. Section 3.5 introduces a case study for the use of QuickMine in identifying orphan genes in bacterial genomes. Finally the performance of QuickMine is evaluated and future developments discussed in 3.6.

3.2 Project Aims and System Requirements

The purpose of QuickMine is to provide a computational pipeline for the analysis of lineage-specific genes in microbial genomes. The system requires, as input, sequence files for each chromosome. From these files, QuickMine generates a BLAST database. Every predicted protein in every proteome file will be BLASTed against this database to produce a BLAST report. QuickMine will allow users free access to modify the BLAST parameters to fit their particular analysis. Due to the volume of biological data stored in public repositories, it is no longer practical to manually examine all BLAST reports. Therefore the resulting BLAST reports will be parsed using Perl scripts to produce human readable output. For a detailed explanation of the functionality of BLAST, see Appendix 3.1.

QuickMine will be designed to allow flexibility with regards to the analyses it can perform. Submission files, to be used by computer clusters (see Appendix 3.2), will be available. These will be required when performing large-scale analyses.

In order to provide maximum use to the community, QuickMine will be designed for use by researchers with a range of computational abilities.

3.3 Design and Implementation

3.3.1 Language

QuickMine was written using the programming language Perl. Perl is widely used in the field of bioinformatics, largely due to its data processing abilities and the ease with which it can run external programmes (Wall, Christiansen & Schwartz, 1996). Additionally there is an active community of open source developers writing Perl modules specifically for use in Bioinformatics, known collectively as BioPerl (Stajich *et al.*, 2002).

3.3.2 Configuration File

Users interact with the QuickMine scripts through the use of a configuration file. This file contains all the arguments required to perform the QuickMine analysis. The configuration file:

- allows the user to determine which section of the pipeline they wish to run.
- is responsible for directing scripts to the relevant input files.
- selects the directory to which the output is written.
- allows the user to select the file endings for the output.
- provides a means for the user to adjust QuickMine's default parameters, for example, the command used to format the BLAST database.
- is written in simple human-readable format and is easily extendable.

The QuickMine scripts make use of the Perl module Config::Simple. This module enables scripts to obtain user specified parameters from the configuration file and use them as variables. The scripts utilise the Config::Simple module in an object-oriented manner. For example, the code below creates a new object ($cfg) containing the parameters from the configuration file ($config_file):

```
my $cfg = new Config::Simple($config_file);
```

User specified parameters are simple to obtain from the Config::Simple object. For example, the code to initialise a variable containing the path to print output to is shown below:

```
my $path2output = $cfg->param('path2output');
```

The configuration file used in QuickMine can be seen in Appendix 3.3.

3.3.3 QuickMine Input Sequences

QuickMine accepts input as DNA or protein sequence. It requires the input sequences to be in FASTA format. Each file may contain any number of FASTA-formatted sequences. When discussing QuickMine, each input file will be considered to be representing a genome, with each predicted coding region delineated by FASTA headers.

3.3.4 QuickMine and Condor

Due to the volume of biological data available, it is necessary to consider the time it will take to run QuickMine on a single machine. If QuickMine is being used to analyse several hundred viral genomes, running the process on a single machine is efficient. However, if QuickMine is used to analyse several hundred bacterial genomes, running on a single machine is not a viable option. The most computationally intensive stage of the QuickMine pipeline is performing the BLAST searches. Therefore, QuickMine provides the option of running the BLAST searches on a local machine or, alternatively, utilising a Condor cluster or the use of Grid technology through Globus. If a distributed computing environment such as Condor is selected, it needs to be specified in the QuickMine configuration file. The generation of submission files and monitoring of the jobs is the responsibility of the user. Once jobs are completed, the QuickMine pipeline can proceed. Perl scripts are available to assist the user in creating a submission file. In addition to using Condor for BLAST, it can also be used for running some of the more complex Perl scripts. Once complete, the output can be integrated with the remainder of the pipeline. This use of Condor needs to be managed independently by the user. For a discussion of both Condor and Grid technologies, see Appendix 3.2.

3.3.5 Dependencies

In addition to the Perl modules that come with the distribution, QuickMine requires BioPerl to be installed. BioPerl (Stajich *et al.*, 2002) is a comprehensive library of Perl modules developed in an open-source environment. The modules are designed for use in managing and manipulating biological data. QuickMine utilises Bio::SearchIO for parsing through BLAST reports and Bio::SeqIO for parsing through sequence files.

Another programme required for QuickMine to function successfully, is Gnuplot. Gnuplot is a command-line driven plotting utility. It is freely distributed and is available from http://www.gnuplot.info. QuickMine utilises Gnuplot for all data plots.

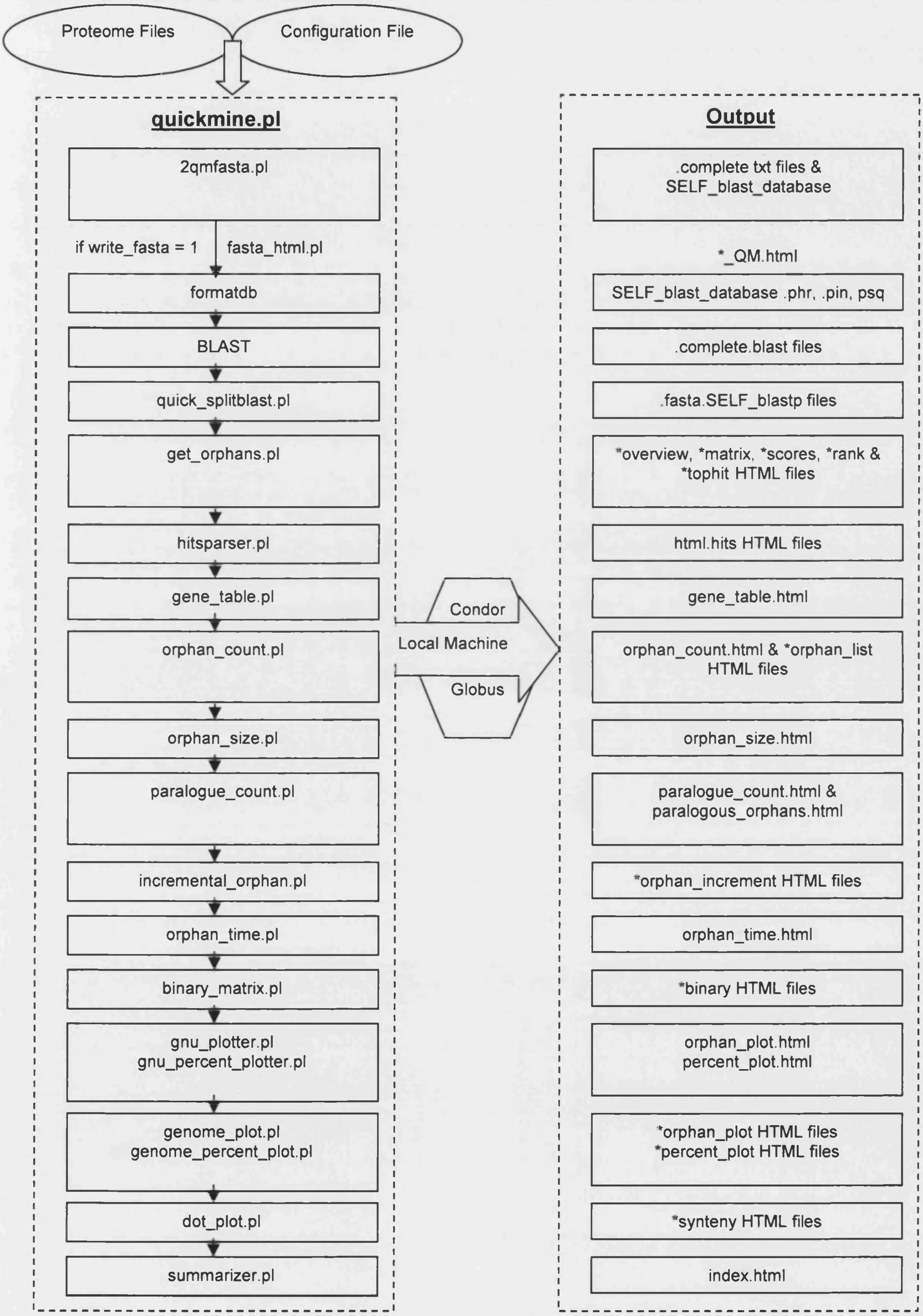
3.4 The QuickMine Pipeline

The QuickMine pipeline consists of eighteen Perl scripts and one configuration file. The Perl script *quickmine.pl* is the script executed by the user. This script is responsible for executing the other Perl scripts. It is also responsible for executing the BLAST searches. The script obtains all the variables from the configuration file. These values are used to determine which sections of the pipeline need to be run and also provide the parameters required by the processing scripts. The QuickMine pipeline can be broadly split into four groups:

1. Pre-processing
2. BLASTing
3. Parsing
4. Plotting

These are described below and can be seen diagrammatically in Figure 3.1.

Figure 3.1. A Diagrammatic Representation of the QuickMine Pipeline. The left hand column lists the Perl scripts that constitute QuickMine, the right hand column lists the main output files produced at each stage of the process.



3.4.1 Pre-Processing

The Perl script *2qmfasta.pl* is responsible for formatting the sequence files ready for use further down the pipeline. This involves adding a unique identifier to the start of the FASTA header, in the format 'file_idorf0000'. The files produced by the script are called '.complete' and are used to generate the self BLAST database. If the value of the 'write_fasta_files' parameter, obtained from the configuration file, is equal to 1, *2qmfasta.pl* will generate a FASTA file for each predicted protein. The script *fasta_html.pl* will then generate a web interface (QM.html), allowing users' access to each FASTA file. *2qmfasta.pl* also concatenates all the formatted input files together, producing a single file called SELF_blast_database.

Quickmine.pl performs a system call, prompting the execution of *formatdb* using the parameters provided in the configuration file. *Formatdb* is a programme for formatting BLAST databases from either FASTA or ASN.1 formats; in this case it formats the SELF_blast_database FASTA file. *Formatdb* generates several files necessary for the successful execution of BLAST. In the case of a protein database, *formatdb* generates 3 files: SELF_blast_database.phr, SELF_blast_database.pin and SELF_blast_database.psq.

3.4.2 BLASTing

When running the entire pipeline on a single machine, *quickmine.pl* is responsible for initiating the BLAST searches. The parameters used in the BLAST search are determined by the user input in the configuration file. An alternative is to stop the pipeline at this point and use Condor for performing the computationally intensive BLAST searches. The script *make_cmd.pl* is available for creating a Condor submission file. Once Condor has finished its jobs, the pipeline can be restarted from the first script in the parsing group (*get_orphans.pl*). In the event of the user having access to a Grid system, the script *make_globus_cmd.pl* is available for creating a Condor submission file that submits jobs to the globus universe.

3.4.3 Parsing

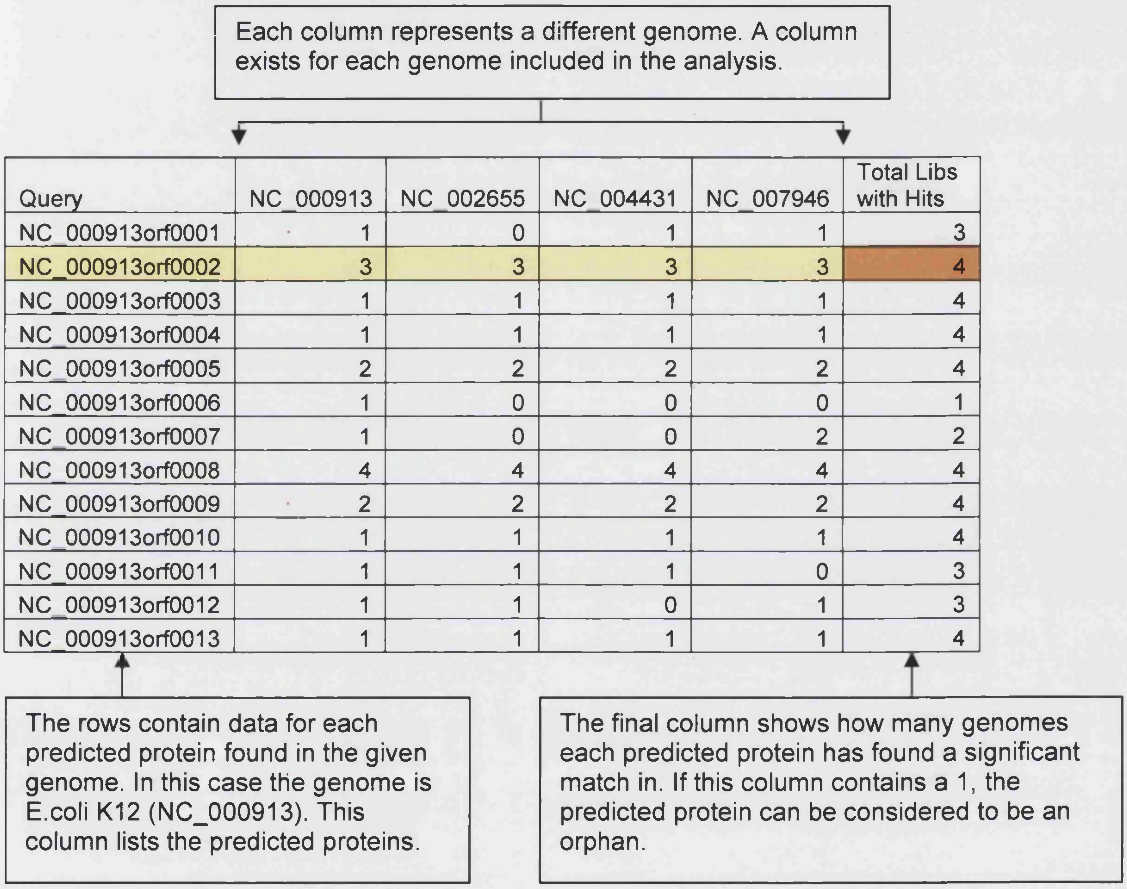
This section of the pipeline generates the majority of the human readable HTML output. The first script in this section is called *get_orphans.pl.* It utilises the BioPerl module 'Bio::SearchIO' to parse through the BLAST reports. *Get_orphans.pl* generates five

HTML files for each input file (or genome). The overview.html file (Figure 3.2) is the most important file created. It constitutes a matrix in which each row represents a predicted protein from the query genome and each column represents a different genome. The numerical value in the element XY indicates the number of predicted proteins in the genome represented by column Y that possess significant similarity to the predicted protein in row X. The final element in each row displays the total number of genomes containing a match to the predicted protein. This overview file is the input of several scripts further down the pipeline. The second output file, matrix.html, has the same matrix format as overview.html. However, in this file, element XY shows the best hit (the protein with the most significant match), from the genome in column Y to the predicted protein in row X. The third output file, rank.html, lists the predicted proteins in the query genome and shows the top hits from each of the other genomes in rank order. The fourth file, tophit.html, lists the predicted proteins in the query genome and shows the single best hit to each protein, the E-value of that hit and the FASTA header information accompanying that hit. The overview.html, matrix.html, rank.html and tophit.html files all provide a link to each predicted protein's BLAST report. The final output file, scores.html, lists all the hits, and the E-value of each hit, to each predicted protein.

In some cases, there may be hundreds of thousands of BLAST reports to parse; hence *get_orphans.pl* can take a long time to run. An alternative to running *get_orphans.pl* as part of the pipeline is to run it on Condor. The script *make_perl_cmd.pl* is available for creating a suitable Condor submission file. Once *get_orphans.pl* has been run on each proteome, the QuickMine pipeline can be restarted from the next script (*hits_parser.pl*).

Hits_parser.pl produces a hits.html file for each genome. The file contains a list of all the genomes that the query genome has been compared against, and displays the number of predicted proteins in the query genome that hit each genome. It displays this value as a percentage of total predicted proteins. It also displays the number of total hits, i.e., some predicted proteins may hit more than one predicted protein in a particular genome.

Figure 3.2. Example output from the overview.html file, generated by *get_orphans.pl*. The left hand column (Query) lists the predicted proteins in the given genome (NC_000913); the right hand column (Total Libs with Hits) shows how many genomes contained a significant match to each predicted protein. All other columns represent genomes used in the analysis (NC_000913, NC_002655, NC_004431, NC_007946), and show how many significant matches were found to the predicted protein in column 'Query'. For example, genome NC_002655 contains 3 proteins with significant similarity to predicted protein NC_000913orf0002. In total, 4 genomes contain one or more matches to this predicted protein.



Each column represents a different genome. A column exists for each genome included in the analysis.

Query	NC_000913	NC_002655	NC_004431	NC_007946	Total Libs with Hits
NC_000913orf0001	1	0	1	1	3
NC_000913orf0002	3	3	3	3	4
NC_000913orf0003	1	1	1	1	4
NC_000913orf0004	1	1	1	1	4
NC_000913orf0005	2	2	2	2	4
NC_000913orf0006	1	0	0	0	1
NC_000913orf0007	1	0	0	2	2
NC_000913orf0008	4	4	4	4	4
NC_000913orf0009	2	2	2	2	4
NC_000913orf0010	1	1	1	1	4
NC_000913orf0011	1	1	1	0	3
NC_000913orf0012	1	1	0	1	3
NC_000913orf0013	1	1	1	1	4

The rows contain data for each predicted protein found in the given genome. In this case the genome is E.coli K12 (NC_000913). This column lists the predicted proteins.

The final column shows how many genomes each predicted protein has found a significant match in. If this column contains a 1, the predicted protein can be considered to be an orphan.

Orphan_count.pl parses through the overview.html files to determine which predicted proteins do not have significant similarity to any predicted protein in a different genome (classed as an orphan). It lists these orphan genes in orphan_list.html files and provides a summary of the number and percentage of orphans in each genome analysed, in orphan_count.html.

Orphan_size.pl produces orphan_size.html and 'orphan.complete' files. The BioPerl module Bio::SeqIO is used by *orphan_size.pl* to parse through the '.complete' files and search for the orphans listed in the orphan_list.html files. Once identified, their sequence is printed out to the 'orphan.complete' files and the number of amino acids is counted. If an orphan sequence contains less than 150 amino acids, it is deemed to be a short orphan. If the sequence contains 150 amino acids or greater, it is classed as a long orphan. The number of each class of orphan is counted up for each genome and the average orphan size is calculated. This information is printed to the orphan_size.html file.

Paralogue_count.pl produces paralogous_orphans.html and paralogue_count.html. Paralogous_orphans.html lists the orphan genes in each genome that have significant similarity to another predicted protein in the same genome and displays the number of proteins the orphan is significantly similar to. Paralogue_count.html provides a summary of the number of paralogous orphans present in each genome.

Incremental_orphan.pl parses through the overview.html files to generate orphan_increment.html files. These files show the same matrix as the overview.html files, however it has an additional indicator column for each genome. This column indicates whether the relevant predicted protein is still considered to be an orphan, i.e., does not possess a significant hit to any predicted proteins in this genome or any of the preceding genomes. If a hit has been found, the indicator column will contain an 'N' (representing non-orphan), if a hit has not been found, it will contain a 'Y'. Once a hit is found, the indicator columns will be set to 'N' for the remainder of the row. The script *orphan_time.pl* uses the orphan_increment.html files to generate orphan_time.html. This file contains a matrix. Each row and each column represents a genome. The number in the element XY^3 represents the number of orphans in the genome X, after being BLASTed against genome Y^3 and also genomes Y^2 and Y^1. Thus the matrix provides data illustrating the change in orphan number in each genome, as more genomes are added to the comparison.

Binary_matrix.pl converts all the values in overview.html files to a 0 (no hits) or a 1 (hit at least one predicted protein in the respective genome).

3.4.4 <u>Plotting</u>

All the scripts written to generate plots utilise Gnuplot. *Genome_plot.pl* generates a plot for each genome, describing the change in orphan number as more genomes are sequenced. It obtains the data from orphan_time.html. In order to produce the plot, several files are generated. Gene_plotter_commands.dat contains the Gnuplot commands necessary for generating the desired plot. Gene_plotter.dat contains the data in a format that can be read by Gnuplot. Gnuplot creates the plot in png format. *Genome_plot.pl* uses a system call to convert png to jpeg. Finally, it generates orphan_plot.html to display the jpeg image. *Genome_percent_plot.pl* is identical to *genome_plot.pl* except it converts the data in orphan_time.html to a percentage of total predicted proteins in each genome.

Gnu_plotter.pl and *gnu_percent_plotter.pl* are very similar to *genome_plot.pl* and *genome_percent_plot.pl.* However, instead of generating a plot for each genome, they generate a single plot displaying a line for each genome.

Dot_plot.pl utilises the data in matrix.html to produce a dot plot of each genome against every other genome. Such plots can give an indication of how closely related two genomes are, and can be useful in finding regions of inversion in closely related genomes. As in the other plotting scripts, it produces a data file, a command file, a png file, a jpeg file and a HTML file. As different files are created for every combination of genomes, it is easy to accumulate a large number of files very quickly.

The final script in the pipeline is *summarizer.pl.* This script generates the file index.html. By default, index.html will be loaded by web browsers when viewing the output directory. *Summarizer.pl* generates a list of all the HTML files created in the QuickMine pipeline and prints links to each output file in the file index.html. Thus, it provides an easy and simple method for the user to navigate through their results.

3.4.5 QuickMine and OrphanMine

Many of the files generated by QuickMine are parsed and formatted for use in OrphanMine. This will be discussed in more detail in Chapter 4.

3.5 Using QuickMine for the Identification of Orphan Genes

Surprisingly, few bacterial genes are truly universal, and many hypothetical coding regions are unique to a given genus or species. It is likely that these sequences play a significant role in defining exclusive ecological adaptations. It has been stated that the frequency of orphan genes has been one of the most surprising results to come from the analysis of bacterial genome sequences (e.g. Doolittle, 2002) and explaining their abundance and functional relevance remains a key challenge in bacterial genomics.

QuickMine was used to generate a list of orphan predicted proteins that could be publicly displayed in the OrphanMine. In this section, the parameters used for the analysis will be described. The results of the analysis are described in Chapter 2.

3.5.1 Data Source

The analysis was performed on the complete genomes of 122 bacterial species. QuickMine required one input file for each chromosome analysed. These files were obtained from the NCBI (ftp.ncbi.nih.gov/genomes/Bacteria) and had the file extension '.faa'. Each '.faa' file contained all the predicted protein sequences from that particular chromosome. The protein sequences were in FASTA format. The NCBI produced these files from the original GenBank (Benson *et al.*, 2006) record using three gene prediction programmes Glimmer (Delcher *et al.*, 1999), GeneMark (Besemer & Borodovsky, 2005) and GeneMark.hmm (Lukashin & Borodovsky, 1998). The predicted proteins were searched against 'NCBI-nr' (the NCBI's non-redundant sequence database). In the case of over-lapping genes, those showing higher sequence similarity to proteins in the database were retained. The collection of complete microbial genome sequences, obtained from the NCBI, is a part of the NCBI Reference Sequence Project (RefSeq), the aim of which is to provide curated sequence data and related information to the community (Pruitt, Tatusova & Maglott, 2005). The RefSeq accession numbers are formatted as two letters followed by an underscore, followed by six, eight or nine numbers. Different alphabetic prefixes indicate the process of generation and the type of molecule processed. This analysis was performed on complete microbial genomes; therefore all files were prefixed with 'NC'. QuickMine was responsible for formatting these files, creating a BLAST database, performing each BLAST job and generating output files in human readable format.

3.5.2 BLAST Parameters

As the input sequences were protein and were used to create the BLAST database, BLASTP was used to perform the alignments. The parameters used in a BLAST search can greatly affect both the sensitivity and the speed of the process. For an analysis of lineage-specific genes, it was necessary to search for distant relatives. Therefore sensitive parameters were required. By default, QuickMine used a significance threshold of 10^{-3} to define a hit. This cut off was chosen as it would permit predicted proteins to be matched with distant, potential homologues and is a threshold commonly used in bacterial genome annotation pipelines. Specifically, this threshold was used in the analyses of Siew & Fischer (2003a). They hypothesised that the maximum number of orphans would be 26,000. In order to test this hypothesis, it was necessary to use the same e-value. The neighbourhood word threshold was lowered from the default BLASTP value of 11 to 9 for use in QuickMine. This increased the chance of an alignment being seeded. The protein similarity matrix was also changed from the BLASTP default of BLOSUM62 to BLOSUM45. The BLOSUM45 matrix was generated by using blocks of proteins that possessed at least 45% sequence identity to another member of the block. This change allowed for greater sequence divergence between reported matches. By default, BLASTP masks low complexity regions in a protein. Soft-masking masks low-complexity sequence in the seeding phase, but allows the extension phase to see the sequence normally (as opposed to the low complexity region being replaced by Xs). Therefore, complexity filters were set to use soft-masking.

An example command used for running BLASTP in these analyses is shown below:

```
blastall -p blastp -i NC_000907.faa.complete -d
SELF_blast_database -o NC_000907.faa.complete.blastp -e 1e-3 -b
500 -f 9 -F 'mS' -M BLOSUM45
```

In the above command, *blastall* is the name of the BLAST command line executable. The –p argument refers to the BLAST programme that will be run. The –i and –o parameters refer to the input and output files respectively, -b is the number of alignments allowed in each report and –d is the database to search against. The parameters that affect the speed and specificity of the BLAST search are –e which is the E-value threshold, -f is the neighbourhood word threshold, -F designates the complexity filter and –M determines the protein scoring matrix. All these parameters can be modified through the configuration file.

3.5.3 Using Local Condor Cluster

Due to time constraints it was necessary to utilise the on-site Condor cluster to perform the BLAST searches. It required 51 hours and 47 minutes to search the 2950 predicted proteins of the *Corynebacterium efficiens YS-31 4T* genome (the genome closest to the mean value of 2910 predicted proteins in the dataset of 122 genomes) against the self-BLAST database. Hence, to perform the BLAST searches for all genomes on a single machine would have taken approximately 6318 hours or 263 days.

The Condor cluster at CEH Oxford was comprised of 48 nodes. Therefore, it was possible to complete the same number of BLAST jobs in under a week (263/48 = 5.48 days), by submitting the jobs to the cluster. The performance of Condor will be discussed in more detail in Section 3.6.4

3.6 Evaluation and Future Developments

QuickMine was developed for the analysis of microbial protein files. The current system meets the requirements. However, there are issues related to QuickMine and the methods employed by QuickMine that need further discussion.

3.6.1 Time Constraints

The time taken to perform a QuickMine analysis varies, depending on the quantity of input data and the computer architecture performing the analysis. The time-limiting steps in the QuickMine pipeline involve performing the BLAST searches and parsing through the resulting BLAST reports using Bio::SearchIO. When dealing with large numbers of bacterial genomes, these two stages can take several months on a single machine. However, analysing several hundred plasmid or viral genomes on a single machine is not such an issue. There are a number of options available to speed up the analyses. Firstly, the BLAST search parameters can be altered. Instead of performing a slow sensitive search, the user can select to perform a fast but less sensitive search. For example, changing the neighbourhood word threshold so that fewer alignments are seeded will speed up the search. If matches are expected to be very similar, a different scoring matrix such as BLOSUM80 can be selected. It is also possible to reduce the number of results shown in the BLAST report by altering the search threshold E-value, or by setting a limit on the number of alignments displayed in the report. This will result in less data being generated and therefore will speed up the data parsing process.

If a faster, less sensitive search will not provide the results required, as was the case when searching for lineage-specific genes, it may be necessary to use a computer cluster, such as a Condor cluster. Scripts are provided with the QuickMine distribution. These scripts provide example submission files for the BLAST jobs and also for *get_orphans.pl*, the script that parses the BLAST reports. The increase in speed depends upon the size of the cluster. If the cluster is not large enough, it may be appropriate to obtain a Grid certificate and utilise Globus.

Reducing processing time is going to be a major challenge for bioinformatics software as the volume of sequence data continues to increase rapidly (http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html). New, more efficient algorithms may be developed, however increasingly efficient use of computing power may be a long term solution. Opening up Grid technologies to the wider community may enable pipelines such as QuickMine to run, without the need to compromise output due to time-constraints.

3.6.2 Data Storage

The issue of data storage is similar to that of time constraints. The extent of the problem is dependent on the data being analysed and the sensitivity of the BLAST reports. Performing all-against-all BLASTP searches using sensitive BLAST parameters, as described in section 3.5, will create more output. For example, performing QuickMine, using sensitive BLAST parameters, on 150 bacterial genomes generated approximately 195 gigabytes of data. Clearly, this volume of data cannot be stored on a typical desktop computer. As more sequence data becomes available, larger comparative analyses are likely to be performed. Before running such analyses, it is important to consider the output of the analyses and the storage of the output.

3.6.3 Use of E-values

The central element of the BLAST algorithm is the Karlin-Altschul equation (Altschul *et al*, 1990):

$$E = kmne^{-\lambda S}$$

The equation states that the number of alignments expected by chance (E) during a sequence database search, is a function of the size of the search space (m * n), the normalised score (λS) and a minor constant (k). The size of the search space is a product of the length of the query sequence (m) and the number of letters in the database searched (n). Lambda (λ) is a matrix-specific constant responsible for converting the raw score to a normalised score. The lower the value of E, the less likely it is that the alignment is a result of random similarity.

For example, if Query A was searched against two databases of different sizes (for example, subsequent versions of the same database) that both contained Sequence A, the resulting perfect matches (100% identity) will have different expect (E) values. This is due to the positive linear relationship between the size of the database and the expect value. Therefore, if database size doubles, so too does the E-value (i.e., a decrease in significance). Hence, changes in database size can have a significant impact on the biological interpretations derived from similarity searches. This is particularly important when analysing genes, such as orphans, that are defined by their lack of significant similarity to other predicted proteins.

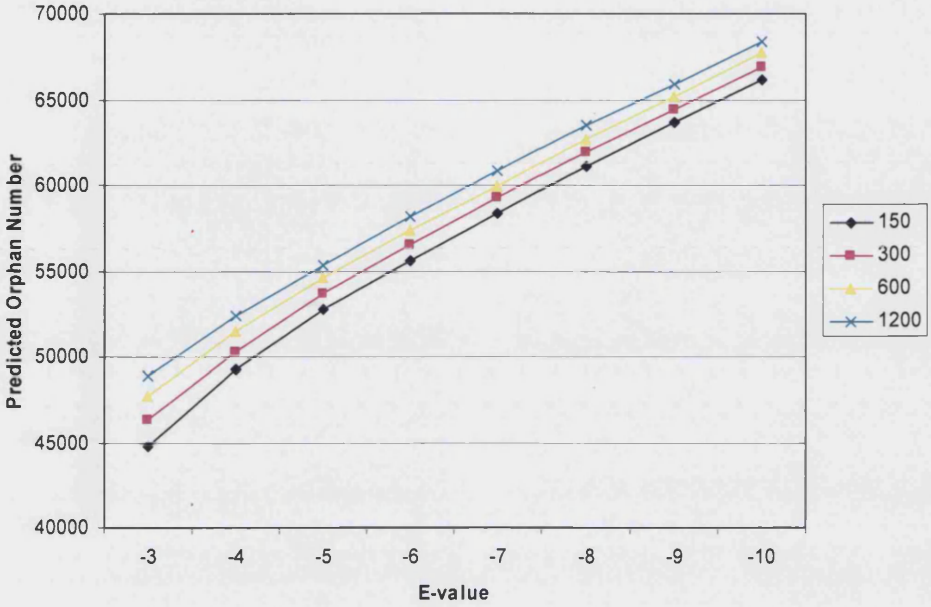
QuickMine utilises E-values to infer homology and create a list of orphan genes. An analysis was performed to determine the effect of a change in database size on orphan genes. The BLAST reports of 150 bacterial genomes were parsed to obtain the data required to calculate the E-value. The E-value was calculated for each predicted protein for the actual database and for 3 virtual databases. This was done by modifying the database size (n) in the Karlin-Altschul equation. The virtual databases represented different numbers of bacterial genomes. Based on the size of the real database containing 150 genomes, it was possible to estimate the size of databases, in amino acids, if they were to contain 300, 600 and 1200 bacterial genomes. The results show that as database size increases, so too does orphan number, despite using the same sequence data. This is because, low scoring matches, in a relatively small database (e.g. 150 genomes), are more likely to be deemed as statistically significant than the same low scoring match in a large database (e.g. 300 genomes). Figure 3.3 shows the results of this analysis. With the database size the equivalent of 300 bacterial genomes, 46367 predicted proteins were deemed to be orphans. This is an increase of 1615 orphans when compared with the database of 150 bacterial genomes, which contained 44752 orphans. The database representing 600 bacterial genomes showed an increase of 2910 orphans and the database representing 1200 bacterial genomes had 48930 orphans, an increase of 4178. These results illustrate the relative nature of

E-values. Hence when analysing output from QuickMine, these issues should always be considered.

A second issue involving E-values is that of query sequence length. It is theoretically possible for sequences of any length to fail to produce a significant match to self, in databases of large enough size. Again this property is a result of the algorithm used to calculate E-values. This issue is becoming a reality with smaller sequences. In a recent study of homology between the genomes of 18 complete baculovirus genomes (personal correspondence from Sarah Turner), it was found that several predicted proteins in each proteome (approximately 5%) failed to produce a significant match in a BLAST search, even though exact copies of these genes were present and the subject database was very small (2500 proteins). Scrutiny of these genes revealed that they were extremely short or contained regions of low complexity. In a dataset of 150 bacterial genomes containing 430826 predicted proteins, 98 predicted proteins failed to match self. These predicted proteins were smaller than 25 amino acids in length (with two exceptions of length 96 and 104 amino acids, but with percentage low complexity of 95% and 90% respectively) and were annotated as hypothetical or as operon leader peptides. As databases increase in size, more sequences will fail to match themselves in homology searches. QuickMine would count these predicted proteins as orphans. Included in the QuickMine distribution is a Perl script, *no_self_hit_count.pl,* that can be run to determine how many orphan genes do not hit self. It also provides a list of these predicted proteins.

The issues described above are challenges beyond the scope of the QuickMine project. However, it is important that users of the QuickMine system are aware of the effect database size can have on their analyses.

Figure 3.3. Change in predicted number of orphans obtained from 150 bacterial genomes at different E-value thresholds, as database size is artificially increased. The size was increased to represent 300, 600 and 1200 bacterial genomes.



3.6.4 The Performance of Condor

As described in Appendix 3.2, the ability to utilise Condor and Grid technologies can save vast amounts of time. The analysis of 122 bacterial species would not have been feasible without access to a computing cluster. However, whilst a large amount of time was saved, the performance of Condor was far from optimal. Condor and Grid are new technologies and as such there is limited knowledge available in the area. This lack of knowledge can affect several stages of the process. For example, creating a submission file that will perform the relevant jobs optimally is not trivial. Also, determining whether or not an error has affected the submissions and, if so, determining the nature of the error, requires knowledge of the cluster and the ability to navigate through large log files. Increasing the general level of expertise in this area will make it more accessible to a greater number of researchers.

In addition, the application being run on the cluster can affect how efficiently the jobs will progress. In the case of QuickMine, the majority of work involves use of the BLAST executable, *blastall*. To run *blastall* on the Condor cluster requires the use of the vanilla Condor universe. If a job is dropped off a machine because, for example, a different user has taken control of the machine, all data generated for that job, up to that point, will be lost. This is because the vanilla universe does not permit jobs to undergo check-pointing (allows jobs to continue from a particular point). Obviously this lack of functionality can cause large time delays, particularly if the nodes in a cluster are not stable.

3.6.5 Integration of QuickMine into YAMAP

YAMAP, originally developed by Dr. Milo Thurston, is a Perl application created for the NERC Microbial Metagenomics programme (http://www.genomics.ceh.ac.uk/mm/). It utilises Perl TK to provide a graphical interface to the user. YAMAP is designed to allow users to run a selection of first pass annotation tools on their sequence data. Examples of these tools include Glimmer (Delcher *et al.*, 1999) and tRNAscan (Lowe & Eddy, 1997). The application is available to Bio-linux users (Field *et al.*, 2006).

In 2006, I was responsible for incorporating the QuickMine functionality into the YAMAP application. QuickMine was able to provide new options and improved functionality to users of the YAMAP system. The QuickMine code integrated into YAMAP has undergone slight modifications from the code that is available as a stand-alone command line programme. These changes are largely due to the restrictions imposed by the Graphical User Interface (GUI) or as a result of the requirement to make the output data from QuickMine suitable for further downstream analysis in YAMAP.

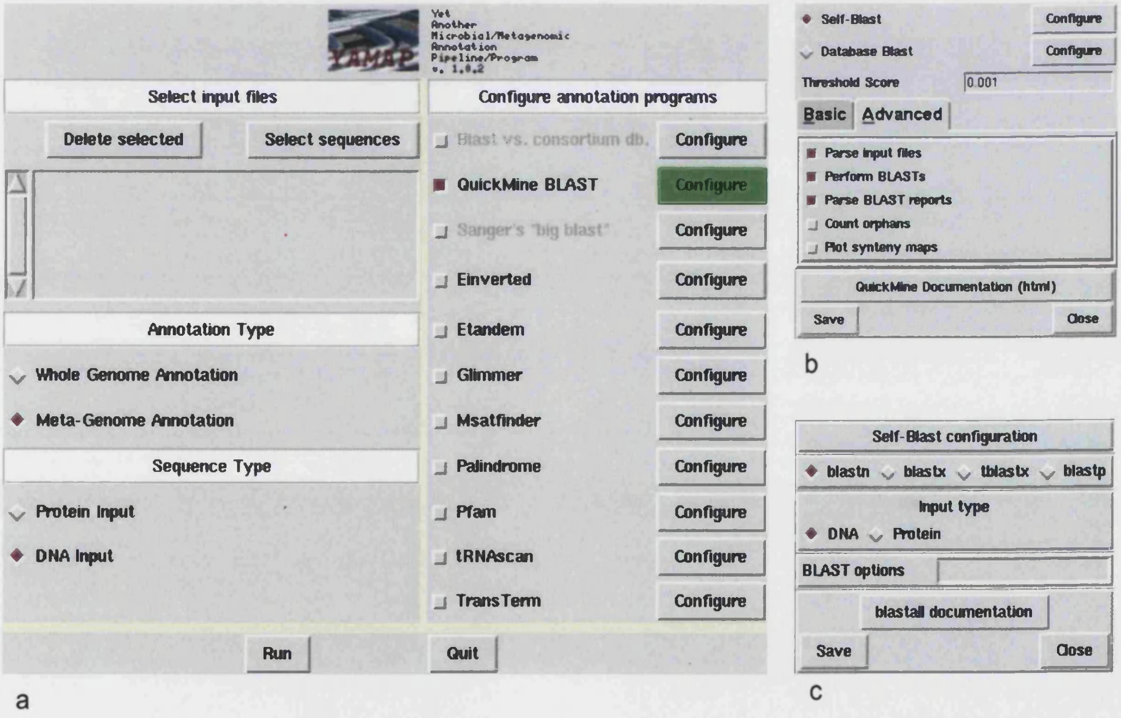
A major advantage gained from the integration of QuickMine into YAMAP, is the use of the GUI (see Figure 3.4). Command line programmes are often found to be intimidating to casual users and may discourage them from using QuickMine. In addition, whilst the configuration file is written in a basic text format, it is possible that less experienced users may introduce hidden characters (for example, line breaks) into the file. Such characters may cause problems when QuickMine obtains parameters from the file. Providing the ability to use QuickMine through a GUI reduces these problems. The GUI makes it a more attractive piece of software for people to use and provides a barrier between less experienced users and the configuration file, thus making the system more stable.

In addition, the GUI has made it possible to present options to QuickMine users that would not have otherwise been known. For example, QuickMine was initially designed for use with self BLAST databases. However, it is also possible to use QuickMine to

BLAST sequences against public databases such as SwissProt. To perform this task using the command line version of QuickMine would require extensive knowledge of the code and the structure of the pipeline. In contrast, by using the GUI, it is possible to provide such tasks as options in the menu, thereby allowing inexperienced users to access extended functionality.

For users with large datasets, the command line version of QuickMine should still be the preferred option. It is more efficient in terms of disk space and time. In addition, it provides more flexibility, thus making it easier to integrate the use of computing clusters into the analysis.

Figure 3.4. YAMAP's Graphical User Interface. On running YAMAP, the user will view (a) the main application window. From here it is possible to select which files to analyse and the types of analysis to be performed. Programme specific windows appear when analyses are configured to select suitable parameters. (b) shows the option window for QuickMine. One of the parameters that need to be entered involves BLAST, to set these parameters, the BLAST window (c) is launched from within the QuickMine window.



3.7 Availability

QuickMine is available on request as a tar file. It is also distributed as part of the YAMAP Debian package available to members of the Environmental Genomics Bio-Linux community.

CHAPTER 4

OrphanMine – A Database for the Analysis of Lineage-specific Genes

4.1 Overview

In the Roberts report (2004) for the American Academy of Microbiology, the need for a prioritised list of genes of unknown function was highlighted. The need for such a list has been elevated by the recent recognition of the pan-genome concept and the realisation that genetic diversity has been vastly underestimated. Many genes of unknown function are restricted to a particular species. Such genes can be referred to as lineage-specific or taxonomically restricted.

Lineage-specific gene lists are particularly sensitive to the dataset used in an analysis. Both the thresholds used in an analysis and the quality of input data have an effect on the output. The output, i.e., the list of lineage-specific genes, is comprised of real biological genes found only in one species or strain, real taxonomically restricted genes that appear as a result of incomplete sampling and, finally, sequences incorrectly annotated as coding (mis-annotation of the genomic sequence). Lists of orphan genes have been produced previously; for example the Orfanage (Siew *et al.*, 2004) and CUPID (Mazumder *et al.*, 2005). Such examples enable the user to generate lists of taxonomically restricted genes. However, it is important that resources provide a method for the user to obtain metadata which describes the orphan genes in an acceptable format. Previously available tools did not provide this function, thus any additional annotation was subsequently lost. Additionally, these resources sought to provide lists of lineage-specific genes without prioritising the genes for experimental characterisation.

OrphanMine, a web-based tool, provides a structured platform to share knowledge with researchers in a logical and natural manner. The underlying data is obtained from QuickMine and formatted for entry into the OrphanMine database. The web interface is generated using PHP, which communicates with the OrphanMine MySQL database. Task-specific help pages, designed to assist the user, have been implemented. OrphanMine provides the microbial community with a new online resource for investigating lineage-specific genes that may be involved in ecological adaptations. Any dataset of genes that a user is interested in can be ranked, according to the likelihood of being a real gene. Additionally, the genes and associated metadata can be printed out in GFF format. It is anticipated that the lists of sequences generated using this database will provide the starting point for subsequent characterisation of particular groups of predicted proteins, through empirical or *in silico* means.

In this chapter, I will describe the design and implementation of the OrphanMine. Section 4.2 provides a brief discussion of knowledge sharing in the biological sciences. Section 4.3 introduces the aims of the project and the requirements of the system before a discussion of the design methodology in section 4.4. Section 4.5 describes the system prerequisites. The database design is described in 4.6 and descriptions of the database tables are provided in 4.7. Section 4.8 discusses the design and functionality of the systems web interface, whilst the evaluations of the OrphanMine system are described in 4.9. Finally a discussion of the system's performance is found in section 4.10.

4.2 Knowledge Sharing

The use of computers to store knowledge has led to the development of 'knowledge bases'. One category of knowledge base is the scientific knowledge base. Their aim is to be a model of a domain of scientific investigation. These knowledge bases are regularly updated and require common sense knowledge to be understood but do not intend to capture it. They may constitute an exchange medium among researchers and may accelerate the scientific discovery process. Molecular biology is a very good example of a discipline which can benefit from knowledge base building. The primary role of a scientific knowledge base is to be a model that helps the researchers to structure their knowledge into a consistent consensual form; it must therefore offer good browsing facilities and allow complex requests (Rechenmann, 1995). It is also highly important that the links between pieces of formalised knowledge and experimental results or data are maintained in order for the scientist to evaluate the degree of validity of the knowledge, for example in scientific literature (Rechenmann, 1995).

Studies in bioinformatics activity have led to the identification of eight distinct categories of science knowledge bases (McMeekin & Harvey, 2002). Three characteristics are used to determine to what class a knowledge base belongs. The combinations of these characteristics lead to the formation of the eight categories. The first characteristic refers to the extent to which the knowledge is accessible after it has been produced; a knowledge base is therefore either open or closed. The second characteristic refers to whether the knowledge is traded. The third characteristic refers to the type of institution that produced the knowledge i.e., private or public (McMeekin & Harvey, 2002). This project will involve the creation of an open knowledge base, developed in the public domain, in which the knowledge is not traded.

In an ideal world, research groups distributed round the world working on similar projects would be in communication with each other. Such communication would enable results and data to be shared and reduce knowledge loss, thus resulting in a more efficient research environment. Previously, this would have been difficult to establish without incurring large travelling expenses, possibly outweighing the benefits gained from knowledge sharing. Information systems, in this case a relational database, enable communities to overcome time and space constraints in knowledge sharing and increase the speed and range of access to information (Ramarapu, Simkin & Raisinghani, 1999), thus forming a virtual community. In biological research, formation of these communities could have great benefits, specifically with regards to genome annotation. Quality genome annotation is currently a bottleneck in the progress of the genome projects. Much of the annotation is done automatically, however these methods provide only a baseline annotation. The problem faced by biologists is how to go beyond this basic level. As Hubbard & Birney (2000) discuss, no single collaborative group will be capable of annotating an entire genome consistently and to a high quality. One solution is to have a 'monolithic single entity that invests 300 person years into annotating the genome' (Hubbard & Birney, 2000). A second and more attractive solution is 'open annotation', where the required annotation is distributed across a community of biologists.

OrphanMine encourages the community to improve upon the current standard of genome annotations. It does this by providing the user with the option of downloading their data of interest from OrphanMine in GFF3 (Generic Feature Format Version 3). Although there are many richer ways of representing genomic features, for example using XML, the preference in the community is for a simple format that can be easily edited either manually or through the use of a script (http://www.sequenceontology.org/gff3.shtml). Previous versions of the GFF format did not provide the required flexibility for many users, which led to different groups extending it in different ways. GFF3 allows users to add any feature of interest to the file whilst remaining in a standard format. Hence, users are able to utilise their files in a variety of different programmes, one example being Artemis (Rutherford *et al.*, 2000). OrphanMine allows users to download their list of genes with additional annotation, for example, the gene's rank score and criteria for ranking. These files can then be layered on top of one another in annotation programmes, such as Artemis, allowing the user to decide upon the validity of a particular annotation. Thus, OrphanMine encourages data sharing whilst preventing knowledge loss.

4.3 Project Aims and System Requirements

The aim of this project was to develop a scientific knowledge base. The knowledge base, called OrphanMine, will take the form of a freely available web-based tool, thus ensuring the information will be immediately available. OrphanMine will provide access to all predicted proteins in all publicly available complete bacterial genomes. It will allow the user to explore the collection of predicted proteins using several search filters, specifically with the intention of assisting in the study of lineage-specific genes. The search filters will allow subsets of predicted proteins to be selected from the complete proteome dataset based on a number of different criteria, for example, the E-value, percentage low complexity, GC content or sequence length. It will also be possible to filter proteins, based on their occurrence in other genomes. Additionally, pre-computed datasets of lineage-specific genes will be available for browsing.

The user will have the opportunity to explore selected genes further by viewing associated metadata and BLAST reports. In addition, all predicted protein sequences will be annotated with supplementary information (see 4.8.3). It will be possible to visualise the distributions of selected proteins in a genomic context with either the Artemis application (Rutherford *et al.*, 2000) or the CGView java applet (Stothard & Wishart, 2005). A search page will be available for basic text searches and an interface will be provided for advanced users to gain direct access to all the database tables using SQL. Users will also be able to BLAST new sequences against the OrphanMine, SwissProt (Boeckmann *et al.*, 2003), COGs (Tatusov *et al.*, 2003) and Pfam (Bateman *et al.*, 2004) databases.

OrphanMine will be designed in a flexible manner allowing for a variety of analyses. For example it could be used to identify predicted proteins restricted to any group of interest, for example, a particular taxonomic group or an ecologically relevant group. An example query could be to find all predicted proteins in *Mycoplasma pneumoniae* that are taxonomically restricted to the genus Mycoplasma. The results page will display only the predicted proteins restricted to this genus.

A further requirement of OrphanMine is to provide the user with a method for ranking any selected subset of genes. It will be possible to rank according to any combination of five criteria (length, percentage low complexity, difference in GC content from the genome average, neighbourhood distribution and average amino acid metabolic cost). A score will be calculated for each of the genes and the genes will be ranked accordingly. Thus, the user will be provided with a prioritised list for experimental characterisation.

The final and essential requirement is to provide the option to download the currently selected subset of genes in a recognisable format such as GFF. By providing this option, OrphanMine will be able to share the metadata generated during the analyses and be used in combination with other annotation data in packages such as Artemis. This prevents the loss of data and knowledge that can often occur during computational analyses.

4.3.1 Currently Available Resources for the Study of Lineage-Specific Genes

Several resources have been developed for the exploration of lineage-specific genes in bacterial genomes. These include the Orfanage (Siew *et al.*, 2004), CUPID (Mazumder *et al*, 2005), GeneQuiz (Andrade *et al.*, 1999), Indigo (Nitschke *et al.*, 1998) and the Neurogadgets Inc. Bioinformatics Web Service (Charlebois *et al.*, 2003). For various reasons, none of theses resources fit the OrphanMine requirements discussed in 4.3 above. In this section, I will provide a brief discussion of these resources and further highlight the motivation for the development of OrphanMine.

The Orfanage was developed specifically for the analysis of lineage-specific genes. Using the Orfanage it is possible to search for genes restricted to a user defined lineage, from the 85 genomes contained in the database. The parameters of the search cannot be defined and are as described in Siew & Fischer (2003a). The results of the search are not instantly available; instead they are e-mailed to the user. Further analysis of the results from the Orfanage was prevented by an error returned when I tried to access the data. CUPID provides a web interface to explore lineage-specific genes. However, the genomes contained in the database are limited to food and water based pathogenic species, thus preventing many analyses from taking place, for example, comparing a pathogenic strain of a species to a non-pathogenic strain. CUPID also fails to provide any genomic context to the results it displays, restricting easy investigation of the location of the genes within the genome. It does not provide files to download for further annotation, nor does it provide the option to automatically load data into annotation software such as Artemis. GeneQuiz provides lists of orphan genes. These genes are annotated with additional information, when available, such as functional and structural data. However, like many of these databases, the contents have not been regularly updated. In the case of GeneQuiz, the last update was in

February 2002. Both Indigo and the Neurogadgets Inc. Bioinformatics Web Service are, despite being published, no longer accessible.

Other less specialised databases are available. Examples include the Integrated Microbial Genomes (IMG) system at the JGI (Markowitz *et al.*, 2006) and TIGR's Comprehensive Microbial Resource (CMR) (Peterson *et al*, 2001). Both these resources provide similar types of analyses. The strength of them lies in the fact that they are regularly updated with the most recent sequence data. However, they tend to provide summary statistics for a genome (e.g. the percentage of hypothetical genes in each genome), rather than providing detailed analyses of lineage specific genes.

In contrast, the OrphanMine will provide the ability to create datasets of taxonomically restricted genes from a pool of genomes larger than those found in other specialised datasets. Unllike the majority of the other resources, it will provide extensive meta-data for each predicted coding region and will provide access to tools that will allow further annotation and will allow the selected dataset to be viewed in the context of the rest of the chromosome. To allow for further use of the data stored in OrphanMine, users will be able to download their datasets in commonly used file formats. The OrphanMine will also allow users to create their own datasets by altering the threshold cut-offs for determining a significant relationship (using either e-value or percent identity), therefore the users will not be limited to the parameters described in Wilson *et al.* (2005). Finally, the OrphanMine will utilise a method for ranking sequences according to different criteria, therefore providing the user with a prioritisied list of sequences for further analysis. The majority of these functions are not provided in the currently available tools discussed above and will be unique to OrphanMine.

4.4 Methodology

In this section, the methodology used whilst developing OrphanMine is discussed. During the development of OrphanMine, an evolutionary (or incremental) model was followed. This method periodically produced a version of OrphanMine that was increasingly complete over time. The first iteration was not to be viewed as the main objective but instead as a stepping stone in the continual development of the system. The second iteration took the existing system from the first iteration, evolved it further and integrated it with new requirements. At the end of this iteration, a significantly greater proportion of needs were fulfilled. In the case of OrphanMine, the first iteration was a database that only contained metadata describing genomes and orphan genes. The second iteration provided the possibility of creating custom orphan datasets by introducing a table to hold all predicted proteins and their associated metadata. This development or evolution will continue until an optimal solution to the requirements is achieved (unlikely to be reached as requirements are also often evolving and changing (Avison & Fitzgerald, 2003)).

Evolutionary development is characterised, not only by its iterative nature, but also by the evolutionary nature of the system's original creation (Orman, 1998). Therefore, the original design of OrphanMine was not a perfect solution to the user requirements, as it addressed only part of the required system. However, it was able to accommodate system changes. As the project progressed, more requirements were answered in the design. Several benefits were gained by using the evolutionary approach. Firstly, it provided quick results. The first implementation, although not a full solution, was developed more quickly than a full traditionally developed system. Secondly, changing requirements over time were expected and catered for.

4.5 System Prerequisites

4.5.1 Data Sources

The data stored in OrphanMine is obtained from a combination of the NCBI and the local output from QuickMine. The NCBI provides the proteome files ('.faa') necessary for running QuickMine. In addition, proteome table files ('.ptt'), proteome files in DNA format ('.ffn') and GenBank files, for all the bacterial genomes stored in OrphanMine, are obtained from the NCBI. The proteome table files list all of the proteins included in the '.faa' proteome files and displays the DNA co-ordinates for each of these proteins. The '.ffn' file is used to calculate the GC content of each predicted protein. GenBank files are required for the Artemis applet to display annotations correctly. Numerous output files from QuickMine are used in OrphanMine, of particular importance are the *overview.html files. These files are described in detail in Chapter 3. Low complexity values were generated using SEG (Wootton, 1993). SEG divides sequences into contrasting segments of low-complexity and high-complexity. Low-complexity segments, defined by the algorithm, represent simple sequences or compositionally-biased regions. Analyses requiring statistical software utilised 'R' (http://www.r-project.org/). R is a language and environment for statistical computing and graphics.

4.5.2 Formatting Data for Submission

The various data files require parsing before they can be entered into the OrphanMine database. Many of the scripts used to parse the data utilise Bio::Perl modules, particularly Bio::Seq, and Bio::SearchIO. The majority of the scripts used are small and were written for a single purpose. This allows flexibility in the use of the scripts and the way in which they are combined. It also accounts for the large number of scripts that constitute the pipeline. This modular style approach owes much to the incremental nature of the database design and implementation. Developing a structured system update pipeline at this stage would have been unnecessary due to the changing requirements over time. Work on designing such a system could take place in the future and would utilise many of the scripts currently used.

The update process, as it stands, can be divided into three stages. The first stage involves parsing through the BLAST reports generated during QuickMine. The second stage obtains data from the '.faa' files formatted in QuickMine and also the '.ptt' and '.ffn' files downloaded from the NCBI. The final stage utilises both QuickMine output and output from the previous two stages. In addition, it requires access to the 'R' statistics package and the manually created genome table file.

The final output comprises a text file for each table in the 'orphandb_v2' MySQL database. These text files can be loaded directly into the database. Additionally, all of the predicted proteins in the proteome files will have been formatted to include additional supplementary annotation in their header line. This includes the GC content, percentage low complexity, the number of genomes in the dataset containing a significant hit and the E-value and identity of the best match.

4.6 Database Design

This section discusses the processes and decisions taken in the design of the OrphanMine MySQL database. A relational database, simply defined, is a database made up of tables and columns that relate to one another. These relationships are based on a key value that is contained in a column.

4.6.1 Normalisation

An important aspect of relational database design is normalisation. The process of normalisation is performed to eliminate anomalies found in the data, thus leading to a more efficient and robust database structure. The degree of normalisation found in a database is defined by its normal form. In general, it is good design policy to have a database that conforms to 3rd normal form. Although a further four normal forms have been defined, they are rarely required. Below is a brief description of the criterion of the first three normal forms.

1st Normal Form: A database in first normal form must have an atomic value in each column, i.e., will only have one value per cell. Each column in a table must have a unique name. The table must have a set of values that uniquely identifies the row. This is known as the primary key of the table. No two rows can be identical and no repeating groups of data are allowed.

2nd Normal Form: A database is in second normal form when each table only stores data on a single entity. Each entity must be described by a primary key.

3rd Normal Form: 3rd normal form is concerned with transitive dependencies. A transitive dependency is a situation where a column exists that is not directly reliant on the primary key; instead the field is reliant on another field, which in turn is dependent on some other field.

When a database is in third normal form it must also have reached the criterion for first and second normal form. Throughout the design of OrphanMine, the aim was to conform to third normal form.

4.6.2 OrphanMine Primary Keys and Indexes

Both the names of the tables and the names of the fields within the tables were chosen to clearly describe the data that they contain. Each table has a unique identifier (the primary key), that is used internally in the database. The primary key is not a real world property and thus permits a change in the properties of an entity without affecting the identity of that entity. The primary key is generated automatically within MySQL by selecting the auto_increment command, when defining the primary key columns.

In addition to selecting columns to act as primary keys, indexes were defined in the tables. Relational databases, in particular MySQL, have the ability to query and sort vast amounts of information at great speeds. In order to achieve these speeds, MySQL, and other RDBMSs, make use of optimised data storage mechanisms, called indexes. An index allows the database server to create a representation of the indexed column, which it can search at great speed. They are particularly useful when searching for specific rows within a large table and they can also speed up table joins and aggregate functions (e.g. count()). In certain circumstances, an index can slow down processes, for example if there are too many indexes or if table data is being updated (Greenspan & Bulger, 2001). Indexes are automatically produced for primary keys. In addition, an index was defined for columns that would be searched against on a regular basis, for example 'orf_name'.

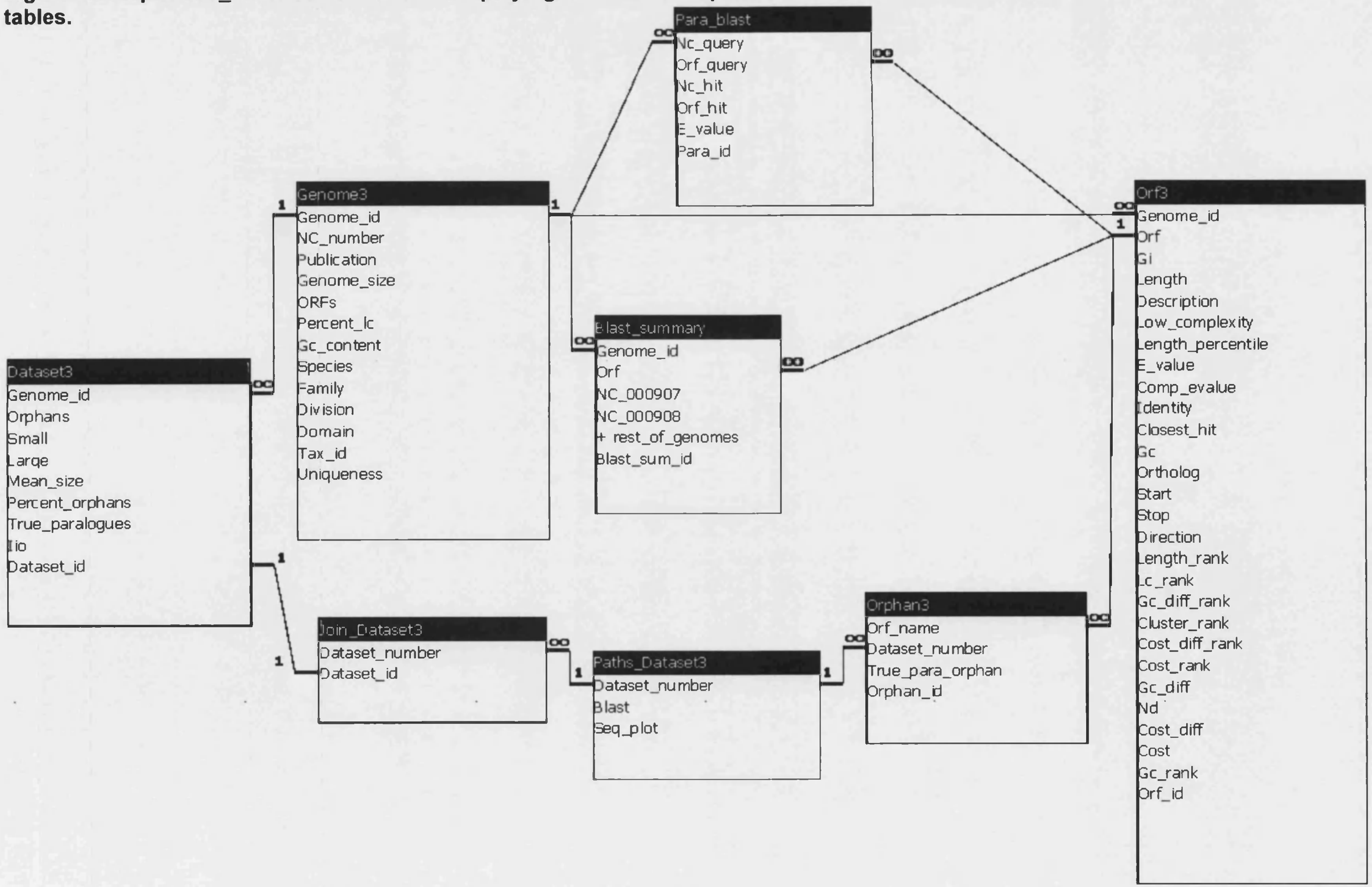
4.6.3 Public versus Private

During the course of my project, it became apparent that I would need to run additional analyses on the data stored in OrphanMine. The output of such analyses would need to be stored in OrphanMine to enable effective querying of the data. The analyses included performing RPS-BLAST on the orphan genes, performing TBLASTN on the orphan genes against the bacterial genomes in DNA format and BLASTing the orphan genes against UniProt (Apweiler *et al.*, 2004). However, the data used in these analyses did not need to be regularly updated, which is in contrast with the public version of the database. OrphanMine was intended to be a publicly accessible tool that contained complete sequenced bacterial genomes. Thus, regular updates are necessary. With the resources available, it would not have been practical, from the view of both time and data storage constraints, to perform the additional analyses noted above. As an alternative, the option is provided that allows the user to BLAST their sequence of interest against a selection of databases including UniProt and the genomes in DNA format. Hence, two OrphanMine models were developed, the private 'orphandb_test' and the publicly available 'orphandb_v2'. The schema for orphandb_v2 can be seen in Figure 4.1. The PHP interface for the two versions is essentially identical. The remainder of this chapter will focus on the design and functionality of the public version of OrphanMine (orphandb_v2).

4.6.4 OrphanMine Datasets

An important requirement, when designing OrphanMine, was the ability to support multiple orphan datasets. Currently, the database stores data on four different datasets. Dataset 1 (D1) contains data on all the orphans found when analysing all available genomes (currently 330). Dataset 3 (D3) contains data on all the orphans found when analysing the first available representative genome of a species (currently 247 genomes). Dataset 2 (D2) and Dataset 4 (D4) contain a subset of the orphans found in D1 and D3 respectively. These orphans are all 150 amino acids in length, or greater, and contain no regions of low complexity. The orphans and associated data vary between datasets. In contrast, genomic data and taxonomic information remains constant. Thus, it became necessary to build both a genome table and a dataset table. The genome table has one entry for each genome contained within the database. The dataset table has as many entries per genome as datasets that the genome is found in. An additional requirement was for the database to support the idea of custom dataset building. For this to be possible, data describing all predicted proteins in the bacterial genomes needed to be captured and stored. This is in contrast to only storing data on those predicted proteins described as being orphans in one or more of the pre-generated datasets. To accomplish this, two tables were required. One table contains all the predicted proteins and their associated data, the other contains the identifiers of the orphan genes and the dataset in which they are an orphan. The latter table is used when querying the pre-generated datasets, the former when creating a custom dataset.

Figure 4.1. orphandb_v2 database schema displaying the relationships between the different database tables.



4.7 Table Descriptions

The following sections detail the table structures used in OrphanMine and describe the fields found within the tables. Table 4.1 summarises this information, indicating the number of fields each table possesses, the primary key of each table and the indexed columns of each table. Appendix 4.1 contains an SQL script detailing the tables and fields contained within the OrphanMine database.

Table 4.1. Summary of OrphanMine MySQL tables

Table Name	Number of Fields	Primary Key	Indexed Columns
Genome3	13	Genome_id	NC_number
Dataset3	9	Dataset_id	Genome_id
Orf3	28	Orf_id	Orf Gi Genome_id
Orphan3	4	Orphan_id	Orf_name
Blast_summary	Variable*	Blast_summ_id	Genome_id Orf
Para_blast	6	Para_id	NC_query
Paths_dataset3	3	Dataset_number	-
Join_dataset3	2	Dataset_id	-

* Dependent on the number of genomes stored in OrphanMine. Currently there are 333 fields (3 + 330 genomes).

4.7.1 Genome3

The genome table contains data describing genomic features that remain constant, regardless of the dataset being viewed. The data is obtained from a variety of sources and constructed manually prior to entry in the table. The majority of the fields come from a file downloaded from the NCBI (http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi). The taxonomic information which is not present in the downloaded file can be found on the Entrez Genome Project page for the relevant genome. The number of predicted proteins and the overall percentage low complexity of the genome are calculated using the script *make_seg.pl* on the output from the programme SEG. The genome_id field is the primary key (and hence is incremented automatically by the database), uniquely defining a set of details about a genome. Other identifiers, such as RefSeq identifiers, are not suitable as each bacterial chromosome is given a unique RefSeq ID. A bacterial species such as *Vibrio cholerae,* which possesses two chromosomes, has two RefSeq identifiers (NC_002505, NC_002506); hence one genome would have two entries in the genome table. Therefore, it was necessary to define a unique Genome_id for use within the OrphanMine system. Genome_id is an integer value and is therefore capable of creating table joins more efficiently than a text value such as the RefSeq. In addition to the primary key, the NC_number column is indexed, as it is common to search on these values.

4.7.2 Dataset3

The dataset table contains data describing features that vary according to the dataset being viewed. The majority of the fields refer to the orphan number for the genome of interest in a particular dataset. Additionally, the table contains the isolation index of an organism (IIO) (Fukuchi & Nishikawa, 2004) for the genome. This value is dataset dependent, as different datasets contain different genomes and hence the isolation of a genome compared with the other genomes in the dataset will vary. The values for the different fields are calculated in the QuickMine and OrphanMine Perl scripts. Dataset_id is the primary key. Genome_id is the foreign key referencing Genome_id in Genome3. Genome_id is indexed in Dataset3 to increase the efficiency of table joins during queries.

4.7.3 Orf3

Orf3 is the most important table in OrphanMine as it is required for all three of the methods used by the PHP interface to interrogate the database. It contains data on every predicted protein in all the genomes contained within the database, currently this stands at 972526 predicted genes from 330 genomes. There are 28 fields in Orf3; the values for the different fields are calculated in the QuickMine and OrphanMine Perl scripts. In addition to fields that describe the sequence directly, such as GC, length and low complexity, the Orf3 table contains data necessary for the functionality of the Artemis and CGView applets. These include the start and stop co-ordinates, the direction of the predicted gene and the number of genomes in the database that contain potential orthologues. Also included are the values for the different ranking criteria (length, percent low complexity, GC content, neighbourhood distribution and metabolic cost) that enable the PHP scripts to rank the predicted proteins according to user defined criteria. Additionally, the table includes the E-value and percent identity for the best hit to another predicted protein. This allows users to create their own custom orphan datasets using a cut-off threshold of their choice. The GI number is included in Orf3 as an additional identifier. It is used primarily when using *fastacmd* to retrieve protein sequences. Orf_id is the primary key. Genome_id is the foreign key referencing Genome_id in Genome3. Orf, Gi and Genome_id are indexed to increase the efficiency of table joins during queries and sequence retrieval.

4.7.4 Orphan3

The Orphan table is responsible for containing a list of predicted proteins and the dataset in which they are found to be orphans. Hence, the same predicted protein may be found in the table more than once, as it may be found to be an orphan in more than one dataset. Additionally, the number of true orphan paralogues associated with the orphan is recorded. Orphan_id is the primary key; Orf_name is the foreign key that references Orf in Orf3. Orf_name is also indexed.

4.7.5 Blast summary

The table Blast_summary contains a representation of the BLAST overview files produced by QuickMine. The overview files are formatted by the OrphanMine Perl scripts so that they can be entered into the database. The table contains three standard columns. These contain the Genome_id, the Orf_name and the Blast_summ_id. Genome_id is a foreign key that references Genome_id in Genome3; Orf_name is a foreign key that references Orf in Orf3. Both these columns are indexed. Blast_summ_id is the primary key. In addition to these columns, there is a column for each of the genomes contained in the database. Therefore, the number of fields in this table is dependent on the number of complete sequenced bacterial genomes. For each element in this table (effectively each predicted protein), there is a numerical value in each of the genome based columns, representing the number of potential homologues to the predicted protein, in that particular genome. These values are used to generate the user defined lists of lineage-specific genes.

4.7.6 Para blast

The Para_blast table consists of six fields. The purpose of the table is to allow the user to view the paralogues associated with the orphan genes. Not all the paralogous genes are also orphans. The data for this table is initially generated by QuickMine and formatted by the OrphanMine Perl scripts. Para_id is the primary key. NC_query is indexed to allow more efficient searching.

4.7.7 Paths dataset3

Paths_dataset3 contains the information required to find the relevant data files for the different datasets. The PHP scripts need to be directed to files such as the BLAST database. Dataset_number is the primary key.

4.7.8 Join dataset3

The Join_dataset3 table was responsible for creating a normalised relationship between Dataset3 and Orphan3. It stores Dataset_number as a foreign key and Dataset_id as a primary key. This table is an artefact of the evolutionary methodology. In future iterations of OrphanMine, the table could be removed and an additional column representing Dataset_number could be added to Dataset3.

4.8 The Web Interface and PHP Query Pages

An elegant database is irrelevant if the end user cannot interact with it in a manner that maximises both usability and utility. For this reason, particular attention was paid to designing an intuitive user interface, which felt both logical and natural to the system users. The interface, whilst maintaining a simple design and therefore faster download speeds, provides both meaningful links and consistent navigation. The page content is controlled by PHP scripts embedded within the HTML. The PHP is responsible for performing the correct actions from the user input, querying the database, and displaying query results in the required manner. When the user submits the HTML form, having filled in the form elements, the browser sends an HTTP request to the web server. In OrphanMine, the user input is generally processed using the POST method.

A detailed description of the functionality of the PHP pages and their associated screenshots can be found in Appendix 4.2. The majority of OrphanMine's web interface is written in PHP. However, Perl CGI scripts had previously been written to perform BLAST searches (*blast.cgi*) and to interrogate the MySQL database directly (*orphan_sql.cgi*), therefore these scripts were utilised (see Appendix 4.2).

In some cases, a web page is not always a satisfactory user interface. The form elements defined for HTML are limited. An alternative is to use applets in web pages to make them look and function more acceptably (Coulouris *et al.*, 2001). However, there is a consequent increase in download time. In the case of OrphanMine, a standard

HTML interface was deemed appropriate, primarily as this would provide the majority of the functions required. In addition, OrphanMine utilises two Java applications; Artemis (Java Webstart) (Rutherford *et al.*, 2000) and CGView (Java Applet) (Stothard & Wishart, 2005).

4.8.1 Artemis Webstart

Using Java Web Start technology, standalone Java software applications can be deployed over the network. It enables developers to deploy full-featured applications to end-users by making the applications available on a standard web server. Unlike Java applets, Webstart applications do not run inside the browser. OrphanMine utilises Artemis through the use of Java Webstart. Artemis is a genome annotation tool created at the Sanger centre. It allows the visualisation of sequence features and the results of analyses within the context of the genome. The tool loads sequence files in EMBL, GenBank and FASTA format. Once the sequence is loaded, it is possible to annotate the sequence with additional features. This can be done by loading in annotation files in EMBL, GenBank or GFF format. Alternatively, the user can annotate the sequence manually.

OrphanMine users can view their datasets of predicted proteins in Artemis simply by clicking on the link provided. OrphanMine generates a JNLP (Java Network Launching Protocol) file. The JNLP file is read by the user's Java Webstart engine, resulting in Artemis being loaded. The Artemis software is hosted at the Sanger Centre. The JNLP file also contains information necessary for the generation of relevant annotation files. The files are automatically created by OrphanMine when Artemis starts and are loaded into the Artemis viewer. The Artemis tool, coupled with the Java Webstart technology, allows users to view their datasets in a genomic context in a quick and simple fashion. It also permits users to study annotations obtained from other resources and compare them to the annotations in OrphanMine, thus enabling the community to efficiently interrogate their data, assisting in the formation of scientific conclusions.

4.8.2 CGView Applet

An applet is a software component that runs in the context of another programme. In the case of OrphanMine, this is a web browser. As an applet executes on the client side, it can provide functionality beyond the default capabilities of the web browser. CGView, or Circular Genome Viewer, exists as both a stand-alone application and as a

Java applet. The primary purpose of the applet is to visualise dynamically generated sequence features. In the case of OrphanMine, bacterial chromosomes are loaded and features selected by the user, for example, orphan genes, are added to the map. CGView permits the user to zoom in on regions of interest and also to move the position of the viewing window in relation to the chromosome. Additionally, OrphanMine uses CGView to indicate how distributed a gene is amongst the other genomes included in the dataset. Therefore, each gene in a given genome has a pink bar associated with it. The height of the pink bar indicates the level of distribution of the gene. CGView is freely available from http://wishart.biology.ualberta.ca/cgview/index.html.

4.8.3 Annotation File Formats

The ability to download data in a variety of formats is provided by OrphanMine. Amino acid sequences of proteins of interest can be downloaded in FASTA format. These sequences have a modified header line, providing the user with descriptive metadata for each sequence. This metadata includes GC content, the number of genomes with a match to the sequence, the taxonomic uniqueness of the genome, the E-value at which the sequence would be deemed an orphan and the best match to the sequence. Additionally, dataset dependent annotation files can be obtained for each genome. These files can be downloaded in simple tab-delimited format or in GFF format. Tab-delimited files are commonly used and can be loaded easily into programmes, such as Microsoft Excel. GFF (General Feature Format) files are also relatively simple. However, they provide a structured framework for the annotation of sequence features, thus encouraging the development of software to utilise the files. The current version, GFF3, allows for greater flexibility in sequence annotation, whilst maintaining the file structure. Many software projects now utilise GFF files, for example GFF2PS (Abril & Guigo, 2000) and Artemis. The GFF files generated by OrphanMine are GFF3 and have been validated as keeping to the conventions of the format by the online validation system (http://dev.wormbase.org/db/validate_gff3/validate_gff3_online).

4.8.4 OrphanMine 'Help' Pages

Every PHP page accessed in OrphanMine displays a link to a Help file. The help file is page specific, i.e., it gives information specifically associated with the page that the user is currently viewing, thus providing useful and relevant information to users and in doing so, increasing the level of usability of the system. The help appears in a pop-up window, initiated by JavaScript.

4.8.5 PHP Database Queries

Information displayed in OrphanMine is obtained by performing a query of the 'orphandb_v2' database, through the PHP script. In order to do this, the script must open a connection to MySQL and select to use the database 'orphandb_v2', defined as the constant 'DB'. The commands used are shown below:

```
$db = mysql_connect(HOST.":".PORT,USER,PASS);
mysql_select_db(DB);
```

OrphanMine uses the file *db.php* to define the information required to connect to the MySQL database. Once a connection is open and the database is selected, the database can be interrogated using SQL (standard query language) and the PHP function mysql_query. The example shown below would query the database for the information that is displayed when viewing *orphan.php*:

```
$orphan = mysql_query("SELECT Orf_name, True_para_orphan, NC_number,
Length, Description, Orf_id, truncate(Low_complexity,2), gc, gi
                    FROM orphan3, genome3, orf3
                    WHERE orf3.genome_id = $genome_id and orf3.Genome_id
=    genome3.Genome_id    and    orf3.orf    =    orphan3.orf_name    and
orphan3.Dataset_number = $dataset",$db)
                    or die (mysql_error());
```

In order to get a meaningful result from the query, the function mysql_fetch_array is used. As the name of the function suggests, this retrieves the results of the query and enters them into an array. This array can then be accessed and the results displayed in HTML. Most of the output in OrphanMine is displayed in HTML tables. This provides an easy method of presenting the data in a neat and uniform style.

Processes similar to that described above will occur repeatedly as a user navigates through the system. It is these functions that enable OrphanMine to operate as a knowledge base.

4.9 OrphanMine Evaluation

The following sections discuss the process of evaluation, the evaluation methods used and the results of those evaluations. Evaluation has three main goals: to assess the extent of the systems functionality, to assess the effect of the interface on the user and to identify any specific problems with the system (Dix *et al.*,1993). Due to the evolutionary methodology followed during the development of OrphanMine, informal evaluation was performed continuously throughout the project. However, once it was felt that the resulting system met the majority of the requirements specified at the start of the project, formal evaluation techniques were utilised. The first technique evaluated the design, the second evaluated the implementation.

4.9.1 Evaluating the Design

To evaluate the design, a heuristic method was used. In this approach, a set of usability criteria or heuristics were identified and the design examined for instances where this criteria was violated. The goal of the heuristic evaluation was effectively to debug the design. The approach is simple and relatively fast. As specific criteria are used to guide the evaluation, the process is not subjective. However, in order to make the most of this type of evaluation more than one evaluator, assessing the design independently, is necessary, as a single evaluator is liable to miss problems (Dix *et al.*, 1993). Paul Swift and I performed the design evaluation of OrphanMine. The ten heuristics used in the evaluation can be found in Appendix 4.3.

4.9.2 Results obtained from Design Evaluation

In general, it was found that the system was natural and logical. The tool was found to be intuitive to use, although it was felt that a user would have to invest time to get the best from the system. This learning curve is largely a result of the intrinsic complexity of the concepts involved. The help system was described as extensive and highly specific. It was felt that the system would certainly be of benefit to researchers interested in analysing lineage-specific genes. In addition to these general thoughts, there were several points raised regarding the system and improvements were suggested. Table 4.2 displays a summary of the responses to the heuristic evaluation and indicates whether the feedback has been implemented.

Table 4.2. Feedback obtained from Heuristic Evaluation

Heuristic Evaluation Response	Implemented?	How or Why?
When working with custom datasets, the menu bar reports information specific to Orphan Dataset 1. This needs to be changed to show that the user is working on a custom dataset or a TRG dataset.	YES	Now prints 'Custom dataset' or 'TRG dataset' in the menu bar.
Once a custom dataset has been generated, the parameters used to create the dataset are not displayed. This could lead the user to forget what they are working with. Add a box that contains a list of the parameters used.	YES	Used dynamic HTML to produce a box displaying parameters when the mouse is moved over the dataset name in the menu bar
When viewing OrphanMine using monitors at low resolution, not all the information fits (horizontally) on the screen. Modify these pages so that a horizontal navigation bar is unnecessary.	NO	The layout is appropriate for the majority of monitors. To cater for low resolutions would lead to the screen becoming too cluttered.
When navigating the site, users have to make use of the back button on the browser. Often this causes the web browser to ask the user to refresh the page. It is particularly noticeable when using Microsoft Internet Explorer.	NO	The problem is associated with the use of forms and the POST method. Changing this would mean large scale changes to the PHP scripts. It is something that can be looked at in the future.
In some cases, the word orphans and predicted proteins are used interchangeably. These terms should be distinct to avoid confusion.	YES	The terms have been checked and changed where appropriate.
Although the page-specific help is extensive, it would be useful to have a few lines of information at the top of most pages. This would give a better indication to the user of what it is they are looking at.	YES	Explanations have been added to a number of pages to help guide the user.
It would be useful if all the page specific help files were also concatenated into one large help file. Currently there is no 'Help' index and no method for navigating from one 'Help' page to another.	YES	The help pages have been merged and placed in *all_help.php*. Users can reach this page through *faq.php*.
On the search page, the submit button is labelled 'GO'. This could be confused with the GO – Gene Ontology. The button should be renamed.	YES	The button has been renamed 'Submit'.
The 'Pretty' QuickMine matrix option seems superfluous to requirements. Takes a long time to download and occasionally seems to generate errors. Possibly remove the option.	NO	Have decided to leave the option in place. However, have provided more warnings to inform the user of the long load times associated.

4.9.3 Evaluating the Implementation

To evaluate the implementation, an observational technique was used, called 'Think aloud' (Dix *et al.*, 1993). This method involved providing the users with a set of pre-determined tasks. The user's actions were watched and recorded. In addition to this observation, the users were asked to elaborate their actions by talking aloud and describing what they believe is happening and what they are trying to achieve. In the evaluation of OrphanMine, a variant on the 'think aloud' methodology was used known as co-operative evaluation (Dix *et al.*, 1993). Users were asked questions and were able to ask questions. This relaxed version of the process provided several advantages; the process was less constrained, the user was encouraged to criticise the system and points of confusion could be clarified at the time they occurred.

The users evaluated in this manner were all scientific researchers but with varying levels of experience at dealing with databases, such as OrphanMine. The evaluation took place in the users working environment, i.e., on their own computer in their research laboratory. This allowed for the evaluation of the interaction as it occurs in actual use. It is likely, however, that the users were still influenced by my presence, for example, failing to utilise the system's help functions. The tasks that the users were asked to complete are shown in Appendix 4.4.

4.9.4 Results obtained from Implementation Evaluation

Generally, users were able to navigate through the system easily and were able to complete the specified tasks. Issues that did occur were often due to limited knowledge with regards to the specific subject of lineage-specific genes. These problems were solved by looking at the options available on screen and using their initiative to choose the relevant route or by asking for my assistance. In my absence, it is assumed users would be forced to use their initiative or make use of the help system. Additional text has been added to several pages to help users navigate, without having to utilise the help pages. During the evaluation, the help system was largely overlooked by the users. Whilst this was anticipated, the degree to which it was ignored was surprising. Having spoken to users specifically about this point, it appears that this is down to habit rather than poor presentation by OrphanMine. However, in an attempt to raise awareness the help system has been highlighted on the OrphanMine home page. Table 4.3 displays a summary of the responses to the evaluation and indicates whether the feedback has been implemented.

The system was praised for its general presentation. By keeping the individual pages as clear as possible, users were not intimidated by the volume of data. The use of colour on the white background was also praised, again because it prevented the screen from appearing too cluttered.

Of particular interest to users was the idea of a QIPP web service. QIPP (Quality Index for Predicted Proteins) is an index used to score potential coding regions and is calculated by analysing various sequence characteristics (length, low complexity, GC content, average amino acid cost and neighbourhood distribution). The development of QIPP will be discussed in more detail in Chapter 5. A QIPP web service would enable users to submit their own annotation files to the server and have output generated, scoring each predicted coding region. With this in mind, Web QIPP was developed. Web QIPP (www.genomics.ceh.ac.uk/orphan_mine/qipp_web.php) provides an interface to the *qipp.pl* Perl script. This script calculates QIPP scores for coding regions found in a GenBank file. The user submitted file must be in GenBank format and must contain a sufficient number of coding regions on which QIPP can be calculated. This version of QIPP is entirely homology independent and so does not calculate neighbourhood distribution. Once the scores have been calculated, the output is printed to screen in either GFF or tab-delimited format. The format can be selected by the user. Screenshots of Web QIPP can be seen in Appendix 4.2.

Table 4.3. Feedback obtained from Implementation Evaluation

Implementation Evaluation Response	Implemented?	How or Why?
When searching for OrphanMine using the Google search engine, the user is directed to *orphan_home.php* instead of *orphanmine.php*. Want to re-direct to *orphanmine.php*.	YES	Changed the name *orphan_home.php* to *orphan_datasets.php*. *orphan_home.php* now automatically redirects to *orphanmine.php*
When ordering the genomes by publication date, it would be useful to see the date, or have a column to indicate they have been sorted.	YES	The database does not currently store the publication dates of original genome papers. Instead, a column has been added that indicates what has been used to order the genomes.
It is not immediately obvious to users that columns on the search page are sortable.	YES	Added a line of text highlighting this property.
When BLASTing a pre-selected sequence, it would be useful to carry the ID of the sequence to the BLAST page. This would make it easier to infer information from the BLAST report.	YES	The orf_name is passed to *blast.cgi*. It is printed to screen before the BLAST is performed and is also printed in the output page.
Not clear to users what method the system uses to rank predicted proteins. This can lead to confusion.	YES	Added line of text stating that the QIPP method is used to rank the predicted proteins. Also provided a link to the relevant publication.
On *customise.php*, clarify the type of number that the user should enter as an E-value.	YES	Put the text '10-' in front of the E-value text box. Indicates an integer is required rather than a decimal number.
Make the ERROR message on *customise.php* larger and more noticeable.	YES	Changed the font-size of the error message and made it bold.
Make it clear how to download sequences from the trolley. Users often ignore the checkbox and get an error message.	YES	Added a line of text to clarify the process.
Location-specific help pages are under used. Users appeared more likely to look at FAQs.	N/A	This is largely due to user habit. However, have highlighted the presence of the help pages in *orphanmine.php* and created *help_intro.php*.

4.10 Discussion & Conclusion

In this section I will discuss whether OrphanMine has succeeded in meeting the requirements set out at the start of the project. Possible enhancements to the system that could improve its functionality and usability are also discussed.

4.10.1 Has OrphanMine met the outlined requirements?

As the primary user of OrphanMine and the sole developer, I have ensured that OrphanMine has met the level of functionality that I required. The current version of OrphanMine meets the requirements and provides a system architecture that enables enhancements and modifications to be made. Generally, enhancements will be easily implemented. The tool provides an interface to the user that is both aesthetically pleasing and minimalist. The various functions are located easily by the users and are performed in a logical and natural manner. These features provide the user with a tool that enables knowledge sharing within the research community.

4.10.2 Future Enhancements to the OrphanMine system

The next stage in the evolution of OrphanMine is to facilitate community annotation of the lineage-specific genes. To create such a function restricted to OrphanMine would be relatively straightforward. However, I believe this would be short sighted and limited. Instead, universal gene function annotation data is required. OrphanMine could implement links to this data.

Currently, there is no universally accepted method or tool for community annotation, thus this demand has yet to be met. It is unlikely that scientists are going to be enthused by the idea of annotating a gene in one database only to find the annotation is missing in a different database. Therefore, a central repository of annotation data is needed. External databases, such as OrphanMine, could then link to this one resource. One idea is to create a gene function wiki (Wang, 2006). The majority of scientists are familiar with the idea of a wiki and so such a tool would not be intimidating to approach.

The NCBI has introduced GeneRIF (Gene Reference into Function) into their Entrez Gene Database (Maglott *et al.*, 2007). GeneRIFs are always associated with specific entries in the Entrez Gene database and each GeneRIF has a pointer to the PubMed ID of the publication, providing evidence for the statement made by the GeneRIF

(Mitchell *et al.*, 2003). Whilst GeneRIFs do provide a service to the research community, it is generally the NCBI indexers that produce the GeneRIFs, rather than the wider biological community. Additionally, these annotations are restricted to the Gene database. An integrated and comprehensive resource for this genomic data is required. Whilst a wiki may not be perfect, for example, the idea of a random scientist editing the functional annotation of a gene will not inspire confidence, it does provide a stepping stone towards obtaining the annotation required. Another method that could be considered is the use of social-tagging as a method of annotation. The idea of social tagging, in essence, is that people add free-text tags to their content, in this case genes, and where people use the same terms, their content is linked.

A major issue preventing the use of such resources is the lack of a universal identifier for a gene. Different databases use different identifiers and update at different times. Hence, a universal annotation page for a particular gene would have the difficulty of maintaining mappings with the EBI and the NCBI, in addition to the multitude of smaller databases such as OrphanMine. This is a major issue that needs to be resolved before any community driven gene annotation project can truly succeed.

A different issue relevant to the future of OrphanMine is the exponential rise in genomic data. An initiative led by Rick Stevens and Eddy Rubin aims to produce draft genome sequences for all prokaryote type strains (Field *et al.*, 2007a). Currently, there are over 300 complete bacterial genomes, this project alone will add several thousand more genomes to that figure. As more genomes are sequenced, the updating times of OrphanMine will increase. In addition, the amount of memory required to store the data will also rise. Whilst the interface to OrphanMine and the design of orphandb_v2 will be able to manage this increase, there will come a point where the infrastructure at CEH Oxford will no longer be suitable. In order to secure the long term future of OrphanMine, it may be necessary to move the location of the data to somewhere more suitable. Alternatively, collaboration between OrphanMine and a larger biological database such as the IMG at the JGI (Markowitz *et al.*, 2006) could see the OrphanMine interface over the top of the JGI data. Thus, continuing to provide the majority of the functionality of OrphanMine, without the danger posed by the volume of data.

4.10.3 Conclusions

During the creation of OrphanMine, I developed a scoring method that allows the prioritisation of sequences, according to certain qualities to ascertain the likelihood of the predicted proteins being real. In doing so, creating a list of prioritised genes for experimental characterisation. The data stored in OrphanMine is open to the public and easily downloadable in a variety of formats. Of particular interest is the use of verified GFF format (version 3). Providing the data in this format allows users to view the data transparently, i.e., they can load it into software and assess the quality of the data themselves. This property is much needed but is very rarely provided. Hence it is clear, just from these two examples, that OrphanMine was a worthwhile endeavour.

With the volume of sequence data increasing rapidly, there is a need to develop OrphanMine further. Such development could take the form of providing web services such as WebQIPP. This allows the user to enter their sequence data (in GenBank format) into the system and calculate QIPP scores on this data before printing the output in GFF. This work is done ‘on the fly’ due to the method being independent of homology, thus data storage is not an issue. Whilst there are clearly challenges associated with the future of OrphanMine in its current guise, there is little doubt of the benefits it can offer the research community. The purpose of OrphanMine, as clearly described, was to make a much needed initial step forward in working on the demands made in the Roberts Report. This target has been achieved.

CHAPTER 5

Large-scale Comparative Genomic Ranking of Taxonomically Restricted Genes (TRGs) in Bacterial and Archaeal Genomes

Gareth A. Wilson, Edward J. Feil, Andrew K. Lilley and Dawn Field
(2007)
PLoS-One
2(3): e324. doi:10.1371/journal.pone.0000324

5.1 Overview

Lineage-specific or taxonomically restricted genes (TRGs), especially those which are species and strain-specific, are of special interest because they are expected to play a role in defining exclusive ecological adaptations to particular niches. Despite this, they are relatively poorly studied and little understood, in large part because many are still orphans or only have homologues in very closely related isolates. This lack of homology confounds attempts to establish the likelihood that a hypothetical gene is expressed and, if so, to determine the putative function of the protein.

We have developed "QIPP" ("Quality Index for Predicted Proteins"), an index that scores the 'quality' of a protein based on non-homology-based criteria. QIPP can be used to assign a value between zero and one to any protein based on comparing its features to other proteins in a given genome. We have used QIPP to rank the predicted proteins in the proteomes of Bacteria and Archaea. This ranking reveals that there is a large amount of variation in QIPP scores and identifies many high-scoring orphans as potentially 'authentic' (expressed) orphans. There are significant differences in the distributions of QIPP scores between orphan and non-orphan genes for many genomes and a trend for less well-conserved genes to have lower QIPP scores.

The implication of this work is that QIPP scores can be used to further annotate predicted proteins with information that is independent of homology. Such information can be used to prioritise candidates for further analysis. Data generated for this study can be found in the *OrphanMine* at http://www.genomics.ceh.ac.uk/orphan_mine.

5.2 Introduction

The availability of hundreds of complete bacterial genome sequences has made it possible to explore how the evolutionary diversification of gene content reflects the ecological needs and opportunities of different taxa. It is well known that the gene content of bacterial and archaeal genomes can vary widely and that only a very few genes are truly universal (Tatusov *et al.*, 2003, Charlebois & Doolittle, 2004 and Ciccarelli *et al.*, 2006). As a consequence, genes can differ significantly in their taxonomic distributions, with more broadly conserved genes having 'housekeeping' functions and less conserved genes being responsible for the phenotypic differences observed between organisms. Lineage-specific, or "taxonomically restricted" genes (TRGs), are defined as being exclusively restricted to a particular taxonomic group (Wilson *et al.*, 2005). In such a framework, genes may be TRGs at any taxonomic level (*i.e.* domain-, family, genus-, species- or strain-specific). TRGs at the species and strain-levels are of most interest in the search for genotypes which help define exclusive ecological adaptations to particular niches.

The study of narrowly distributed TRG's is confounded by the fact that many are short, repetitive or have unusual A+T contents (Daubin & Ochman, 2004a), and the assumption that many such short coding sequences (CDS) represent annotation errors (Skovgaard *et al.*, 2001). Over-annotation of genomes, resulting in an excess of small predicted proteins, is clearly evident in certain genomes (*e.g.* the initial annotation of *Aeropyrum pernix* (Kawarabayasi *et al.*, 1999)) and is proposed to be an unfortunate feature of many genomic annotations (Skovgaard *et al.*, 2001, Fukuchi & Nishikawa, 2004 and Ussery & Hallin, 2004). This overannotation could mask intergenic regions containing small non-coding RNAs. It is also possible that many TRGs remain 'orphaned' for no other reason than the sampling bias in public genome databases (Siew & Fischer, 2003a). It is well-known that the current collection is highly biased towards certain organisms (most notably pathogens, γ-Proteobacteria, and Firmicutes) (Martiny & Field, 2005). This results in the trend that taxonomic isolation is correlated with an increased percentage of orphans (Fukuchi & Nishikawa, 2004). It is therefore expected that homologues for many orphan predicted proteins, in taxonomically isolated lineages that lack close relatives in genomic databases, will be found once the taxonomic gaps in the genomic database begin to be filled (Siew & Fischer, 2003a).

Despite potential errors in our current estimation of the numbers and identities of narrowly distributed TRGs, there is growing evidence that many, including those that are currently orphaned, are of biological significance. Hence, there is a growing need to untangle erroneous CDS from authentic species- and strain-level TRGs (Alimi *et al.*, 2000, Kolker *et al.*, 2004 and Shmuely *et al.*, 2004). Dispersed examples of the latter are most frequently found as the result of in depth *in silico* (Daubin & Ochman, 2004a) or empirical studies (Alimi *et al.*, 2000) of a particular organism or small group of organisms. Increasingly, examples are being identified as the result of whole genome sequencing (Shmuely *et al.*, 2004). One example to come from complete genome sequencing is the TCP virulence locus of *Vibrio cholerae* Tor N16961. Once a cluster of largely orphaned CDS, a homologous region has now been found in the squid symbiont *Vibrio fischeri* (Ruby *et al.*, 2005). The TCP genes code for the toxin co-regulated pili in *V. cholerae* and serve as its critical intestinal colonisation factor, providing the receptor for entry of the temperate filamentous phage CTX$^{\Phi}$, which contains the cholera toxin genes, *ctxAB* (Waldor & Mekalanos, 1996), into the cell (Manning, 1997). Likewise, the sequencing of many genomes is confirming the presence of many strain-specific genes which form the "pan-genome" of many species (Tettelin *et al.*, 2005 and Medini *et al.*, 2005).

Given the potential significance of orphaned and narrow-range TRGs and the confounding sources of error associated with currently annotated genomes, it is clear that a reliable objective measure of the potential 'quality' of a given CDS would be useful. This could be used to prioritise it, either as a candidate for further characterisation or as an error.

There are several methods that could be used to rank and prioritise CDS for further analysis. Previously such analyses and methods have focussed on the degree of conservation to a particular CDS, in other genomes. The greater the number of species a homologue is found in, the higher the rank of the CDS (Galperin & Koonin, 2004). A project called GTPS (Gene Trek in Prokaryote Space) aims to assign a degree of reliability to all predicted protein-coding genes in bacterial and archaeal genomes held by the INSDC (International Nucleotide Sequence Database Collaboration) (Kosuge *et al.*, 2006). This method grades predicted coding regions according to the results from a number of, largely homology based, analyses. However, GTPS does not provide a quantitative measure and provides no means for ranking CDS in the absence of homology.

Gene prediction programmes such as Glimmer (Salzberg *et al*, 1998), calculate a score based on the calculated probability of an ORF being a gene. This score could be used to provide a rank to CDS within a genome. Programmes designed to locate pathogenicity islands utilise criteria such as dinucletide bias and GC content (in addition to non-quantitative criteria, e.g. mobility genes) (Hsiao *et al.*, 2005). However, there is currently no explicit method for scoring and ranking CDS in the absence of homology. Motivated by this requirement, and with a specific focus on orphans and narrow-range TRGs, we have devised a scoring system that allows the 'ranking' of predicted proteins based on a variety of features, reflecting the likelihood that a given CDS encodes a protein.

We previously reported that the absolute number of single-copy TRGs from the complete and published genomes of Bacteria and Archaea is increasing (Wilson *et al.*, 2005). The most phylogenetically and ecologically unique species contribute the most unique genes, in part due to undersampling of these genetic lineages (Wilson *et al.*, 2005). For that study we generated two datasets. The first contained all orphans as defined by BLAST (using a threshold of 10^{-3}), the second applied an arbitrary length cut-off of $\geq$ 150 amino acids and excluded all CDS with low complexity (highly repetitive) regions to remove likely CDS enriched in artefacts. The method of scoring CDS described here extends this 'selective filtering' approach and is called the 'Quality Index for Predicted Proteins' (QIPP). We describe the use of QIPP as it is applied to the reanalysis of this dataset, based on the inclusion of five criteria selected for their presumed ability to detect purifying selection and CDS which are unlikely to occur by chance alone. These are length (Skovgaard *et al.*, 2001), percentage low complexity (a measure of the degree of repetition) (Altschul *et al.*, 1994), difference in G+C composition of sequence and genome (Navarre *et al.*, 2006), average amino acid cost (Akashi & Gojobori, 2002 and Heizer *et al.*, 2006) and neighbourhood distribution (ND) (Zheng *et al.*, 2005).

5.3 Results

5.3.1 The orphan and non-orphan components of many proteomes have different overall characteristics

To examine whether orphaned CDS, which are expected to be on average smaller (Skovgaard *et al.*, 2001) and more A+T rich (Daubin & Ochman, 2004a and Yin & Fischer, 2006) have significantly different QIPP scores than non-orphans, we re-examined our original dataset (Wilson *et al.*, 2005). QIPP scores were calculated for each protein in this dataset of 122 proteomes (Wilson *et al.*, 2005) as described in the Materials & Methods (5.5.2). In total, the distributions of all five criteria (length, low complexity, G+C content, amino acid cost and neighbourhood distribution (Table 5.1)) differ significantly between orphans and non-orphans in 61 of the 122 species examined ($p < 0.05$, Mann-Whitney). 3 or more criteria are significant in 117/122 species. Four of the remaining five species contained fewer than 10 orphans, and when all such genomes (n=6) were excluded 115 of the remaining 116 species had orphans that differed significantly from the non-orphans for three or more criteria. The strikingly different values for *Escherichia coli* K12 can be seen in Figure 5.1 as an example of these trends. The distribution of the QIPP scores for orphan and non-orphan TRG's were found to be significantly different for 119 of the 122 genomes ($p < 0.05$, Mann-Whitney). The remaining three genomes contained 2 or less orphans and thus could not provide significant discriminatory power. Overall, the QIPP scores for all orphan (mean = 0.38, +0.14) and non-orphan (mean = 0.54, +0.14) TRG's were significantly different (p = 0.000, Mann-Whitney). These results confirm that the criteria used for the QIPP scores can reliably distinguish between "orphan-like" (less well conserved) and "non-orphan-like" (more widely conserved) genes.

Figure 5.1. Distributions of orphans and non-orphans in *E. coli* K12. The predicted proteins in *E. coli* K12 that were found to be unique (light grey) when compared to 122 bacterial proteomes (shown in Appendix 5.1) were designated as orphans (n=174). All remaining proteins (dark grey) were non-orphans (n=4137). Distributions of values for both groups were calculated as a percentage for (a) length, (b) percent low complexity, (c) G+C difference from the mean, (d) Cost and (e) Neighbourhood Distribution.

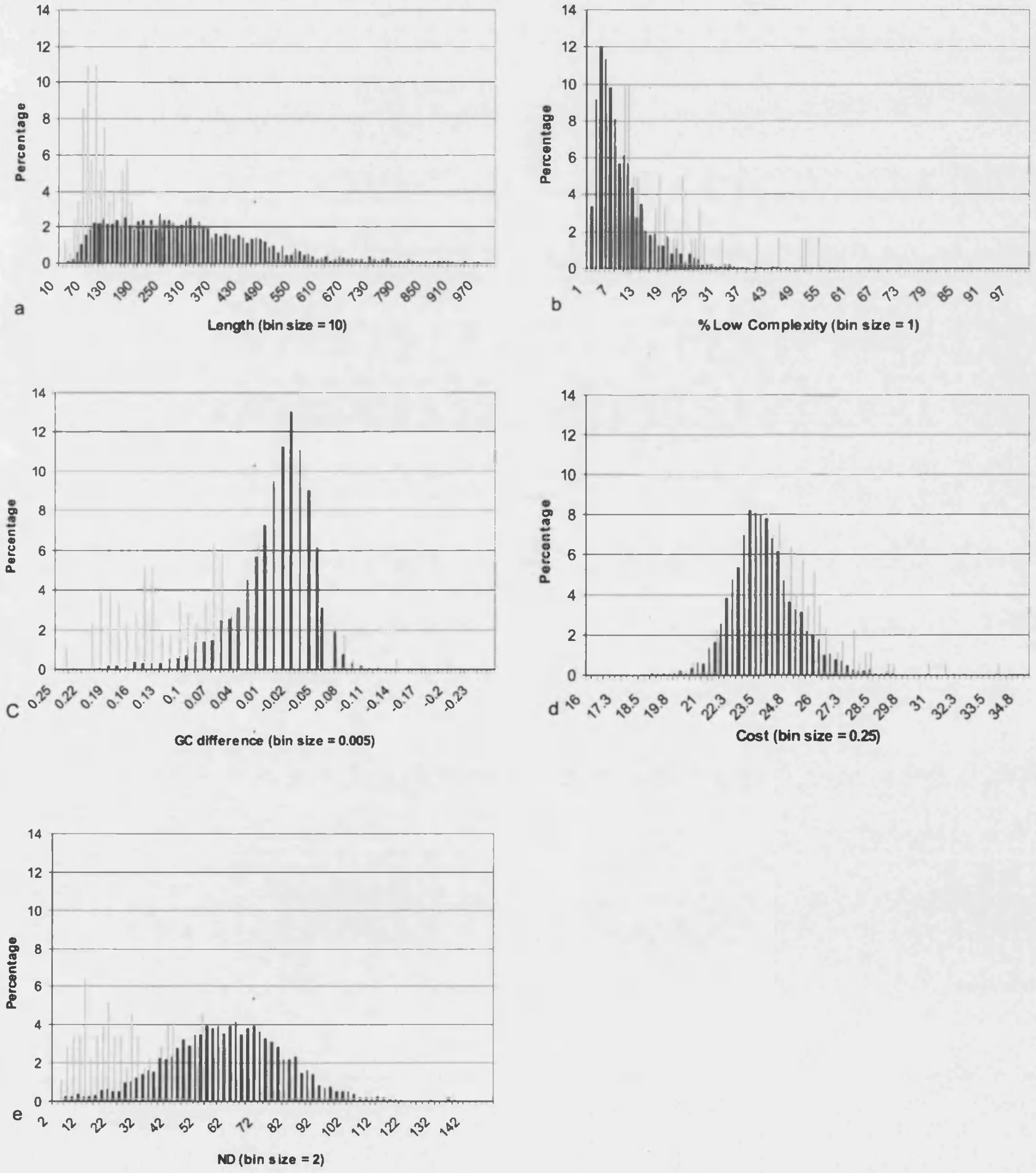


Table 5.1. Criteria used for the calculation of QIPP

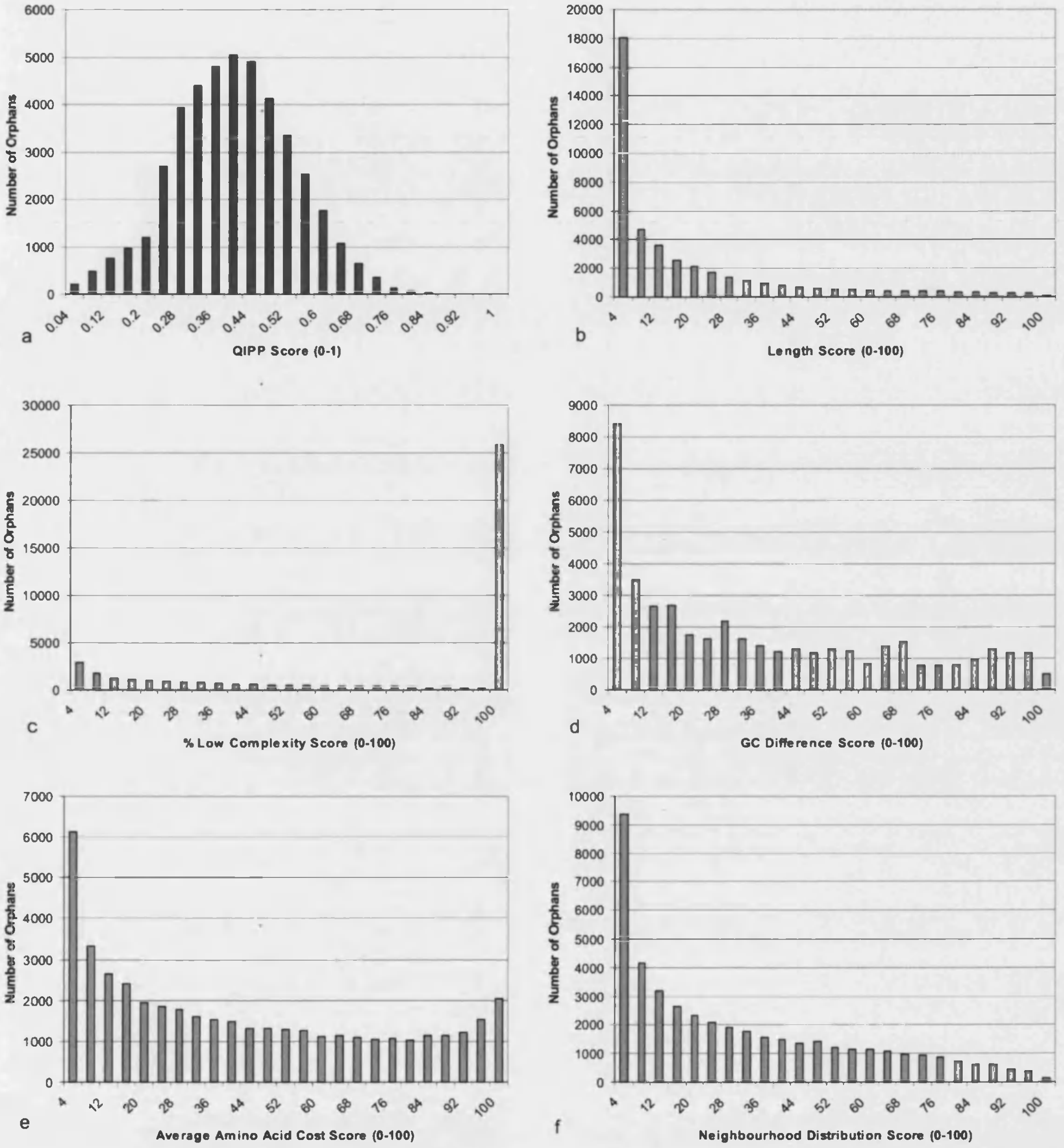
Optimality Criteria	Desirable Values	Ranked by
Length	Long	Distribution of absolute lengths of non-orphans
Complexity	Complex	Distribution of percent low complexity in non-orphans
Cost	Low	Distribution of the average cost per amino acid of non-orphans
G+C Composition	Average composition	Distribution of the difference in G+C content of non-orphans and the genome G+C composition
Neighbourhood Distribution	Location among genes with a broad distribution	Average of the number of genomes with homologues to the 5 genes flanking either side of a gene.

5.3.2 Ranking orphan CDS using QIPP scores

The distribution of QIPP scores across the orphans in this dataset was examined to determine if there was sufficient variation to rank them. Figure 5.2a shows that QIPP scores range from 0.0 to 0.9 (out of a possible range from zero to one) and so the index does have discriminatory power. The overall QIPP scores for each proteome deviate from the normal distribution for all five reference genomes, with too few high-scoring CDS and a longer than expected left-hand tail of low-scoring proteins (Darling-Anderson $p < 0.005$). This is due to the fact that for each criterion (with the exception of low complexity) there are few proteins with very high ranks (Figure 5.2b-f).

We then examined the quality of the highest-scoring orphans to see if our list contained a significant number of potentially 'authentic' orphans – i.e., those unlikely to occur by chance. The extreme right hand distribution of these QIPP scores contains a total of 2,010 single-copy TRGs ($\geq$ 95th percentile with a minimum score of 0.62), 1,260 are longer than 200 amino acids, a criterion that, when used in isolation, is generally accepted to signify 'authentic' CDS (Skovgaard *et al.*, 2001). Relaxing the QIPP score threshold, and using only length as a criterion, a total of 9858 (22.66%) single-copy TRGs are found in this dataset which are $\geq$ 200 amino acids. A subset of these, 2,445 (5.62%), are $\geq$ 400 amino acids.

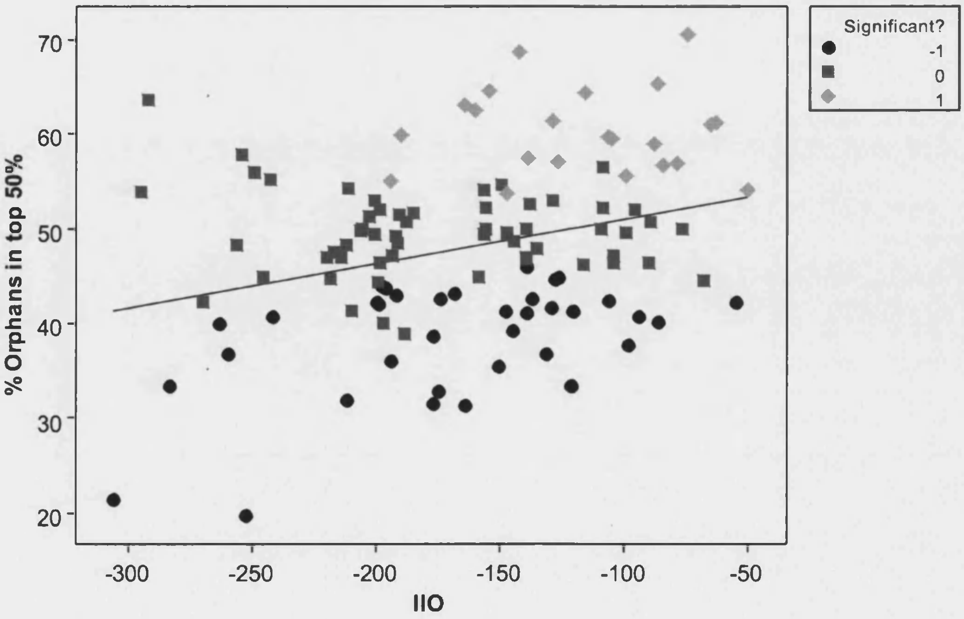
Figure 5.2. QIPP and Criterion Distributions of orphans in 122 bacterial genomes. The orphans (n=43513) obtained from 122 bacterial genomes were scored and the distribution plotted according to (a) QIPP and the individual criteria that constitute QIPP: (b) length, (c) percent low complexity, (d) G+C difference from the mean, (e) cost and (f) neighbourhood distribution.



When interpreting the origins of such high-quality single-copy TRGs, the taxonomic uniqueness of each parent genome must be considered. Of those with QIPP scores above the 95th percentile (>=0.62), only 467 (23%) are from the 62 species (8 per genome) sampled down to the species level (i.e. another species from the same genus is available in the dataset) (average QIPP score = 0.66). In contrast, 1,543 (77%) originate from the 60 species which only have more distant relatives in this dataset. It is presumed that these genomes include many TRGs exclusive to higher taxonomic levels; 24 genomes are unique at the genus level (259 orphans, 11 per genome, average QIPP score = 0.66), 30 at the family level (931 orphans, 31 per genome, average QIPP score = 0.67) and 6 at the division level (353 orphans, 59 per genome, average QIPP score = 0.67). Of those larger than 200 amino acids, 2,878 (29%) are from 62 species (46 per genome) sampled down to the species with an average QIPP score of 0.43. The remaining 6,980 (71%), originate from 60 species unique at the genus level (1,439 total, 60 orphans per genome, average QIPP score = 0.44), the family level (4,263 total, 142 orphans per genome, average QIPP score = 0.48) and the division level (1,278 total, 213 orphans per genome, average QIPP score = 0.50).

When plotted against genetic similarity, more distantly related genomes contribute on average more high-quality, single-copy TRGs (Appendix 5.1 and Appendix 5.2). Chi-squared tests were used to identify genomes that made a greater contribution than expected to the top 50% of the ranked list (Figure 5.3). Genomes that did not contain enough orphans (>5) to perform a chi-squared test were removed from the analysis (n = 6). Genomes that contribute more high ranking QIPP scores are more distantly related (Figure 5.3, ANOVA p = 0.000) but only a low proportion of variability in top-ranking scores is explained by a regression analysis (p = 0.000, R-squared = 10.63%).

Figure 5.3. Genomes which are more taxonomically isolated have larger numbers of high-scoring orphan predicted proteins. Chi-squared tests were used to determine which genomes had significantly more predicted proteins in the top 50% of the list of ranked orphan predicted proteins, than would be expected by chance (-1 = significantly less orphans than expected in top 50% rank, 0 = no significant difference and 1 = significantly more orphans than expected in top 50% rank).



5.3.3 Less conserved genes have lower QIPP scores

The difference between orphan and non-orphan QIPP scores suggests that it might be possible to predict *a priori* how conserved a particular CDS might be using QIPP scores in the absence of homology. To explore this further, we selected a subset of five reference genomes from the best-sampled taxa in our original dataset for which intra-specific comparisons yielding high numbers of strain-specific orphans were also availabie (Tabie 5.2). For each reference genome the taxonomic distribution of all predicted proteins at the Archaea/Bacteria level, domain, division, family, genus, species and strain level (Figure 5.4) was determined.

Table 5.2. Numbers and percentages of species-specific and strain-specific genes after the addition of a second strain in five bacterial species.

Reference Genome	Second Genome	Orphan genes (N=122)	Species-specific (N=122+1)	Strain-specific (orphan genes) (N=122+1)
Escherichia coli K12 (NC_000913)	*Escherichia coli UPEC-CFT073* (NC_004431)	174	52 (29.89%)	122 (70.11%)
Helicobacter pylori 26695 (NC_000915)	*Helicobacter pylori J99* (NC_000921)	258	181 (70.16%)	77 (29.84%)
Neisseria meningitides MC58 (NC_003112)	*Neisseria meningitides Z2491* (NC_003116)	431	222 (51.51%)	209 (48.49%)
Prochlorococcus marinus CCMP1375 (NC_005042)	*Prochlorococcus marinus MIT9313* (NC_005071)	291	40 (13.75%)	251 (86.25%)
Vibrio vulnificus CMCP6 (NC_004459,NC_004560)	*Vibrio vulnificus YJ016* (NC_005139,NC_005140)	348	101 (29.02%)	247 (70.98%)

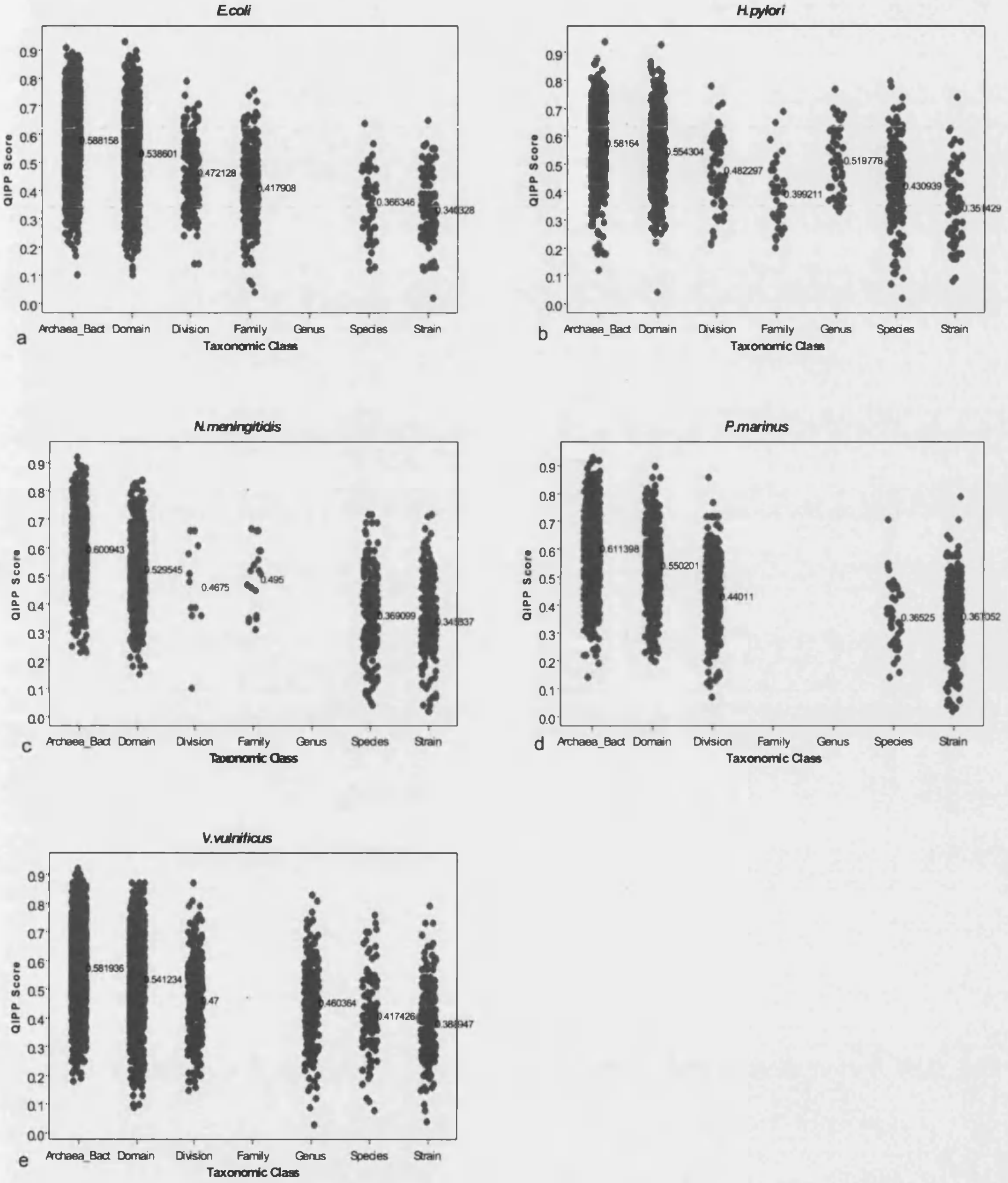
The average QIPP scores and percentages of predicted proteins exclusive to each of these taxonomic levels are given in Table 5.3. Overall, average scores are relatively uniform across the five genomes at each of the 7 taxonomic levels examined. Scores range from an average of 0.60 for proteins conserved across bacteria and archaea down to 0.35 for proteins conserved at the strain-level. These average scores are significantly different across TRG's exclusive to different taxonomic levels (ANOVA, p=0.000 for every genome). The data show an overall decrease in QIPP score as the degree of conservation narrows (Figure 5.4). For the five genomes, when all CDS are taken into account, a regression analysis provides a p-value of 0.000 with R-squared values ranging from 20.3% to 36.3%.

Table 5.3 Table showing the average QIPP score for predicted proteins at each taxonomic level for five selected bacterial genomes.

	Bacteria/ Archaea	Bacterial Domain	Division	Family	Genus	Species	Strain	1st – 3rd Quartile Range
E. coli K12	0.59 (47.75)	0.54 (36.31)	0.47 (4.36)	0.42 (7.54)	N/A	0.37 (1.21)	0.34 (2.83)	0.45 – 0.65
H. pylori 26695	0.58 (43.04)	0.55 (30.58)	0.48 (4.70)	0.40 (2.42)	0.52 (2.86)	0.43 (11.51)	0.35 (4.90)	0.44 – 0.64
N. meningitides MC58	0.60 (41.85)	0.53 (35.98)	0.47 (0.58)	0.5 (0.87)	N/A	0.37 (10.68)	0.35 (10.05)	0.42 – 0.64
P. marinus CCMP1375	0.61 (44.10)	0.55 (21.15)	0.44 (19.29)	N/A	N/A	0.37 (2.13)	0.37 (13.34)	0.42 – 0.65
V. vulnificus CMCP6	0.58 (44.06)	0.54 (36.96)	0.47 (6.46)	N/A	0.46 (4.85)	0.42 (2.23)	0.39 (5.44)	0.44 – 0.64

The numbers in brackets show the percentage of proteins in that genome at that taxonomic level. Scores are highest for proteins which are most highly conserved and decrease across taxonomic categories. N/A = genome not available for comparison. The final column shows the values for the CDS of the 2 quartiles around the median QIPP score in each of the five genomes.

Figure 5.4. Calculated QIPP scores for 5 bacterial genomes split into taxonomic classes. Every predicted protein in (a) *E. coli* K12, (b) *H. pylori* 26695, (c) *N. meningitides* MC58, (d) *P. marinus* CCMP1375 and (e) *V. vulnificus* CMCP6 was put into the taxonomic level at which it was restricted and scored according to QIPP. The numbers on the plots represent the mean QIPP score at each taxonomic level.



The differences in mean QIPP scores between different groups of TRG's are largest for comparisons between groups of CDS conserved above the level of division and those conserved at the species- and strain-level (Table 5.3). Still, average QIPP scores are significantly different between all higher TRG groups when compared to the average for species-level TRGs, while groups of species- and strain-level TRGs cannot be distinguished (Table 5.4). Interestingly, scores from the gene prediction software Glimmer could be used to separate only 7 of the 15 comparisons presented in Table 5.4. Hence QIPP provides additional information which is useful for post-processing gene predictions such as those made by Glimmer, in the absence of homology.

Table 5.4. Statistical significance of QIPP (Q) and Glimmer (G) scores when differentiating between species-specific genes and a respective taxonomic rank.

	Bacteria/ Archaea		Bacterial Domain		Division		Family		Genus		Species	
	Q	G	Q	G	Q	G	Q	G	Q	G	Q	G
E. coli	***	***	***	***	***		**		N/A	N/A		
H. pylori	***	***	***	***	***		*	**	***	***	*	
N. meningitides	***	***	***	***	**		**		N/A	N/A		
P. marinus	***	***	***	***	***	***	N/A	N/A	N/A	N/A		
V. vulnificus	***	***	***	***	***		N/A	N/A	***			

*** = p <= 0.001, ** = p <= 0.01, * = p <= 0.05 , N/A = No representative genomes at that taxonomic level.

In addition to using QIPP to rank individual CDS, we also investigated whether the data had biological meaning. Using quartile analysis, 50% of the CDS in each of these genomes fall uniformly between the absolute values of 0.43 and 0.64 (Table 5.3), suggesting rule of thumb cut-offs for QIPP scores associated with the least (below 0.43) and most (above 0.64) highly conserved CDS in a genome. The data further suggest that the most extreme values of QIPP have the highest degree of predictive power for level of conservation (Figure 5.4). For example, using a minimum threshold score of ≥0.8, 98% of all CDS are members of the most conserved gene families (above the division-level). A total of 58% of CDS with scores less than 0.2 are species- and strain-specific TRGs.

To observe the range of QIPP scores that might be expected from the most highly conserved CDS, we examined a subset of universally conserved genes (Ciccarelli *et al.*, 2006). We found the homologues of these 31 previously defined protein families (Ciccarelli, 2006) in the *E. coli* K12 genome and examined their QIPP scores. These QIPP scores range from 0.5 to 0.87 with a mean of 0.69 (±0.099). A large number of these proteins are ribosomal proteins, which are all of shorter than average size for *E.*

coli. QIPP score is very poorly correlated with the overall length of these proteins (R-squared = 0.012) suggesting that QIPP is not overly sensitive to any one component criterion. The two highest-scoring proteins, both with a QIPP score of 0.89, are extremely different in length (1,138 for the DNA-directed RNA polymerase, beta subunit versus 323 for the DNA-directed RNA polymerase, alpha subunit). When length is removed as a component criterion of QIPP, the scores of the shortest proteins increase by up to 0.16, while those of the very longest proteins decrease by a maximum of 0.09 giving a new mean value of 0.75 (± 0.14).

5.3.4 Validation of orphans with low QIPP scores using results from transcriptomic and proteomic studies

To test whether we could validate the expression of orphans with low QIPP scores in a well-studied model organism, we searched the MicrobesOnline database (Alm *et al.*, 2005) for *E. coli* K12 orphans identified in this study. This database provides experimental microarray results for this organism, for four stress conditions: heat shock (Gutierrez-Rios *et al.*, 2003), pH (Kang *et al.*, 2005), UV exposure (Courcelle *et al.*, 2001) and tryptophan metabolism (Khordursky *et al.*, 2000). We examined the fifty highest and lowest ranked species-level TRGs (N=100). The scores of the top ranking CDS ranged from 0.41-0.64 and the bottom from 0.02-0.28. To illustrate the range of CDS involved, the top scoring CDS was 547 amino acids in length, zero percent low complexity, average G+C content, but was more costly than average and came from a poorly characterised region of the genome. By contrast, the CDS with the lowest score of 0.02 was only 60 amino acids in length, 35% low complexity, had a highly deviant base composition, it was also more costly than average and was found in a poorly characterised region of the genome. Of these 100 orphans, 17 had identifiers not found in the MicrobesOnline database and were excluded. Of the remaining 83, only one failed to show any change in expression levels in any of these experiments. In total there were 46 occasions (involving 35 of these 100 orphans) when one of these orphans was included in the list of the 200 proteins reported in Microbes Online showing the largest (up or down) fold change in expression in one of these experiments. Of particular interest was the pH stress experiment where 12 (three in the top and nine from the bottom 50) of the top 100 up-regulated genes were orphans ($p<0.001$, chi-square).

These results suggest that, despite opinions to the contrary (for example, Skovgaard *et al.*, 2001), sequences that appear unlikely to be coding using both conventional methods (e.g. length) and QIPP, are found to be transcribed. Additionally, a number of these sequences show relatively large changes in their expression levels when exposed to environmental stress, highlighted in the results obtained from the pH stress experiment. Therefore, taking into account the limitations of microarray expression data and the implications of analysing a model bacterial species like E.coli, this data suggests that annotation artefacts are not as common as originally thought.

E. coli K12 proteomic datasets (Corbin *et al.*, 2003, Gevaert *et al.*, 2002 and Taoka *et al.*, 2004) were also searched. When combined these investigations identified approximately 1,800 expressed proteins. While mRNA was found for 64 of the 174 CDS in *E. coli*, only 4 proteins could be identified for all 174 single-copy TRGs in this dataset. These four CDS had an average QIPP score of 0.32 compared to mean score of 0.35 for all *E. coli* orphans. Due to the small number of proteins being found in the proteomic analyses, it is not possible to say anything conclusive about the ability of QIPP to rank CDS according to those most likely to produce a protein. However, it is interesting to note the small number of *E.coli* orphan sequences identified in the proteomic analyses.

.

5.4 Discussion

We have developed an index called "QIPP" ("Quality Index for Predicted Proteins") which can be used to assign a value between zero and one for a CDS, compared to the rest of the genome on the basis of a set of selective criteria. This provides an objective measure of the probability that a given CDS either encodes a protein or is an annotation artefact. Very long CDS, with typical nucleotide and amino-acid compositions, no low complexity regions, and which are found in well conserved regions have the highest QIPP scores and are considered most likely to encode proteins.

The distributions of QIPP scores, and trends in the component variables, confirm that orphans show consistent differences when compared with well characterised protein-coding genes, i.e., they are short, repetitive, possess atypical G+C content, have high average cost for amino acids and are located in poorly characterised regions of the genome. The significant differences in the distributions of QIPP scores between orphan genes and non-orphan genes confirms that QIPP scores represent a valid means to rationalise and automate the identification of those CDS most likely to encode proteins (and find homologues among other available sequences). Because orphans generally have low QIPP scores it is also possible to meaningfully rank them as a subset of all CDS, selectively filter for high-scoring 'authentic' orphans, and begin to address the issue of correcting for the high percentage of orphans in current databases that are simply an artefact of sampling bias.

Our data show that the lowest-scoring CDS encode the least evolutionary conserved proteins, i.e., those orphans restricted to single strains or species. As such, this approach can also provide evidence on the likely taxonomic range of a CDS in the absence of any useful homology. This is particularly significant given the unrepresentative sampling of the current genomic databases. Low-scoring, taxonomically restricted orphans are most likely to be annotation artefacts: we tested this in the case of *E. coli* K12 by reference to online transcriptomic and proteomic expression data. Surprisingly, these data revealed that even these low-scoring CDS are potentially expressed (given the caveats associated with using microarray data to validate orphans (Skovgaard *et al.*, 2001) and the fact that *E. coli* is one of the most thoroughly studied organisms) and therefore suggest that annotation artefacts may not be as common as previously suspected. It should be noted that the use of QIPP is not limited to trying to identify annotation artefacts. For example, it can also be used to indicate the dispensability of a coding sequence (for more details see 6.3.2). It is clear that empirical validation of genomic annotations is necessary and should be of the highest priority (Roberts, 2004, Roberts *et al.*, 2005 and Galperin & Koonin, 2004). At a minimum, it would appear premature to dismiss all very low-scoring orphans as having little biological relevance without further evidence.

It could be argued that some of the criteria used in the QIPP score reflect the extent of purifying selection acting upon a sequence, which, in the absence of homology, precludes the use of more widely-used methods such as examination of dN/dS ratios (Nei, 2005). Over time, metabolically costly amino-acids should be preferentially purged through the process of purifying selection, thus lowering the average amino acid cost for the sequence (Hurst, Feil & Rocha, 2006). Similarly, mutation pressure tends to move in the direction GC->AT rather than *vice versa* (Petrov & Hartl, 1999 and Ochman, 2003) and AT enrichment has commonly been cited as a footprint for relaxed or inefficient purifying selection (but see Foerstner *et al.*, 2005). This can explain the high AT content of obligate endosymbionts or intracellular parasites which are adapted to a restricted niche, undergo restricted gene exchange, and possibly mutate at a high rate due to the loss of DNA repair genes (Wernegreen, 2002). It is also well documented that phage and other mobile elements tend to show a higher AT content than the host bacterial genome (Daubin & Ochman, 2004a and Hurst *et al.*, 2006). As highly conserved proteins are likely to encode essential housekeeping functions, and therefore be subject to high levels of purifying selection, the noted correlation between the taxonomic range and QIPP score can be partially explained within this selective framework. This phenomenon also provides further validation for the use of the QIPP score in identifying "real" genes, as it is expected that CDS which are simply annotation artefacts should be evolving neutrally and hence have low QIPP scores.

This analysis provides proof of principal that the combined use of different criteria can be a powerful approach to determining the biological relevance of putative CDS. The power of the QIPP score could be improved by the use of additional criteria which are likely to reflect purifying selection, such as codon bias, for example. It is acknowledged that the criteria presently used are unlikely to be independent, and multivariate analysis is required to determine the interactions between the variables and to put corrections in place to improve the predictive power of the index. Preliminary analysis on five reference genomes has revealed a significant correlation ($p<=0.05$) between sequence length and complexity, with longer proteins showing more low complexity regions. Further, a significant correlation between G+C content and amino-acid cost was noted in four out of five genomes (the exception being *V. vulnificus*; data not shown). Additionally, the possibility that some of the relationships explored in this chapter are a result of circularity in the methods used needs to be explored. For example, it is possible that short CDS have fewer homologues because they contain fewer functional domains than longer sequences and are therefore likely to significantly match fewer proteins when compared against a sequence database using BLAST.

There is a growing need for metrics that offer a deeper understanding of the detailed content of genomes, especially now that we have such large numbers (Galperin & Kolker, 2006). QIPP provides such a metric and can be used in combination with other *in silico* methods that can now be used to sift out potentially authentic orphans and improve genomic annotation. Such complementary methods include the analysis and removal of short CDS (Skovgaard *et al.*, 2001), gene fragments (Amiri *et al.*, 2003), and pseudogenes (Fukuchi & Nishikawa, 2004) and the ranking of CDS based on the availability of homology-based information (Kosuge *et al.*, 2006). Integration of the information from such studies would provide the foundation for a single, global list of uncharacterised predicted proteins that could be used to systematically subject them to further *in silico* examination (Kosuge *et al.*, 2006, Roberts, 2004 and Galperin & Koonin, 2004). This dataset could further be integrated with empirical evidence from a range of experimental studies, especially high throughput 'omic studies, as is the case for databases like STRING (von Mering *et al.*, 2006). *In silico* studies of predicted proteins can help identify candidates for further examination, but any validation of the biological relevance of a particular protein must be based on empirical evidence (Kolker *et al.*, 2004, Roberts, 2004, Romine *et al.*, 2004 and Kolker *et al.*, 2005). In order to comply with the principle of the transparent access to data for the sake of integration (Field *et al.*, 2007b), all of the data generated in this study is available online in a searchable database, the *OrphanMine*, a database that supports wide-scale downloads of data, including lists of CDS with rich annotations in GFF3 (Generic Feature Format Version 3) (http://song.sourceforge.net/gff3.shtml) format.

In conclusion, the QIPP index supports an objective rationale for prioritising predicted genes for further study, including 'authentic' single-copy TRGs. Although further work is required to refine the approach, this represents an important step in the standardisation and automation of identifying biologically important genes in the absence of homology.

5.5 Material and Methods

5.5.1 Processing of Genomes and Proteomes

All genomic annotations and proteomes as both amino acid and DNA were downloaded from the NCBI RefSeq FTP site. Orphans were detected as previously described (Wilson *et al.*, 2005) using NCBI BLAST (Altschul *et al.*, 1990) and a cut-off of 10^{-3} and then loaded into the *OrphanMine* database for post-processing. The *OrphanMine* interface was used to generate groups of TRGs for each taxonomic level. A custom Perl script was used to calculate length, G+C content and cost and to parse BLAST reports to generate a "neighbourhood distribution" (ND) for each CDS. All of the data used in this study is publicly available through the *OrphanMine*. The code used to generate lists of orphans from proteomes is available in the YAMAP package (www.genomics.ceh.ac.uk/yamap) and all other code (any additional Perl scripts) is available on request (gawi@ceh.ac.uk).

5.5.2 Calculation of QIPP scores

For each genome and for each of the five selected criteria, the distribution of non-orphans was generated and the percentiles for that distribution were calculated. For the criteria of length and ND, the absolute value of each component criterion (e.g. length of 200 amino acids) was transformed into a sub-score from 0 to 100 depending on the percentile in which it fell (e.g. the 35th percentile from the shortest CDS found would be given a score of 35). For low complexity and cost, where more of either actually suggests a less probable CDS, the score was subtracted from 100 (e.g. a protein with 50% low complexity might fall in the 70th percentile and therefore be given a low score of 30). G+C content had to be calculated as the deviation from the mean value. Values above the 50th percentile were corrected by the equation 100 minus the percentile value multiplied by two and values below had their percentile doubled.

Length was calculated as the total number of amino acids and percentage low complexity regions was calculated from regions masked with the SEG programme (Altschul *et al.*, 1994) using default parameters. G+C content was calculated from the proteome as DNA. The average amino acid cost of a sequence was calculated using the relative costs for each amino acid according to the values given in Akashi & Gojobori (2002). Randomised proteomes (i.e., any sequence evolving neutrally) are of average cost, whilst purifying selection appears to select for amino acids that are less metabolically expensive (Akashi & Gojobori, 2002). ND was calculated by determining the level of conservation of the five flanking CDS on either side of a particular CDS. For each of these ten genes, the number of species in which a similar sequence was found was recorded (maximum of 121 for this dataset). Those numbers were then summed, averaged and percentiles generated for the distribution.

The scores from all five criteria are normalised with respect to each particular genome and can therefore be summed. To obtain a final QIPP score between zero and 1, the average is taken and divided by 100. Zero would be the worst possible candidate for a real gene while 1 would be ideal. Using the interface to the *OrphanMine,* it is possible to perform user-selected rankings of subsets of the CDS held in the database, on the basis of one or all of the component criteria used in QIPP. To compare QIPP and Glimmer scores, the five reference genomes were run through Glimmer (v2.13) (Delcher *et al.*, 1999) with default settings.

5.5.3 Genetic Similarity of Genomes and the Taxonomic Distribution of TRGs

The Index of Isolation of an Organism (IIO) similarity measure was calculated by averaging the logarithm of the best E-value for each CDS in a proteome, as described by Fukuchi & Nishikawa (2004). The taxonomic distribution of each CDS in the five reference genomes (Table 5.3) was obtained through interrogation of the *OrphanMine* database (Wilson *et al.*, 2005). For each genome, appropriate queries were performed to find genes restricted to each taxonomic level. The output was scored and downloaded in a tab-delimited format. A Perl script was written to parse the output to ensure that every predicted protein was only counted once and each protein could be classed according to its lineage-specificity.

5.5.4 Obtaining Empirical Data from Microarray and Proteomic Studies

The MicrobesOnline database (Alm *et al.*, 2005) was queried for the *E. coli* orphan genes using their unique VIMSS ID. A file was provided by Keith Keller to map the GenBank IDs of the orphan genes obtained from *OrphanMine* to the VIMSS ID. EchoBASE is a database that curates information regarding the genes and gene products of the model bacterium *E. coli* K-12, including links to literature describing proteomic analyses of this bacterium (Misra *et al.*, 2005). The 'b number' identifiers provided in the literature were used to map data from the proteomic analyses to the *E. coli* orphan genes obtained from *OrphanMine*. When 'b numbers' were not provided, the gene name, if present, was used.

CHAPTER 6

Using the "Quality Index for Predicted Proteins" (QIPP) to Explore the Global Properties of Genomes

Manuscript in preparation for submission to PLoS-ONE as:

Gareth A. Wilson, Eugene Kolker, Rob Edwards, Edward J. Feil and Dawn Field
Using the "Quality Index for Predicted Proteins" (QIPP) to Explore the Global
Properties of Genomes

6.1 Overview

An index for assessing the quality of a predicted protein based on the combined features of its coding sequence (CDS) (length, percentage low complexity, G+C content, amino acid cost, and neighbourhood distribution) was recently proposed. These five criteria were selected for their ability to detect purifying selection and therefore provide a means to gauge the probability that the CDS encodes a functional protein. This index, called the "Quality Index for Predicted Proteins" (QIPP) expresses the 'quality' of a CDS as a number between zero and one. Using QIPP, it is possible to rank and prioritise taxonomically restricted genes (TRGs) for further characterisation and select those species-and strain-specific orphans most likely to represent authentic genes. In an analysis of Bacterial and Archaeal proteomes, a trend for more highly conserved proteins to have higher QIPP scores was found, suggesting that QIPP also contains information about the biological properties of authentic CDS. Here the use of QIPP to characterise the global features of genomes is explored further. Specifically, it is shown that QIPP scores are related to the level of functional information available for a given CDS and also its biological role, as demonstrated by an analysis of subsystem annotations in the SEED database. Secondly, QIPP scores differ between the stable 'core' regions of genomes and CDS associated with the pan-genome. Third, lower QIPP scores are associated with less robust annotation. Fourth, QIPP scores provide a range of biologically meaningful predictions about the nature and evolution of individual proteins and groups of proteins in sequenced genomes. Finally and equally important, especially taking into account the large number of genes without known homologues in current genomes, these predictions can be made even in the complete absence of information on homology. A web server that calculates QIPP scores for GenBank CDS is available at: http://www.genomics.ceh.ac.uk/orphan_mine/qipp_web.php

6.2 Introduction

The pressing need to introduce new metrics for better characterising sequenced genomes is well known (Galperin & Kolker, 2006). An index, "QIPP" ("Quality Index for Predicted Proteins") was developed to further characterise the unknown portions of complete bacterial and archaeal genomes (Wilson *et al.*, 2007). Lineage-specific, or "taxonomically restricted" genes (TRGs), are defined as being exclusively restricted to a particular taxonomic group (Wilson *et al.*, 2005). TRGs are relatively poorly studied and little understood, in large part because many are still orphans or only have homologues in very closely related isolates. This lack of homology confounds attempts to establish the likelihood that a hypothetical gene is expressed and, if so, to determine the putative function of the protein (Kolker *et al.*, 2005).

QIPP scores the 'quality' of a protein without requiring access to direct information about homology for a given coding sequence (CDS). The original analysis was based on the inclusion of five criteria selected for their presumed ability to detect purifying selection and CDS which are unlikely to occur by chance alone (Wilson *et al.*, 2007). These are length (Skovgaard *et al.*, 2001), percentage low complexity (a measure of the degree of repetition) (Altschul *et al.*, 1994), difference in G+C composition of sequence and genome (Navarre *et al.*, 2006), average amino acid cost (Akashi & Gojobori, 2002 and Heizer *et al.*, 2006) and neighbourhood distribution (ND) (Zheng *et al.*, 2005). By combining information on the relative rankings of these features, QIPP was introduced to assign a value between zero and one to any protein based on comparing its features to other proteins in a given genome.

It has been shown that there are significant differences in the distributions of QIPP scores between orphan and non-orphan genes for many genomes (Wilson *et al.*, 2007). QIPP was used to rank the predicted proteins in the proteomes of Bacteria and Archaea and it was found that QIPP scores ranged from 0.0 to 0.9 (out of a possible range from zero to one). This ranking reveals that there is not only a large amount of variation in QIPP scores but also allowed the identification of many high-scoring 'authentic' (expressed) orphans. Perhaps most interestingly, a trend for less well-conserved genes to have lower QIPP scores was observed. This suggests that QIPP can be used not only to prioritise CDS which are likely to be authentic from those most likely to be non-coding but can be also used to provide an indication of the likely taxonomic breadth of CDS.

In this study, the use of QIPP to characterise the global features of genomes is explored further. Since QIPP shows a relationship with the level of conservation of a CDS, it should also be useful in defining several other aspects of CDS biology. For example, highly conserved gene families are often associated with the most functional information as they have been subjected to the most experimental studies. Likewise, it is expected that QIPP scores can provide *a priori* information about the amount of functional information available for a particular CDS. At the same time, low QIPP scores correspond to the isolate-specific genes that are characteristic of the pan-genome and also, the dispensable regions of a genome, which can then be targeted in synthetic, reduced genome experiments (Posfai *et al.*, 2006). QIPP has previously been shown to be useful in highlighting the regions of the genome most likely to contain artefactual CDS (Wilson *et al.*, 2007). This study builds on this and shows that QIPP scores can be used to define the 'brittle' regions of genomic annotations most prone to change over time. Additionally it can highlight conflicts in annotation when different methods of gene prediction are being applied. To further facilitate these and other analyses, a new generally applicable version of QIPP is introduced. This version is entirely independent of any information on homology and is therefore far less computationally demanding. A web server to calculate QIPP scores for GenBank CDS is now available, free of service for the scientific community.

6.3 Results

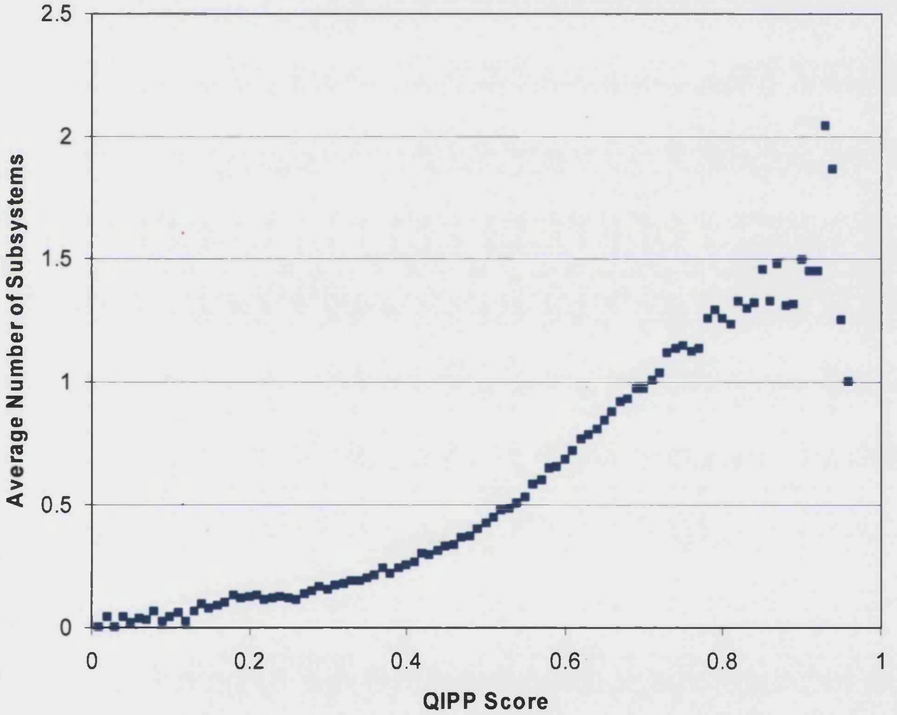
QIPP was originally developed as a method to rank orphans and TRGs in an attempt to prioritise them for further characterisation and help distinguish 'authentic' (expressed) CDS from non-coding artefacts. Such narrowly distributed CDS constitute a significant proportion of all CDS in public genomic databases and are likely to be responsible for unique ecological adaptations, yet they are extremely poorly characterised (Wilson *et al.*, 2005). An overall trend for more highly conserved CDS to have higher QIPP scores has been shown. Here, the type of information that could be contained in QIPP scores is explored further. Specifically, the hypothesis that QIPP scores scale with the amount of functional information available for coding regions is tested. Additionally, the hypothesis that low QIPP scores characterise CDS associated with the pan-genome, and also highlight conflicting gene predictions generated by different methods, is tested.

6.3.1 QIPP Scores are proportional to the amount of functional information available for CDS

Firstly, the expectation that QIPP scores are related to the amount of annotation available for a given CDS was tested, with low-scoring CDS being the least well-characterised. To do this, the annotations in the SEED database were examined. The SEED database uses a subsystem approach to gene annotation. Using this methodology, curators work with automatically processed data to generate expert curations of groups of genes (in a particular subsystem) across the entire genome collection (Overbeek *et al.*, 2005). A subsystem is comprised of a set of functional roles corresponding to a real biological process or structural complex.

QIPP scores were examined to determine whether they could predict the quality of annotation for a given CDS in the SEED database, using QIPP scores from the published dataset of 122 genomes (Wilson *et al.*, 2007). To do so, it was determined whether or not each CDS in this dataset belonged to at least one subsystem. For each possible QIPP score, the average number of subsystems (0 to a maximum of 24, mean = 0.57) to which CDS with that score belonged, was plotted. The results show a clear trend with CDS having higher QIPP scores generally belonging to 1 or more subsystems (Figure 6.1, R-squared = 0.76, p = 0.000).

Figure 6.1. Relationship between QIPP Scores and the number of subsystem annotations in the SEED database.



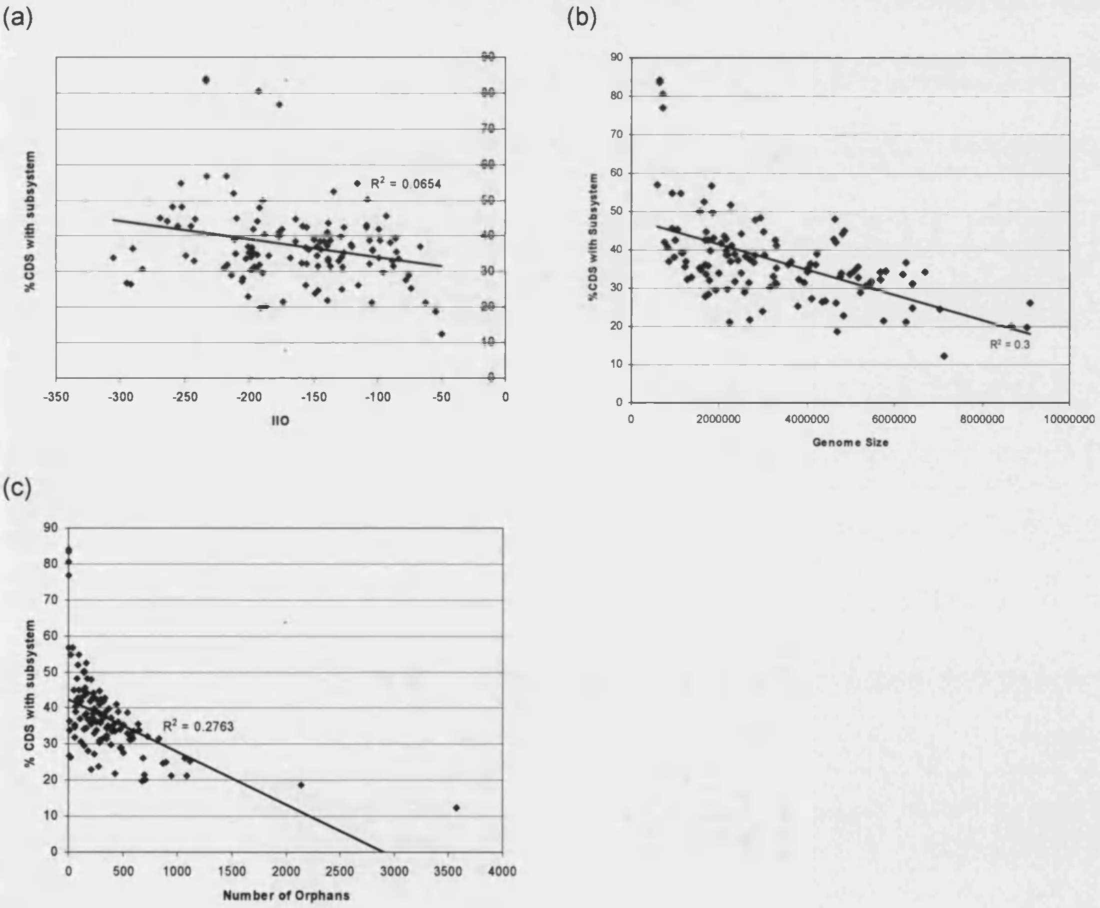
Since QIPP scores correlate with the degree of annotation for a given CDS, it was examined whether average QIPP scores varied between different classes of subsystems. This was to test the hypothesis that CDS involved in core metabolism would have higher QIPP scores compared to those involved in more dispensable functions. Table 6.1 provides a list of parent subsystem classes and their average QIPP score for the dataset of 122 genomes. There is overall variation in average QIPP score across these classes and a clear trend for subsystems responsible for housekeeping functions to have the highest QIPP scores. Likewise, unclassified CDS had among the lowest scores. CDS belonging to prophage were the lowest.

Table 6.1. Average QIPP score for different parent classes of subsystems. For each parent subsystem the average QIPP score is given for an analysis of 122 proteomes along with the median, variance, standard deviation, minimum, maximum and the number of CDS belonging to that parent subsystem.

Parent Subsystem	Mean	Median	Variance	Stdev	Min	Max	Count
Amino Acids and Derivatives	0.63	0.64	0.01	0.11	0.14	0.91	11094
Nucleosides and Nucleotides	0.62	0.63	0.01	0.12	0.18	0.95	6345
Regulation	0.62	0.63	0.01	0.11	0.29	0.91	585
DNA Metabolism	0.62	0.64	0.02	0.13	0.07	0.93	6124
Cell Division and Cell Cycle	0.61	0.63	0.02	0.14	0.06	0.94	2801
Fatty Acids and Lipids	0.61	0.62	0.01	0.12	0.22	0.89	2109
Carbohydrates	0.61	0.62	0.01	0.12	0.14	0.92	12534
RNA Metabolism	0.6	0.61	0.02	0.13	0.09	0.94	4939
Cell Wall and Capsule	0.59	0.6	0.02	0.13	0.03	0.93	6463
Protein Metabolism	0.59	0.6	0.02	0.13	0.06	0.96	11831
Cofactors, Vitamins, Prosthetic Groups, Pigments	0.59	0.6	0.01	0.12	0.11	0.92	16732
Sulfur Metabolism	0.58	0.59	0.01	0.11	0.25	0.88	1010
One-carbon Metabolism	0.57	0.58	0.02	0.13	0.09	0.88	1058
Stress Response	0.57	0.57	0.02	0.12	0.05	0.9	2459
Metabolism of Aromatic Compounds	0.56	0.57	0.01	0.11	0.2	0.84	1748
Unknown	0.55	0.56	0.02	0.15	0.07	0.89	2897
Membrane Transport	0.55	0.56	0.02	0.14	0.1	0.89	1251
Motility and Chemotaxis	0.55	0.56	0.02	0.13	0.07	0.93	5050
Nitrogen Metabolism	0.55	0.57	0.02	0.13	0.15	0.85	1443
Phosphorus Metabolism	0.55	0.55	0.02	0.13	0.12	0.89	1282
Miscellaneous	0.54	0.55	0.02	0.13	0.11	0.91	1834
Virulence	0.54	0.55	0.02	0.14	0.04	0.93	768
Cell signalling	0.54	0.54	0.01	0.11	0.12	0.86	3288
Secondary Metabolism	0.52	0.515	0.01	0.11	0.33	0.75	6370
Respiration	0.52	0.53	0.02	0.14	0.06	0.93	44
No subsystem	0.49	0.49	0.02	0.14	0	0.92	235501
Photosynthesis	0.43	0.45	0.02	0.15	0.01	0.8	391
Sporulation	0.4	0.41	0.02	0.12	0.07	0.61	70
Prophage	0.34	0.35	0.02	0.14	0.1	0.53	16

To further understand the low-QIPP scoring portion of this dataset not assigned to any subsystem, it was examined whether different genomes are annotated to variable qualities. First, the density of subsystem annotations compared to the genetic relatedness of a given genome to the rest of this dataset of 122 species, was investigated. The percentage of CDS annotated within any subsystem for a given genome was plotted against Isolation Index for an Organism (IIO) (Fukuchi & Nishikawa, 2004) (Figure 6.2a). While there is a significant inverse relationship, the amount of variability explained is low (R-squared = 0.07, p = 0.001). Second, the percentage of annotated CDS was compared with genome size (Figure 6.2b). This also shows a significant inverse relationship with smaller, more compact genomes, having more CDS in annotated subsystems and the amount of variability explained is larger (R-squared = 0.30, p = 0.000). This suggests that, as would be expected, genomes that are taxonomically unique within this dataset, and are relatively large, contain a higher proportion of unannotated CDS than smaller genomes and genomes that are members of well-characterised taxonomic groups. Finally, the number of orphans in each genome was compared with the percentage of annotated CDS (Figure 6.2c). As expected there was a significant inverse relationship between the two (R-squared = 0.28, p = 0.000).

Figure 6.2. Relative densities of subsystem annotations in the SEED database. Percentage of CDS in each of the 122 genomes found in one or more subsystems versus (a) IIO (b) genome size (c) the number of orphans (Wilson *et al.*, 2007).



6.3.2 The 'dispensable' CDS in a genome have lower than average QIPP scores: using QIPP to define the Pan-Genome

In a second analysis, QIPP was applied to an analysis of the pan-genomes of different bacterial species. *Escherichia coli* was examined first. This species is of particular interest because, firstly, it is a model laboratory bacterium, secondly, many isolates are available (Liolios *et al.*, 2006), and finally, streamlined *E. coli* K-12 isolates are now being generated using synthetic biology (Posfai *et al.*, 2006). Figure 6.3 shows a genome plot generated by the Genome Atlas (Hallin & Ussery, 2004) showing homology shared between *E. coli* K-12 and 7 other isolates of *E. coli.* The regions deleted from *E. coli* K-12 to form the stream-lined strain are indicated around the outside of the plot. The outermost circle is a plot of QIPP scores for *E. coli* K-12. It is clear from this plot that low QIPP scores appear to correlate with regions of the genomes that are both dispensable in *E. coli* K12 and which correspond to components of the pan-genome. To further quantify this trend, the QIPP scores for CDS in the deleted regions of *E. coli* K-12 were calculated (mean = 0.45, s.d = 0.14) compared to the remaining CDS in the genome (mean = 0.56, s.d = 0.14). The QIPP scores for the two groups were found to be significantly different (t-test, p = 0.000). Of the 718 deleted CDS, 41% (n = 296) were found in the bottom 20% (n = 847) of the *E. coli* K-12 QIPP distribution and 77% (n = 555) were found in the bottom 50% (n = 2118). Chi-Square analysis shows both these results to be significant (p=0.000).

QIPP scores of all of the CDS in the pan-genome of *E. coli* and a range of other bacterial species were analysed. These six additional species were selected either because they were representative species selected in a previous study (Wilson *et al.*, 2007) (due to the availability of a wide range of related genomes from different taxonomic levels) or because they had 9 or more available published and complete genomes from other isolates. The former included, in addition to *E. coli* (n = 8), *Helicobacter pylori (n = 3)*, *Neisseria meningitides (n = 3)*, *Prochlorococcus marinus* (n = 9) and *Vibrio vulnificus* (n = 2) and the latter *Streptococcus pyogenes* (n = 11) and *Staphylococcus aureus* (n = 9). Figure 6.4 shows plots of the average level of conservation, within a given pan-genome, for CDS at each possible QIPP score (from 0 -1, in increments of 0.01). In all cases, there is a clear trend for CDS which are not conserved among all isolates to have QIPP scores lower than the mean for that isolate. An ANOVA was performed on the distributions for each species. In each case there was a significant difference between the QIPP scores for CDS found in different numbers of isolates (p = 0.000).

Figure 6.3. The Pan-Genome of E. coli. A Genome Atlas (Pedersen *et al.*, 2000) image displaying the *E. coli* pan-genome based on the strains available at the NCBI (*E. coli* K-12 (NC_000913), *E. coli* W3110 (AC_000091), *E. coli* 0157 RIMD (NC_002695), *E. coli* 0157 EDL93 (NC_002655), *E. coli* 536 (NC_008253), *E. coli* CFT073 (NC_004431), *E. coli* UTI189 (NC_007946) and *E. coli* APEC01 (NC_008563)). The two outermost circles display QIPP scores for CDS in K-12 and the regions deleted to make the artificial genome (Posfai *et al.*, 2006).

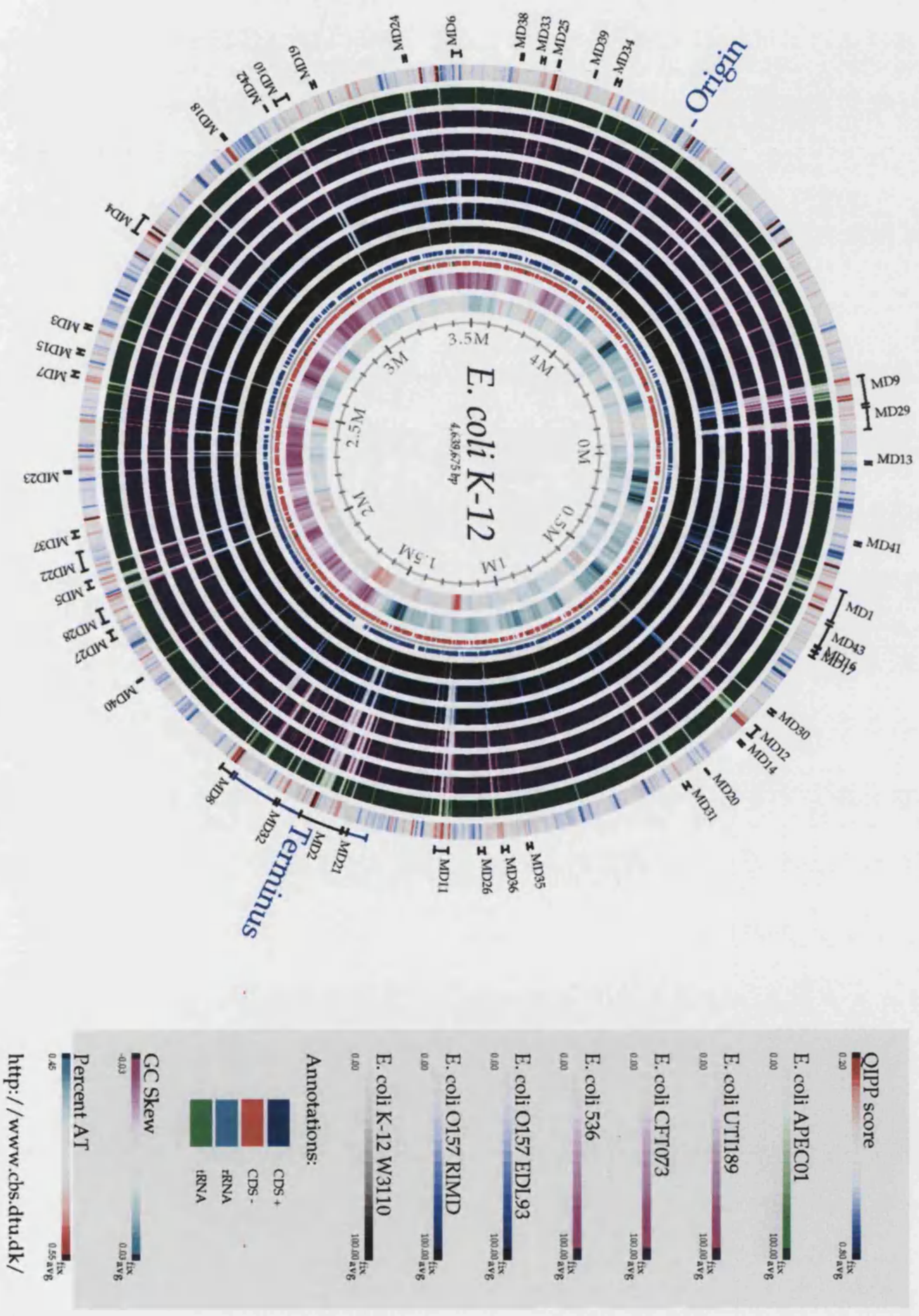
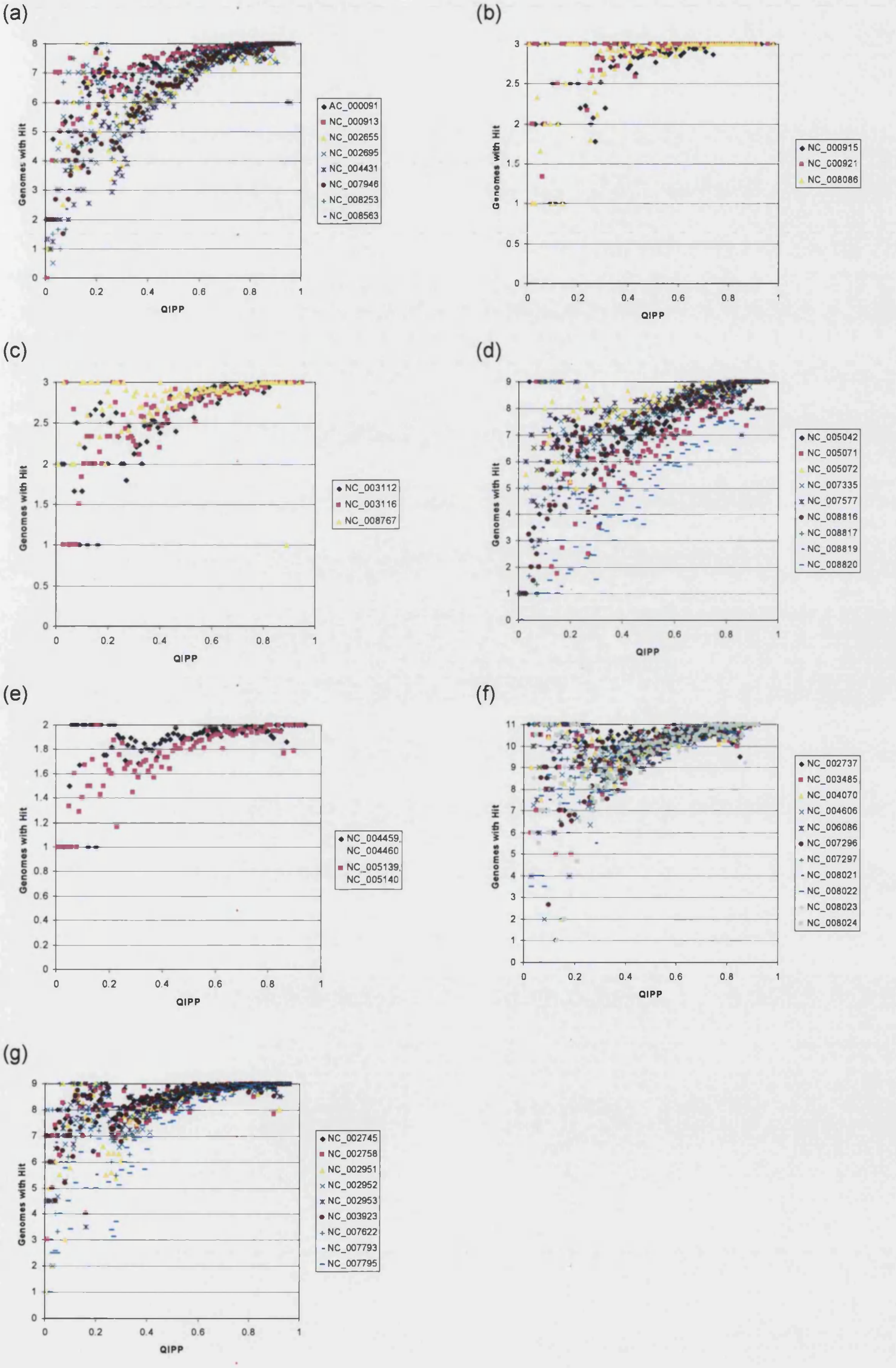


Figure 6.4. Relationship between the frequency of a CDS within a species pan-genome and QIPP scores. The species shown are (a) *Escherichia coli*, (b) *Helicobacter pylori*, (c) *Neisseria meningitides*, (d) *Prochlorococcus marinus*, (e) *Vibrio vulnificus*, (f) *Streptococcus pyogenes* and (g) *Staphylococcus aureus*



6.3.3 'Brittle' Annotations are characterised by low QIPP scores

While there is growing evidence that the pan-genome is an authentic phenomenon (Medini *et al.*, 2005 and Tettelin *et al.*, 2005), an alternative explanation for the lack of conservation of all CDS across strains of a species is mis-annotation or incomplete annotation. QIPP scores were analysed to determine whether they correlate with the portion of a genome most likely to contain CDS which give conflicting results when alternative gene prediction methods are used. For each annotated proteome in its RefSeq collection, NCBI provides access to the output files from two gene prediction programmes, GeneMarkHMM (Lukashin & Borodovsky, 1998) and Glimmer (Salzberg *et al.*, 1998).

Outputs of these two gene prediction algorithms were compared. Four categories of gene predictions were defined and the average QIPP scores for CDS in each of these four categories are shown in Table 6.2. There is a clear trend for 'ambiguous' gene predictions unique to only one of these algorithms to have lower QIPP scores and for gene predictions for which the two algorithms reached a consensus to have higher QIPP scores. This trend was found to be significant (p = 0.000) in all 5 genomes.

Table 6.2. The Number of CDS and Average QIPP score for CDS predicted by Glimmer and GeneMarkHMM. All CDS were placed into one of four categories (1) shared start and stop, (2) same stop codon but a different start codon, (3) unique to Glimmer, (4) unique to GeneMarkHMM. Both the number of CDS in each category and the average QIPP scores are provided.

	Identical		Different Start, Shared Stop		Unique to Glimmer		Unique to GeneMarkHMM	
	Number	QIPP	Number	QIPP	Number	QIPP	Number	QIPP
E. coli	3695	0.59	521	0.57	260	0.38	172	0.39
H. pylori	1453	0.59	163	0.54	74	0.35	119	0.35
N. meningitides	1637	0.62	462	0.61	508	0.45	203	0.47
P. marinus	1457	0.59	399	0.57	122	0.41	52	0.42
V. vulnificus	4043	0.59	504	0.57	320	0.38	157	0.39

6.4 *Discussion*

These results provide further support for the use of QIPP as a genomic index. It can be applied to extract a range of information about CDS in the absence of homology. Specifically, it is shown that QIPP scores can be used to provide an indication of the amount of functional information likely to be available for a CDS and the type of function a CDS may encode e.g. a core house-keeping gene or an accessory gene, for example involved in virulence. Low QIPP scores also characterise the most dispensable regions of a genome. The fact that orphaned and narrowly distributed TRGs have low QIPP scores was further explored at the intra-specific level. These findings show that QIPP scores correspond to the regions of genomes more likely to be strain-specific and involved in the pan-genomes of several bacterial species. Those CDS found in all isolates of a species achieved significantly higher QIPP scores than those found in only a selection of the strains, suggesting such high-scoring regions are more likely to encode core functions shared at the level of species. Regions scoring poorly are likely to be coding for strain-specific accessory functions that may enable an isolate to inhabit a unique niche. Finally, it is shown that the most brittle regions of a genomic annotation, those for which gene prediction programmes provide conflicting results, have very low QIPP scores. CDS predicted by only one programme score significantly lower than CDS predicted by both Glimmer and GeneMarkHMM. CDS predicted by two independent algorithms are more likely to be correct than a CDS predicted by only one algorithm. QIPP scores reflect this tendency and hence provide an alternative measure of confidence in a particular annotation.

6.4.1 Extending QIPP and its application

In this study, QIPP was applied as originally described (i.e., the analysis of the SEED database) but it was also modified to make its calculation entirely homology independent (see Materials and Methods (6.5.2)). This makes QIPP less computationally intensive, more widely applicable, and far more easily implemented. It also removes the dependence on an appropriate database of relevant genomes from which to generate values for neighbourhood distribution (ND). For example, giant viruses (http://www.giantvirus.org) and large environmental plasmids (Tett *et al.*, in submission) have few related genomes available making the selection of an appropriate 'background' database challenging. A web server that calculates homology-independent QIPP scores from GenBank files has been created (http://wwww.genomics.ceh.ac.uk/orphan_mine/qipp_web.php).

Ideally, in the future, QIPP could be refined in a number of ways and methods could be developed to test the best fit of various 'models' of QIPP to real data. For example, the predictive power of QIPP with and without particular criteria, for example ND, could be assessed. The way in which particular criteria are calculated could also be studied in more detail. Currently, the determination of the percentage of low complexity in a CDS does not take into account predicted transmembrane domains. These biologically relevant regions could be 'subtracted' out of the low complexity estimates, thus perhaps improving the predictive power of QIPP scores.

6.5 *Materials and Methods*

6.5.1 Processing of Genomes and Proteomes

All genomic annotations and proteomes, as both amino acid and DNA, were downloaded from the NCBI RefSeq FTP site, along with Glimmer and GeneMarkHMM outputs (ftp.ncbi.nih.gov/genomes/Bacteria). Perl scripts were written to parse through these files and calculate QIPP scores. The protein sequences of the CDS, from the previously published dataset of QIPP scores for 122 bacterial and archaeal genomes (Wilson *et al.*, 2007), were BLASTed against the SEED database (Overbeek *et al.*, 2005) to retrieve the number of subsystems associated with the annotations. A list providing the co-ordinates of the dispensable regions of the E. coli K-12 genome (NC_000913) was obtained from Posfai *et al.* (2006). Homology for the analysis of the CDS in the pan-genome was detected as previously described (Wilson *et al.*, 2007) using NCBI BLAST (Altschul *et al.*, 1990) and a cut-off of 10^{-3}.

6.5.2 Calculation of QIPP scores

QIPP scores were calculated as previously described (Wilson *et al.*, 2007) with the modifications introduced below. In brief, the general procedure behind the calculation of QIPP scores is the generation of distributions for selected quantitative criteria (continuous variables). For each criterion, the distribution of values and the corresponding percentiles are calculated. Subscores for each criterion, for each CDS, are calculated by converting the percentile in which a particular CDS is found, into a score between 0 – 100. These are then added together and divided by the number of criteria used. Finally, dividing by 100 provides a tractable QIPP score between 0 – 1. Zero would be the worst possible candidate for a real gene, while 1 would be ideal. For more information on how the percentiles are converted to scores and the criteria used for the calculation of QIPP scores, see the previously published description of QIPP (Wilson *et al.*, 2007).

6.5.3 Modifications to QIPP

The original calculations of QIPP (Wilson *et al.*, 2007) were based on the inclusion of five criteria: length (Skovgaard *et al.*, 2001), percentage low complexity (a measure of the degree of repetition) (Altschul *et al.*, 1994), difference in G+C composition of sequence and genome (Navarre *et al.*, 2006), average amino acid cost (Akashi & Gojobori, 2002 and Heizer *et al.*, 2006) and neighbourhood distribution (ND) (Zheng *et al.*, 2005). QIPP scores for the SEED database analysis were generated in this way. While this formulation of QIPP does not directly rely on information on homology for any given CDS, it does utilise homology-based information in two ways. First, the background distributions from which percentiles were derived were based on non-orphan CDS only. Second, Neighbourhood Distribution (ND) used information on the level of conservation of ten flanking CDS to calculate the QIPP score for a given CDS. In the generation of QIPP scores for both the analysis of pan-genomes and brittle annotations, QIPP calculations were modified to remove this dependence. Percentiles were calculated based on all CDS in a genome and ND was not used. These two modifications have the benefit of vastly reducing the computational overhead of calculating QIPP (i.e., no need for all-against-all similarity searches).

6.5.4 Other Analyses

The Index of Isolation of an Organism (IIO) similarity measure was calculated by averaging the logarithm of the best E-value for each CDS in a proteome, as described by Fukuchi & Nishikawa (2004). The number of orphans and the genome size of each genome were obtained from the OrphanMine (Wilson *et al.*, 2007).

6.5.5 Software available for the calculation of QIPP

QIPP scores for the 122 proteomes that originally included ND are available from the OrphanMine (http://www.genomics.ceh.ac.uk/orphan_mine/orphan_home.php) (Wilson *et al.*, 2007). The Perl script used to generate the non-homology-based version of QIPP is now available as a web server at www.genomics.ceh.ac.uk/orphan_mine/qipp_web.php. It accepts GenBank files and outputs QIPP scores in GFF or tab-delimited format. The code used to analyse the pan-genome is available in the YAMAP package (www.genomics.ceh.ac.uk/yamap/).

All other code (any additional Perl scripts) is available on request (gawi@ceh.ac.uk). All statistical analyses were performed using Minitab version 4.

CHAPTER 7

A Re-assessment of the Orphan Gene Phenomenon and Directions for Future Research.

7.1 Overview

During the course of this thesis, several advancements have been made in the study of lineage-specific genes.

- The QuickMine pipeline was designed and developed. This freely available open source software is capable of performing BLAST searches on large volumes of data. Additionally, it presents the output in a human readable format that is simple to navigate. It was designed to identify genes unique to a particular genome in a self BLAST database; however it can be applied more generally for analysing BLAST reports from any BLAST database. It has been implemented in the YAMAP system to perform a role in the first pass annotation of genomes. YAMAP is distributed in the NEBC Bio-Linux system (Field *et al.*, 2006) and is used by members of the NERC Environmental Genomics Program.
- The OrphanMine database is publicly available at www.genomics.ceh.ac.uk/orphan_mine. It provides a user friendly interface to explore the data generated by QuickMine. In addition to providing access to pre-computed orphan gene datasets, it provides users with the opportunity to create their own custom dataset of orphan genes, using their defined parameters. Importantly, OrphanMine provides the opportunity to explore datasets of lineage-specific genes. This allows for several different analyses, from investigating genes unique to a particular bacterial division, to exploring the pan-genome of a well sampled species. Data of interest can be downloaded in a variety of formats, including GFF.
- In contrast to the predictions made by Siew & Fischer (2003a), I show how the number of orphan genes found in bacterial genomes has continued to increase.
- The Quality Index for Predicted Proteins (QIPP) was developed. This scoring system ranks proteins according to different criteria (length, low complexity, GC content, amino acid cost and neighbourhood distribution). Those proteins scoring highly were found to be most conserved amongst other bacterial species. Hence QIPP can be used to rank orphans from taxonomically isolated genomes and provide a prioritised list for experimental characterisation. Determining the function of such genes will assist future annotation efforts. QIPP can be calculated for any user-defined dataset, in addition to the four orphan datasets, in OrphanMine.

- The prevailing paradigm regarding bacterial orphan genes was that the majority were annotation errors, caused by an over-annotation of small ORFs. I show that even the lowest quality predicted proteins, as ranked by QIPP, can potentially be coding. Results obtained from microarray analysis of *E. coli,* in different experimental conditions, showed the expression of both high ranked and low ranked orphan genes.
- It was shown that QIPP scores are related to both the level of functional information available for a given CDS and its biological role. This was demonstrated by an analysis of subsystem annotations in the SEED database. It was also found that scores differ for those CDS that comprise the different parts of the pan-genome. The core regions of a species genome, on average, scored more highly than the variable regions.
- The QIPP web server (http://www/genomics.ceh.ac.uk/orphan_mine/qipp_web.php) was created. This allows for the calculation of QIPP scores from GenBank files, in the complete absence of information on homology.

Throughout the course of this thesis, the software and resulting data analyses have been discussed in depth. This brief discussion will re-examine some of the observations made. In addition, it will focus on work that needs to be done in the future, in order to make further progress in this field.

7.2 Numbers of Orphan Genes in Bacterial Genomes

In 2005, I examined the accumulation of bacterial orphans using the proteomes of the first 122 published bacterial species (Wilson *et al.*, 2005). This dataset of 122 genomes was found to be highly biased, with the Proteobacteria and Firmicutes over-represented in the collection. The analyses showed that those species that were taxonomically isolated from other species in the collection provided the largest number of orphan genes. This suggested that by sampling genomes to a sufficient depth, the number of orphan genes would fall. However, it was found that the number of orphan bacterial genes was rising on the addition of each new genome, and this increase was approximately linear. Hence, it was not possible to predict what the maximum number of bacterial orphan genes would be.

Since this analysis was performed, many more genomes have been sequenced. There are now over 300 complete genome sequences, obtained from bacterial species. This increase in genome number provides the opportunity to update the original analyses, using the proteomes of the first 247 completed bacterial species (Figure 7.1). The methods used to generate the data are as described in Chapter 2.

Figure 7.1A shows that the number of predicted orphan genes is continuing to rise. The increase is linear (as it was after 122 species), however the value of the slope has dropped from 0.1279 (n=122, D1) to 0.1082 (n=247, D1). This shows that on average, there are fewer orphans per genome after 247 species (277 orphans per genome), than there were after 122 species (357 orphans per genome). A similar pattern emerges for D2 in which there were 48 orphans per genome after 122 species and 39 orphans per genome after 247 species. Figure 7.1B shows the number of orphans as a percentage of total predicted proteins. It was estimated from the original dataset of 122 species that after 200 species, the percentage of orphans would be 10%, if the trend in selecting candidates for genome sequencing continued. After 247 genomes, the orphan percentage is 9.39% and after 200, the percentage was 10.65%. Therefore these predictions were accurate, suggesting that the trend in genome sequencing has not changed.

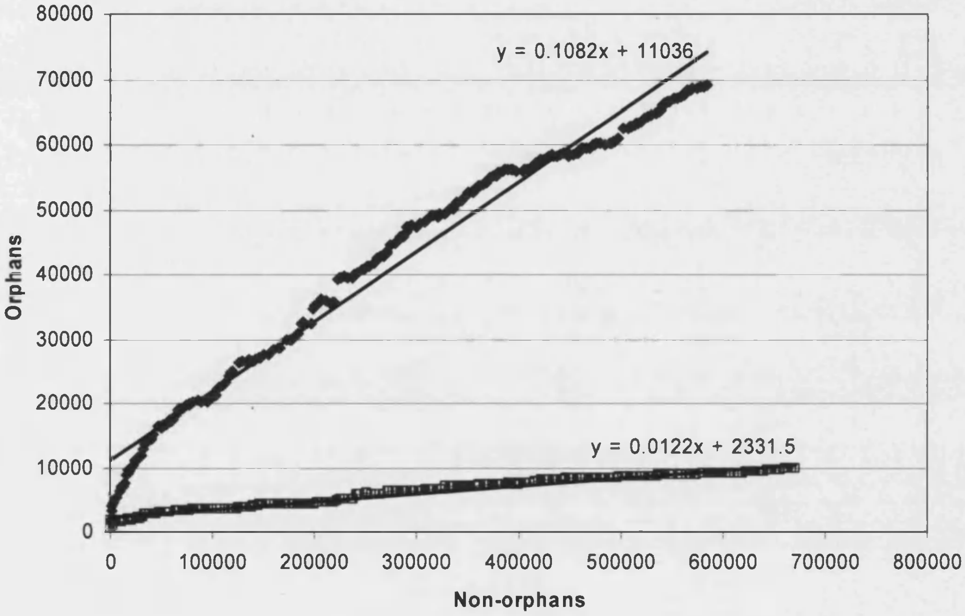
7.3 Trends in Bacterial Genome Sequencing

The results show that calls for an increase in the selection of ecologically diverse organisms for complete genome sequencing, have not been heeded. On average, genomes are less taxonomically isolated (using IIO as a measurement) after 247 species than after 122 (p=0.022, two sample t-test) with an average value for IIO of -179.19 and -164.64 respectively. The Isolation Index of an Organism (IIO) is calculated from the E-values obtained from BLAST reports, the closer to 0 this value is, the more isolated the genome (Fukuchi & Nishikawa, 2004).

Table 7.1 shows the number of genomes in each bacterial division after 122 and 247 species were sampled. After 122 species, 6 bacterial divisions were represented by a single genome. The increase in genome number only improved this poor sampling in one of these divisions. In contrast, the over-representation of the Proteobacteria has been further amplified. The dataset of 122 species contained 46 Proteobacteria (37.7% of the total collection). Of the 247 species, Proteobacteria accounted for 119 (48.18%).

Figure 7.1 A & B. The continued accumulation of bacterial orphans. For this analysis, data on the number of orphans in complete bacterial genomes was taken from the OrphanMine database (www.genomics.ceh.ac.uk/orphan_mine). The dataset D1 represents all the orphans found in the bacterial genomes using BLASTP similarity searches and a cut-off threshold of 10^{-03} (corresponds to dataset D3 in database). In addition a more conservative dataset (D2) was created in which all predicted proteins smaller than 150 amino acids in length containing any regions of low complexity were removed (corresponds to dataset D4 in database). **A.** A plot of the cumulative number of orphans versus non-orphans. The number of orphans in datasets D1 (■) and D2 (□) are plotted showing that the number of orphans is continuing to rise in a linear fashion. Each data point represents the addition of a complete genome sequence in chronological order of publication (N=247 species). **B.** The decline in the number of orphans in datasets D1 (■) and D2 (□) as a percentage of all predicted proteins. A power curve was fitted and the R^2 value is shown.

A.



B.

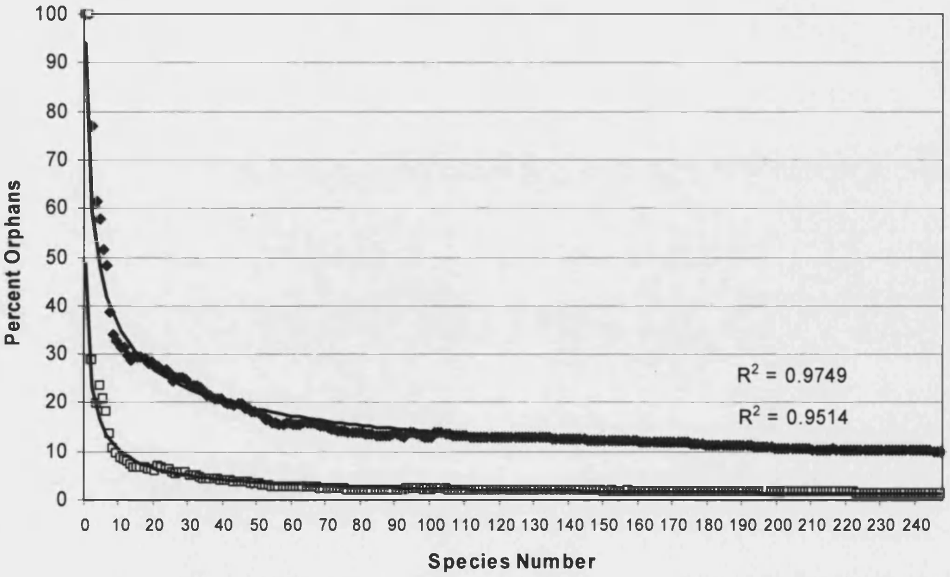


Table 7.1. The number of species representing each bacterial division after 122 and 247 bacterial species.

Division	122 species	247 species
Actinobacteria	9 (7.38%)	18 (7.29%)
Proteobacteria	46 (37.70%)	119 (48.18%)
Aquificae	1 (0.82%)	1 (0.40%)
Bacteroidetes/Chlorobi	3 (2.46%)	7 (2.83%)
Chlamydiae	3 (2.46%)	7 (2.83%)
Chloroflexi	0 (0%)	2 (0.81%)
Crenarchaeota	4 (3.28%)	5 (2.02%)
Cyanobacteria	6 (4.92%)	8 (3.24%)
Deinococcus-Thermus	1 (0.82%)	2 (0.81%)
Euryarchaeota	12 (9.84%)	21 (8.5%)
Firmicutes	30 (24.59%)	48 (19.43%)
Fusobacteria	1 (0.82%)	1 (0.40%)
Nanoarchaeota	1 (0.82%)	1 (0.40%)
Planctomycetes	1 (0.82%)	1 (0.40%)
Spirochaetes	3 (2.46%)	5 (2.02%)
Thermotogae	1 (0.82%)	1 (0.40%)

In order to obtain the genome of 122 unique bacterial species, 150 genomes had to be sequenced. Therefore, 28 genomes represented a species already sequenced. To have the genome of a new bacterial species available to the public, in total, 1.23 genomes had to be sequenced. After 247 unique bacterial species, the genome collection contains 330 genomes. 83 genomes represent an already sequenced species, giving the ratio of 1 species to every 1.34 genomes. This suggests that, rather than expanding the ecological diversity of our genome collection, an increasing number of analyses involve searching for intra-specific differences in gene content. Intraspecies comparisons have enabled scientists to approach fundamental evolutionary questions with renewed vigour. The role of horizontal transfer events in bacterial species has been highlighted by such work, for example in *Prochlorococcus marinus* (Rocap *et al.*, 2003). Intraspecies comparisons have also led to further progress in the study of pathogenicity and drug resistance, for example in *Staphylococcus aureus* (Diep *et al.*, 2006). In the collection of 330 genomes, 43 species were represented more than once and 17 of these were represented more than twice. *Staphylococcus aureus* was sampled in the greatest depth (9 strains) followed by *Streptococcus pyogenes* (7 strains), *Escherichia coli* (6 strains) and *Prochlorococcus marinus* (5 strains). Three of these four species possess pathogenic potential (the exception being *P. marinus*) as do another 31 of the 43 species with multiple representations (in total 79.07%).

Despite the increasing number of sequenced genomes, many remain taxonomically isolated. After 150 genomes, representing 122 species, *Rhodopirellula baltica SH1* (previously *Pirellula sp.1*) contained 3568 orphans (48.7% of the total predicted proteins). After 330 genomes, representing 247 species, it contains 3386 orphans (46.25% of the total predicted proteins). Therefore, the increase in complete genome sequences in the public domain has had minimal impact on the annotation of this organism (approximately 1 orphan is homed for every genome added). For scientists working with genomes that come from taxonomic lineages unlikely to be sequenced in depth, QIPP provides a useful tool. In the absence of homology, it is capable of ranking the predicted genes. This ranked list can provide researchers with a basis for the determination of candidates for experimental characterisation. The average QIPP score for the *R. baltica SH1* orphans, in the 150 genome dataset, that were homed in the 330 genome dataset, was 0.49. In contrast, the remaining *R. baltica SH1* orphans scored an average of 0.39. The difference in scores was found to be significant (p=0.00, two sample t-test). This adds to the evidence suggesting that proteins scoring more highly in QIPP are more likely to find a homologue in the future and therefore be of greater benefit to the wider community.

7.4 Exploring Diversity through Metagenomics

It is likely that there are many million species of bacteria, yet only a few thousand have been formally described (in contrast to the 350000 described species of beetles) (Eisen, 2007). This discrepancy is largely due to inherent problems associated with studying organisms that can not, currently, be cultured. The promising new field of metagenomics provides the opportunity to study microbes directly in their natural habitats, thus bypassing the need for isolation and lab cultivation of individual species. Metagenomics makes use of shotgun genome methods to sequence random DNA fragments from microbes in an environmental sample. There are now more than 70 such projects in various states of completion (Lioslios *et al*, 2006), assaying a range of environments, for example, from the human gut (Gill *et al.*, 2006) to waste water sludge (Garcia *et al.*, 2006).

The largest and most ambitious metagenomic projects have been carried out by Craig Venter. In 2004, Venter *et al.*, performed shotgun sequencing on samples obtained from the Sargasso Sea. Their study resulted in the identification of more than 1.2 million new genes, from the DNA extracted from approximately 1500 litres of surface seawater. The Sargasso Sea is one of the world's most nutrient impoverished bodies of water, thus the fact that such a massive number of novel genes was obtained from so few samples provides an indication of the true scope of Earth's genetic diversity

(Falkowski & de Vargas, 2004). More recently, the results of the Sorcerer II Global Ocean Sampling (GOS) expedition have been released (Rusch *et al.*, 2007). These environmental samples were found to contain 6.12 million predicted proteins, effectively doubling the number of known proteins (Yooseph *et al.*, 2007). Known protein families now contain a greater diversity of protein sequence. In addition, new protein families are being discovered at a linear rate. 6044 sequences, previously described as orphans, were found to have matches to the GOS data (Yooseph *et al.*, 2007). Hence, the data coming out of metagenomic analyses will make a significant contribution to finding gene families for orphans, in environmental bacterial species.

QuickMine is suited for analysing metagenomic data. Since its incorporation into the YAMAP annotation package, it has been used for first pass analyses of sequence data obtained from the environment. Members of the NERC funded Microbial Metagenomics project (http://www.genomics.ceh.ac.uk/mm/index.php) have utilised QuickMine, as part of the YAMAP annotation package, to perform first pass annotation of sequence data obtained from water samples. As more researchers gain access to metagenomic sequence data, the demand for software such as QuickMine and YAMAP will grow.

7.5 The Future of the Genome Collection

The science of genomics is technology driven. As new technologies and methods evolve, more ambitious sequencing projects can be performed (Eisen, 2007). One example is that of 'community whole genome sequencing'. A metagenomic approach has already been applied to the human gut microbial community (Gill *et al.*, 2006), however, the Human Gut Microbiome Initiative aims to produce deep drafts of 100 intestinal species (http://www.genome.gov/Pages/Research/Sequencing/SeqProposals/HGMISeq.pdf). This will be performed by utilising new technologies, such as pyrosequencing (often referred to as 454 sequencing). Such studies will allow scientists to perform a number of different analyses. For example, they could determine the total number of genes involved in producing the metabolic capacity of a community, or analyse the rates of horizontal gene transfer and investigate the role of the pan-genome in bacterial adaptation (Field, Wilson & van der Gast, 2006).

Given the current rate of genome sequencing, it has been estimated that by 2010, there will be over 4000 bacterial genomes available (Overbeek *et al.*, 2005). Such a genome collection will be of great scientific importance and the financial investment required to generate it will be substantial. Therefore, as a community, we should make every effort to describe it accurately. This not only involves the annotation of the sequence data, but also of the genomic metadata.

It should be essential that metadata is captured accurately. This includes putting each genome sequence into its correct geospatial and temporal context (latitude, longitude, altitude/depth, date and time of sampling) and also providing details of the experimental method used (e.g. sequencing method) (Field *et al.*, 2007b). Obtaining such data will allow many questions to be asked of the genome collection that are not currently possible. For example, analyses of different annotation methods may highlight biases in particular procedures, such as the over-prediction of genes. Currently, metadata describing a particular species or strain, for example the primary habitat and host associations, are often found only in the primary literature on a per-genome basis, or alternatively in reference works, such as Bergey's Manual (Garrity, 2001). The distributed and patchy nature of this information creates great difficulties when trying to curate comparable data for hundreds of genomes.

The lack of accurate and complete genomic metadata, coupled with the questionable accuracy of genome annotation, acted as a major bottleneck in comparative analyses investigating factors affecting the numbers of orphan genes. The GSC (Genomic Standards Consortium) (Field *et al.*, 2007b) formed in order to reach a consensus regarding the collection of genomic metadata. The goal of the GSC is to promote mechanisms that standardise the description of genomes and the exchange and integration of genomic data. Such standards will not be restricted to bacterial genomes, but will also be relevant for other projects ranging from viral genomes to large metagenomic projects. Only by developing such standards, with active involvement from the international research community, will it be possible to have a genome collection that can be interrogated with confidence. Once initiatives such as the GSC are fully supported by the community, it will become trivial to obtain necessary metadata. Additionally, QIPP scores could be used to act as a threshold value, therefore allowing only predicted coding regions scoring above a given value to be used in a particular analysis. The availability of the QIPP web server means that users can calculate QIPP scores for any genome and are not reliant on the updating of the OrphanMine. In the future, QIPP, in conjunction with reliable genomic metadata, could be used to perform interesting analyses. For example, it will be possible to accurately investigate the effect of habitat, or the effect of annotation methods, on the number of orphan genes. Such analyses will need to control for the effect of taxonomy. This could be done using quantitative measures such as IIO, which can indicate the taxonomic isolation of a sample within a given dataset.

7.6 Future Applications of QIPP

As a stand-alone method, QIPP is still in the early stages of its development. However, both the results presented in this thesis and the support from the research community, suggest that further research to refine the technique would be of benefit. Such refinements could simply be the addition of new criteria, for example, dinucleotide frequencies. An analysis centred on the correlation between different criteria may highlight biases in the scoring. Additionally, correlations between criteria and real biologically relevant regions (e.g. low complexity and transmembrane regions) also need to be explored. Such work would be greatly enhanced by the availability of an experimentally verified dataset. This would allow for the accurate exploration of the value and meaning of QIPP scores.

QIPP could also be extended to other taxa, for example, to determine if the patterns seen in bacterial genomes hold true for eukaryotic genomes or large genetic elements. For example, giant viruses (http://www.giantvirus.org) (Raoult *et al.*, 2004) and large environmental plasmids (Tett *et al.*, in submission) contain large numbers of orphans. For the analysis of such genomes, non-homology based metrics hold special appeal, because such taxa have relatively few related genomes available in public databases, making the selection of an appropriate 'background' database challenging.

Metagenomic analyses provide a different challenge. The calculation of QIPP scores is based on the distribution of the various criteria within a genome. Metagenomic analyses do not provide this genomic context. Therefore QIPP is not, as a complete method, transferable to metagenomic datasets. However, there is a need to develop methods to provide an indication of the likely coding potential of a given sequence. This is of particular importance in the large datasets generated by pyrosequencing. Using homology-independent criteria, such as those used in QIPP, may provide a starting point for the development of such a method.

It is possible, though purely hypothetical, that QIPP could also be used to provide an overall evaluation of the depth of annotation in a genome. This could be done by determining the proportion of orphans above a certain QIPP score, e.g. 0.7. It is plausible that an inverse relationship could exist, between the number of orphans above this threshold score and the level of knowledge regarding the given organism.

7.7 Conclusion

As both our knowledge and resources expand in the area of microbial genomics, we can begin to penetrate the issue of the orphan genes. As each new genome project is completed, more orphans are placed in families. The results of the large metagenomic analyses highlight the extraordinary levels of microbial diversity present in our environments, and in doing, so discover new gene families and find families for orphans to join (Yooseph *et al.*, 2007). Whilst my data shows that the numbers of orphans are still increasing, it no longer seems so unexpected, given the vast levels of genetic diversity being uncovered. Hence, the majority of bacterial orphans appear to be an artefact of a lack of sampling depth.

Laboratory techniques, such as expression and proteomic analyses, can help in elucidating the accuracy of gene predictions. Such research is still in its early stages but results suggest that small CDS are expressed surprisingly often. Therefore, such regions may not be errors in annotation and should not automatically be regarded as such (Wilson *et al.*, 2007). Further proteome based studies will assist in providing evidence of a protein product resulting from such sequences. The orphan sequences that arise as a result of annotation errors, may slowly be removed from the public databases by using such techniques.

Resources need to be developed to permit effective knowledge exchange. For such developments to be useful and widely used, the resources need to be centrally linked and easy for people to use. The community will be required to provide annotations in a structured format. Evidence for their annotation will need to be captured, as will their name and institution. It may also be necessary to provide links to the experimental data used to form their judgements. Capture of such data will help to provide good provenance to the annotations. This transparency, together with the evidence and associated metadata, should help with providing useful knowledge to the community. For such a resource to be fully utilised, a change in the way in which sequence identifiers are applied and used in databases is required. A universal gene identifier needs to be introduced. This will enable effective linking of the community annotations to all relevant databases. Such an initiative would facilitate more effective knowledge sharing and would permit more detailed analyses. In conjunction with a structured community led sequence annotation project, much progress could be achieved.

Despite these developments, sequence annotation accuracy remains a key issue. Whilst metrics, such as QIPP, can help assess the quality of predicted proteins, it appears more work needs to be done to stop the errors at their source. This could be achieved through the development of novel algorithms; however, it is through close integration of computational biologists and relevant experts (for example, a specialist on the genome of interest) that I see progress being made. Whilst this is not a novel suggestion (McInerney, 2002), it is one that is often over-looked. Breaking down the barriers between different disciplines and permitting knowledge sharing between groups should result in cleaner and more accurate annotations.

APPENDICES

Appendix 3.1 – Detecting Homology using BLAST

BLAST (Basic Local Alignment Search Tool) (Altshul *et al.*, 1990) is one of the most heavily used sequence tools available in the public domain (McGinnis & Madden, 2004) and is claimed to be the 'single most important piece of software in the field of bioinformatics (Korf, Yandell & Bedell, 2003). It is commonly used via a web interface but can also be used as a stand-alone tool capable of performing batch analyses. BLAST was first developed in 1989 at the NCBI, since then several versions have become established. Examples include BLASTN, used for comparing a nucleotide sequence with a nucleotide database and BLASTP which compares amino acid protein sequences against a protein sequence database.

Sequence similarity is a powerful tool for providing putative functional assignments to newly obtained sequence data. Thus a major goal of sequence alignment is to enable a researcher to determine whether two sequences display sufficient similarity to infer homologous relationships between each other (Baxevanis & Ouellette, 2005). BLAST is a fast and reliable (both statistically and computationally (Korf *et al.*, 2003)) method to analyse sequence similarity.

Amino-Acid Scoring Matrices

A scoring matrix is a two dimensional matrix containing all possible pairwise amino acid scores. In the PAM (Percent Accepted Mutation) matrix (developed by Margaret Dayhoff in the late 1960's and early 1970's), each element shows the probability that the original amino-acid will be replaced by another amino-acid over a defined evolutionary interval.

More recently a second type of scoring matrix was introduced. S. Henikoff and J.G. Henikoff (1992) developed the family of BLOSUM (Blocks Substitution Matrix) matrices. The goal was to replace the PAM matrix with a matrix that would perform better in identifying distant relationships (Lesk, 2005). BLOSUM matrices were constructed by extracting ungapped segments (known as blocks) from aligned protein families. These blocks were further clustered on the basis of their identity. For example, the blocks used to derive the BLOSUM62 matrix all have at least 62% identity to another member of the block. Generally, today, BLOSUM is more commonly used, as it is believed that BLAST searches employing BLOSUM matrices offer greater sensitivity (Korf *et al.*, 2003).

The BLAST Algorithm

Broadly speaking, there are two methods for aligning two sequences. Sequences can be aligned globally or locally. Global similarity algorithms, such as Needleman-Wunsch, optimise the overall alignment of sequences. This method is best suited for finding matches in long stretches of sequence with low levels of similarity. Local similarity algorithms, such as Smith-Waterman, identify relatively short alignments. This is useful in biological sequences as there are often regions of local similarity (domains, active sites) but not global regions of similarity.

BLAST searches for local regions of similarity. However, unlike the Smith-Waterman method, it does not explore the entire search space between two sequences. This fact is key to its speed and sensitivity. The reason why BLAST can produce accurate alignments quickly comes down to the heuristic nature of its algorithm. The algorithm contains three heuristic layers: seeding, extension and evaluation.

Seeding refers to the initiation of an alignment. It assumes that significant alignments have 'words' in common. A word is a defined number of letters. When two sequences are compared, only those regions with word hits will be used as alignment seeds. In BLASTP, the idea of a 'neighbourhood' is introduced. The neighbourhood of a word is a list containing the word itself and all other words whose score is at least as big as a pre-defined threshold (T) when compared via a protein scoring matrix such as BLOSUM62. By adjusting the value of T, it is possible to control the size of the neighbourhood and therefore the number of word hits. The interplay between word size (W) and T is the most effective method for controlling the speed and sensitivity of BLAST (Korf *et al.*, 2003).

Extension refers to the extension of the seeded alignment. The extension occurs in both directions. The endpoint of the alignment extension is calculated using the pre-defined value of X. X is a measure of how much the alignment score is allowed to drop, since the last maximum value. Once the score has dropped by the value of X, the extension is terminated and is trimmed back to the previous maximum score.

The final stage is the evaluation. This refers to the evaluation of the alignments to determine if they are statistically significant. A significant alignment is called a 'HSP' (high-scoring pair). The evaluation is not as simple as just using a score threshold because of the presence of multiple HSPs. Instead an alignment threshold is used. This threshold is set by the software and therefore is not a user definable parameter.

The alignment threshold is an effective method for removing many random, low-scoring alignments. Once the HSPs have been organised they are evaluated using a final threshold. The final score calculated for a sequence is utilised in the Karlin-Altschul equation (see 3.3.3) to determine if the match is statistically significant. The output from the Karlin-Altshul equation is compared with the final threshold. The final threshold (E) is a parameter entered by the user. If the calculated value for E is less than the threshold value provided by the user, the alignment is printed out to the report (Korf *et al.*, 2003).

Karlin-Altshul Equation

In 1990, Samuel Karlin and Stephen Altshul published a theory of local alignment statistics. The central element of this theory is the Karlin-Altshul equation:

$$E = kmne^{-\lambda S}$$

The equation states that the number of alignments expected by chance (E) during a sequence database search is a function of the size of the search space (m * n), the normalised score (λS) and a minor constant (k). The size of the search space is a product of the length (in amino acids) of the query sequence (m) and the number of letters in the database searched (n). Lambda (λ) is a matrix specific constant responsible for converting the raw score to a normalised score. The lower the value of E, the less likely it is that the alignment is a result of random similarity.

Appendix 3.2 – Condor

Clustered computing at its most basic level involves two or more computers serving a single resource (Bookman, 2002). Many scientists, particularly in the field of molecular biology, are now involved in the type of research that needs a large amount of computational power over a long period of time. This form of computing environment is called a 'High Throughput Computing' (HTC) environment..

Condor is a system that takes advantage of resources that would otherwise be wasted. Condor is a result of the work of the Condor Research Project based at the University of Wisconsin-Madison (http://www.cs.wisc.edu/condor/). A long running job, expected to require the exclusive use of a workstation for several days, may produce results overnight using Condor (dependent on the size of cluster used). To utilise Condor, users submit their jobs through the use of a submission file. Condor places these jobs in a queue and chooses when and where to run the jobs based upon a pre-defined system, known as Class-Ads. The progress of the jobs is monitored and, when completed, the user is informed. Class-Ads allow machines to advertise resources available for use, and allow the submitted jobs to advertise for the resources they wish to use (Mausolf, 2005b).

The universe, under which the user wants their jobs to be run, must be specified during job submission. The universe refers to the run-time environment (Mausolf, 2005b). There are six different universes available: standard, vanilla, PVM, MPI, globus and java. The most commonly used are the standard and vanilla universe. The vanilla universe is generally used when users do not have access to the source or object file and thus the jobs can not be linked with the Condor library. This lack of access prevents the use of the standard universe. As a result, the vanilla universe cannot provide functionality such as job check-pointing. Check-pointing allows a job to resume from the most recent check-point if the job fails.

Condor and The Grid

The Grid refers to the networking of a potentially unlimited number of computer devices within a grid. This approach to computing has been likened to the electricity grid that serves electricity directly to our homes and businesses (Joseph & Fellenstein, 2003). It is believed that the Grid may be able to tap into a reservoir of computational power when and where it is needed. However, such a scenario is still some time in the future.

Currently the easiest use of Grid computing is to run an existing application on a different machine. Even for this simple example, prerequisites exist. For example, the application must be executable remotely and the remote machine must meet any requirements such as specific hardware or software. Also, in order for a user to access a Grid, they must first enrol. This is likely to involve establishing identity with a Certificate Authority (Ferreira *et al.*, 2003). In order to connect to resources over a Grid, computational tools will be required. The Globus Toolkit is a set of tools useful for building a Grid.

The Globus Toolkit was developed to enable resource sharing across administrative domains. It allows for job submission, monitoring and control in a heterogeneous environment. Over time, the Globus Toolkit has emerged as the standard for Grid infrastructure (Mausolf, 2005a). However, the Globus Toolkit does not include a scheduling component. A scheduler is responsible for determining when and where to run a job. The scheduler co-ordinates with Globus, this allows the job to run on the selected resource. Condor can act as the scheduler by using the Condor universe called globus. Using this universe, Condor submits jobs to remote Grid resources through the Globus Toolkit (Mausolf, 2005a).

Using Condor in combination with Globus is known as Condor-G. In effect Condor-G should provide a window to the Grid for users to access resources and manage jobs running on remote machines.

Appendix 3.3 – QuickMine Configuration File

```
# config file to be used with "quickmine.pl" script
# Cared for by Gareth Wilson (gawi@ceh.ac.uk)

#
# Notes:
#     All the specified directories must exist before you run the
# "quickmine.pl" script
#     End any lines that wrap to the next line with "\" or
#  Config::Simple will throw an error (like: "can't call method
#"param")


#
# Where are the proteomes located?
#

# Note: use no trailing /
path2proteins = "/home/gawi/proteomes"

#
# What ending is used for the proteome files?
#

ext = "\.faa"


# What ending is used after 2qmfasta parses the proteomes
# (Should just leave as .fasta)
# (read by quickmine)
#

fasta_file_ending = "\.fasta"


#
# Where to write the website (all output)
#

# Note: use no trailing /
path2output = "/home/gawi/output"


#
# Where to read blast from
#

path2blast = "/home/gawi/output"



#
# Where to run the scripts from
#

path2scripts = "/home/gawi/quickmine"


#
# Path to files to be viewed over network
```

```
#

path2public = "/gawi/output"



#
# What record separator to use on all the output tables created?
#

#record_separator = ","
record_separator = "tab"

#
# Which parts of the pipeline to run?
#

# parse input files to rename headers and create the
#SELF_blast_database ?

parse = 1

# format the SELF_blast_database ?

format = 1

# run quickmine to do all the blast searches?
# Alternatively stop the pipeline at this point, run your blasts using
# condor, then continue from split_blast below

quickmine = 1

# split each genome blast file into individual files and place them in
# a genome specific directory.

split = 1

# parse all blast reports to determine numbers of hits ?

orphans = 1

# summarise these hits for each input proteome ?

hits = 1

# create a matrix of shared genes between all proteomes ?

genetable = 1

# detemine the number of orphans in each proteome ?

orphan_count = 1

# determine the size of orphans in each proteome and create fasta
files # containing the orphan sequences?

orphan_size = 1

# create list of paralogous orphans?

paralogue_count = 1

# modify "overview files" to see number of orphans decline over time ?
```

```
increment = 1

# modify "overview files" to see number of orphans decline over time ?

time = 1

# create binary matrix?

binary = 1

# All plot sections of pipeline require gnuplot to be installed
# create single plot containing all genomes?

plots = 1

# create a plot for each individual genome?

indiv_plot = 1

# run dot_plot.pl

dot_plot = 0

# create a final "index.html" file that summarised all the results ?

summarizer = 1



#
# Write individual fasta files (2qmfasta.pl)
#

write_fasta_files = 1


#
# keep this value set to 1
#

condor_output = 1


#
# Is the BLAST against the SELF_blast_database? (If in doubt leave as
#default value 1)
#

self_hit = 1


#
# Command to format the SELF_blast_database?
#

# Note: make sure correctly set for either a protein or a dna database

formatdb = "/usr/software/blast/blast/bin/formatdb -i
/home/gawi/quickmine_pack/sarah_output/SELF_blast_database -p T -o F"


#
# Command to run blast
#
```

```
blast_command = "/usr/local/bin/blastall -p blastp -d
SELF_blast_database -e 1e-3 -b 500 -f 9 -F 'mS' -M BLOSUM45"

#
# Significance threshold to use in detecting orphans? (get_orphans.pl
#script)
#

sig_thresh = 0.001


#
# Which file ending to use in the genetable script?
#


end = "_SELF_blastp_overview.html.hits.html"


#
# Which file ending to use in the orphan_count script?
#


count_end = "_SELF_blastp_overview.html"


#
# Which file ending to use in the orphan_time script?
#


time_end = "_orphan_increment.html"


#
# Which file ending to use in the dot_plot script?
#


matrix_end = "_SELF_blastp_matrix.html"


#
# Which cascading stylesheet to use?
#

stylesheet =
"http://darwin.nox.ac.uk/gawi/quickmine_pack/quickmineoutput.css"
```

Appendix 4.1 – OrphanMine orphandb_v2 SQL file

The SQL script below describes the tables created in OrphanMine and the indexes within those tables.

```
use orphandb_v2;
create table genome3    (NC_number char(29),
                  Publication int (4),
                  Genome_size float(8),
                  Orfs int(6),
                  Percent_lc float(5),
                  Gc_content float(5),
                  Species text,
                  Family text,
                  Division text,
                  Domain text,
                  Tax_id int (10),
                  Uniqueness text,
                  Genome_id int(4) not null primary key
auto_increment);

create table orf3 (Genome_id int(4),
                  Orf char(22),
                  Gi int(8),
                  Length int(5),
                  Description text,
                  Low_complexity float(5),
                  Length_percentile int(3),
                  E_value char(6),
                  Comp_evalue int(3),
                  Identity float(5),
                  Closest_hit char(16),
                  Gc float(5),
                  Ortholog int(3),
                  Start int(9),
                  Stop int(9),
                  Direction int(1),
                  length_rank int (3),
                  lc_rank int(3),
                  gc_diff_rank int(3),
                  cluster_rank int(3),
                  cost_diff_rank int(3),
                  cost_rank int(3),
                  gc_diff float(5),
                  nd float(20),
                  cost_diff float(20),
                  cost float(20),
                  gc_rank int(3),
                  Orf_id int(7) not null primary key auto_increment);

create table orphan3    (Orf_name char(22),
                  Dataset_number int(3),
                  True_para_orphan int(4),
                  Orphan_id int(7) not null primary key
auto_increment);

create table paths_dataset3   (Dataset_number int(3) not null primary
key auto_increment,
                        Blast text,
                        Seq_plot text);
```

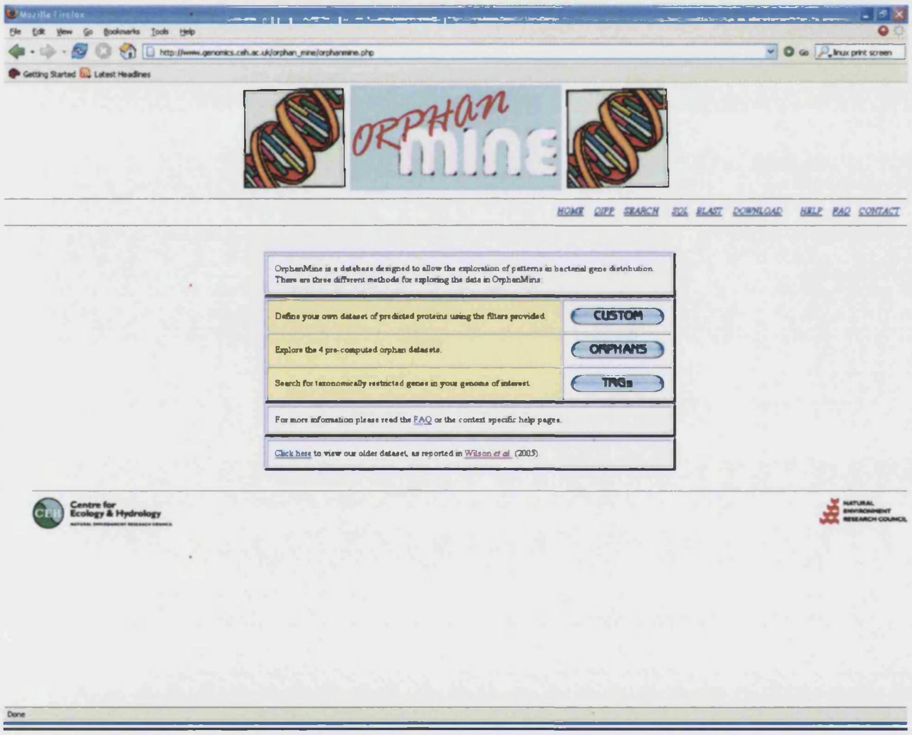
```
create table join_dataset3    (Dataset_number int(3),
                        Dataset_id int(5) not null primary key
auto_increment);


create table dataset3   (Genome_id int(4),
                  Orphans int(5),
                  Small int(5),
                  Large int(5),
                  Mean_size float(5),
                  Percent_orphans float(5),
                  True_paralogues int(5),
                  Iio float (15),
                  Dataset_id int(5) not null primary key
auto_increment);

create table Para_blast (NC_query char(29),
                  Orf_query char(22),
                  NC_hit char(29),
                  Orf_hit char(22),
                  E_value char(6),
                  Para_id int(10) not null primary key
auto_increment);

create table blast_summary    (Genome_id int(4),
                        Orf char(22),
                        NC_000907 int(1),
                        NC_000908 int(1),
                        NC_000911 int(1),
                        NC_00XXXX int (1), # require a column for each
genome in the dataset.
                        Blast_summ_id int(10) not null primary key
auto_increment);

create index index_on_nc on genome3(NC_number);
create index index_on_Orf on orf3 (Orf);
create index index_on_gi on orf3(Gi);
create index index_on_orf_genome_id on orf3(genome_id);
create index index_on_Orf_name on orphan3 (Orf_name);
create index index_on_genomeid on dataset3(genome_id);
create index index_on_nc_query on Para_blast(nc_query);
create index index_on_blast_Orf on blast_summary (Orf);
create index index_on_blast_genome_id on blast_summary(genome_id);

load data local infile
"/home/gawi/orphan_database/orphans_331/genome_table_330.txt" into
table genome3;
load data local infile
"/home/gawi/orphan_database/orphans_331/orf_table.txt" into table
orf3;
load data local infile
"/home/gawi/orphan_database/orphans_331/orphan_table_D1.txt" into
table orphan3;
load data local infile
"/home/gawi/orphan_database/orphans_331/dataset_table_1.txt" into
table dataset3;
load data local infile
"/home/gawi/orphan_database/orphans_331/dataset_1_paths.txt" into
table paths_dataset3;
load data local infile
"/home/gawi/orphan_database/orphans_331/join_dataset_table_1.txt" into
table join_dataset3;
```

```
load data local infile
"/home/gawi/orphan_database/orphans_331/para_blast_table_1" into table
Para_blast;
load data local infile
"/home/gawi/orphan_database/orphans_331/blast_summary_table.txt" into
table blast_summary;
```

Appendix 4.2 - OrphanMine Web Page Descriptions

orphanmine.php

- Provides user with the option to enter OrphanMine in one of 3 ways. They can choose Custom, in which they create their own gene dataset. They can choose Orphans, in which they explore one of the pre-generated orphan datasets. Alternatively they can choose TRGs, this allows the user to search for lineage-specific genes in their genomes of interest.
- Provides user with the ability to enter one of the general pages (Search, SQL, BLAST, Download, Help, FAQ and Contact). These pages remain constant regardless of the section of the site currently being utilised (the exception being the page specific help files discussed in 4.8.4).


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/orphanmine.php
Go
linux print screen
Getting Started Latest Headlines
ORPHAN mine
HOME OIPP SEARCH SQL BLAST DOWNLOAD HELP FAQ CONTACT
CUSTOM
ORPHANS
TRGs
Centre for Ecology & Hydrology
NATURAL ENVIRONMENT RESEARCH COUNCIL
Done

OrphanMine is a database designed to allow the exploration of patterns in bacterial gene distribution. There are three different methods for exploring the data in OrphanMine:

Define your own dataset of predicted proteins using the filters provided.

Explore the 4 pre-computed orphan datasets.

Search for taxonomically restricted genes in your genome of interest.

For more information please read the FAQ or the context specific help pages.

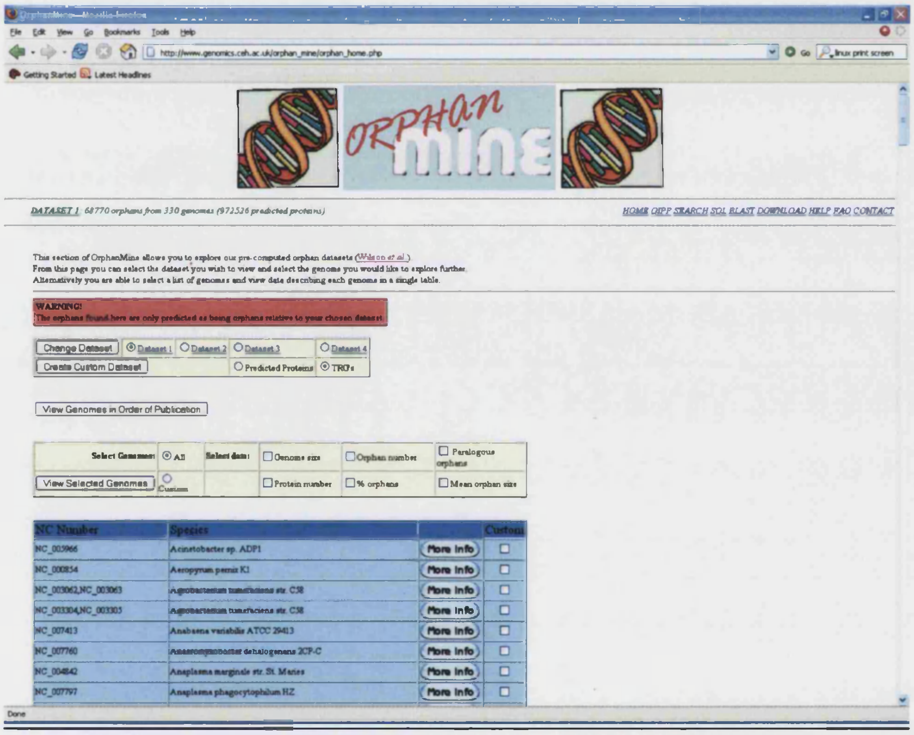
Click here to view our older dataset, as reported in Wilson *et al.* (2005).

orphanmine.php Implementation

On entry to the site the user will be directed to *orphanmine.php*, known as Home. The top section of the page is coded for by the script *header1.php* and is included in every page in the OrphanMine system. In addition to the OrphanMine logo (which contains a link back to *orphanmine.php*), *header1.php* codes for the navigation bar, from which the user can enter various parts of the system. The navigation bar remains constant throughout the system thus providing freedom of movement to the user in a familiar style. Three large buttons are found centrally on the page, these are labelled Custom, Orphans and TRGs and direct the user to *customise.php*, *orphan_home.php* and *restriction_v2.php*, respectively.

orphan_home.php

- Allows the user to view details about the pre-generated orphan datasets and select the dataset they wish to explore.
- Displays a list of the genomes included in the currently selected orphan dataset. The list can be ordered alphabetically or chronologically. Each genome has a 'More Info' button associated.
- Permits the user to select a subset of genomes and view associated data such as Genome size and Orphan number in one table.



DATASET 1: 68770 orphans from 330 genomes (972526 predicted proteins)

This section of OrphanMine allows you to explore our pre-computed orphan datasets (Wilson et al.)
From this page you can select the dataset you wish to view and select the genome you would like to explore further.
Alternatively you are able to select a list of genomes and view data describing each genome in a single table.

WARNING!
The orphans found here are only predicted as being orphans relative to your chosen dataset

Change Dataset	● Dataset 1	○ Dataset 2	○ Dataset 3	○ Dataset 4
Create Custom Dataset			○ Predicted Proteins	● TRGs

View Genomes in Order of Publication

Select Genomes:	● All	Select data:	□ Genome size	□ Orphan number	□ Paralogous orphans
View Selected Genomes	○ Custom		□ Protein number	□ % orphans	□ Mean orphan size

NC Number	Species		Custom
NC_005966	Acinetobacter sp. ADP1	More Info	□
NC_000854	Aeropyrum pernix K1	More Info	□
NC_003062,NC_003063	Agrobacterium tumefaciens str. C58	More Info	□
NC_003304,NC_003305	Agrobacterium tumefaciens str. C58	More Info	□
NC_007413	Anabaena variabilis ATCC 29413	More Info	□
NC_007760	Anaeromyxobacter dehalogenans 2CP-C	More Info	□
NC_004842	Anaplasma marginale str. St. Maries	More Info	□
NC_007797	Anaplasma phagocytophilum HZ	More Info	□

Done

orphan_home.php Implementation

The information in the table is obtained by performing a query involving the MySQL tables Genome3 and Dataset3. The query determines which genomes should be included in the table. The user can alter the query by selecting a different dataset. The final column in the table of genomes contains a checkbox, this allows the user to select that particular genome so that it appears in the output of *compare.php*. Clicking on the 'More Info' button leads to *genome_info.php*, generated for the genome selected by the user (due to the relevant Genome_id being passed in the URL as var).

compare.php

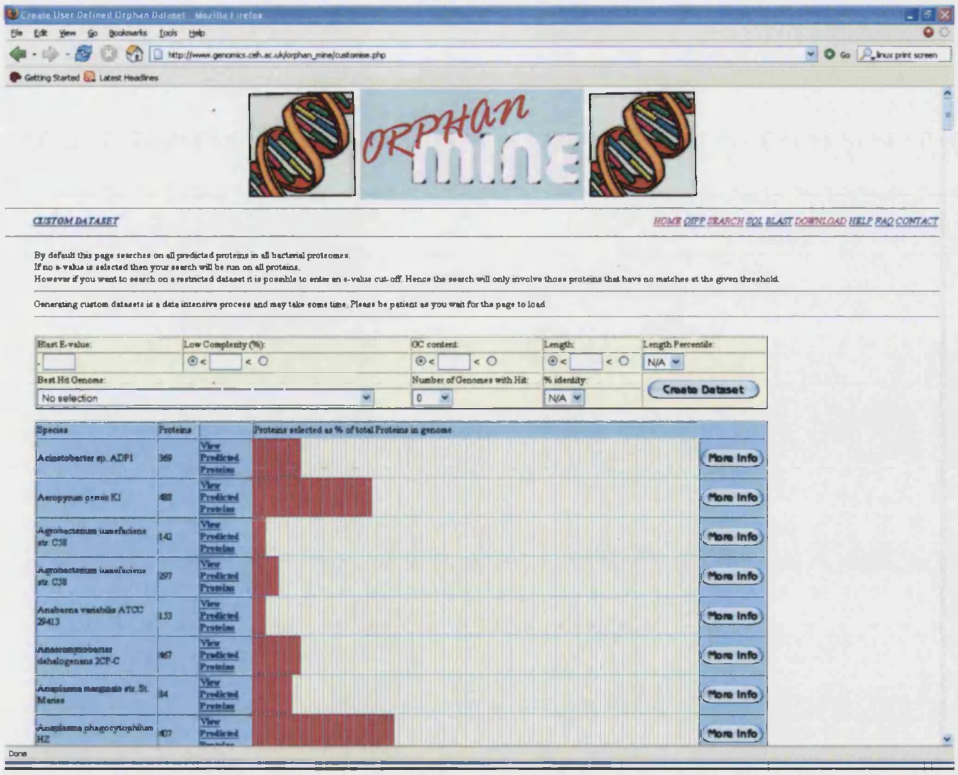
- Provides the user with the list of genomes they selected in the checkboxes on *orphan_home.php* plus the data they selected to describe those genomes.
- Provides a 'More Info' button for each genome.

compare.php Implementation

The genome_id for each selected genome is passed from *orphan_home.php* to *compare.php* in an array via the POST method. The data types to be shown are also passed using this method. An SQL query is performed for each genome_id in the array and the results displayed in the HTML table.

customise.php

- Provides the user with the option of creating their own dataset of genes by selecting from a number of different parameters (E-value, low complexity, GC content, length, length percentile, best hit genome, number of genomes with a hit and percent identity). These parameters can be combined.
- Displays a table listing all the genomes, the number of predicted proteins in each genome that fit the user defined criteria and a bar chart illustrating the number of predicted proteins that fit the criteria as a percentage of the total number of predicted proteins in the genome.
- Provides links to enable the user to view the predicted proteins matching the criteria within a given genome and also view genomic level information with the predicted proteins matching the criteria.


Create User Defined Orphan Dataset - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/customise.php
Getting Started Latest Headlines
ORPHAN MINE
CUSTOM DATASET
HOME OIPP SEARCH SQL BLAST DOWNLOAD HELP FAQ CONTACT
Done

By default this page searches on all predicted proteins in all bacterial proteomes.
If no e-value is selected then your search will be run on all proteins.
However if you want to search on a restricted dataset it is possible to enter an e-value cut-off. Hence the search will only involve those proteins that have no matches at the given threshold.

Generating custom datasets is a data intensive process and may take some time. Please be patient as you wait for the page to load.

Blast E-value:	Low Complexity (%):	GC content:	Length:	Length Percentile:
-	● < ☐ < ○	● < ☐ < ○	● < ☐ < ○	N/A
Best Hit Genome:		Number of Genomes with Hit:	% identity:	Create Dataset
No selection		0	N/A	

Species	Proteins		Proteins selected as % of total Proteins in genome	
Acinetobacter sp. ADP1	369	View Predicted Proteins		More Info
Aeropyrum pernix K1	488	View Predicted Proteins		More Info
Agrobacterium tumefaciens str. C58	142	View Predicted Proteins		More Info
Agrobacterium tumefaciens str. C58	297	View Predicted Proteins		More Info
Anabaena variabilis ATCC 29413	153	View Predicted Proteins		More Info
Anaeromyxobacter dehalogenans 2CP-C	467	View Predicted Proteins		More Info
Anaplasma marginale str. St. Maries	84	View Predicted Proteins		More Info
Anaplasma phagocytophilum HZ	407	View Predicted Proteins		More Info

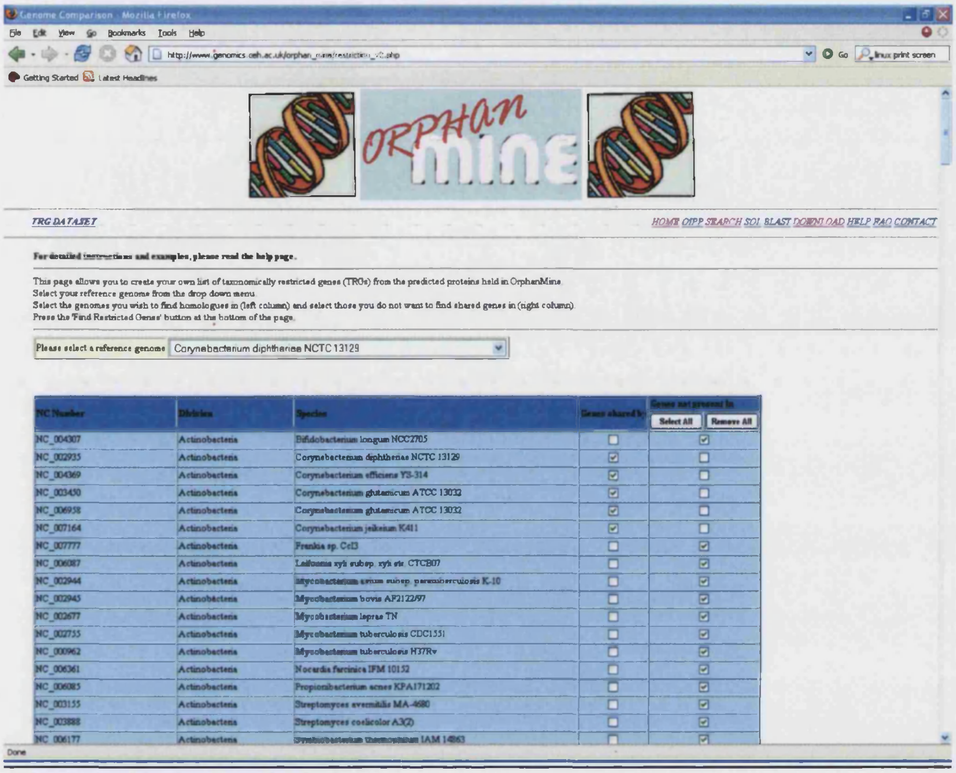
customise.php Implementation

When the user initially loads the page, a query is performed against the Genome3 and Orf3 MySQL tables. This determines the number of predicted proteins there are in each genome when there are no restrictions imposed. Each criteria type has an associated form element allowing the user to enter their specific thresholds. To apply these thresholds, the button 'Create Dataset' should be selected. This causes the page to reload, however this time the user input is stored as session variables and integrated into the SQL query. Before the query is performed, each element entry is checked to ensure the user has entered an expected value e.g. numerical. If the value fails the check, an error is reported back to the user. The bar chart is constructed from 100 table elements. The number of elements filled red is calculated in the PHP once the query has been resolved. The user may follow the link to 'View Predicted Proteins' or 'More Info', with their selected thresholds stored as session variables for use in other PHP scripts.

restriction v2.php

- Provides the user with a drop down menu to select a reference genome with which to compare other genomes against.
- Displays a list of the genomes held in the database. Two checkboxes are associated with each genome, only one of which can be selected for each genome.

By selecting the left hand checkbox, the user is selecting to view those genes in the reference genome that are shared with the selected genome. By selecting the right hand checkbox, the user is selecting to view those genes in the reference genome that are not present in the selected genome.


Genome Comparison - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/restriction_v2.php
Go
linux print screen
Getting Started
Latest Headlines
ORPHAN MINE
TRG DATASET
HOME OIPP SEARCH SQL BLAST DOWNLOAD HELP FAQ CONTACT
Done

For detailed instructions and examples, please read the help page.

This page allows you to create your own list of taxonomically restricted genes (TRGs) from the predicted proteins held in OrphanMine.
Select your reference genome from the drop down menu.
Select the genomes you wish to find homologues in (left column) and select those you do not want to find shared genes in (right column).
Press the 'Find Restricted Genes' button at the bottom of the page.

Please select a reference genome: Corynebacterium diphtheriae NCTC 13129

NC Number	Division	Species	Genes shared by	Genes not present in Select All Remove All
NC_004307	Actinobacteria	Bifidobacterium longum NCC2705	□	☑
NC_002935	Actinobacteria	Corynebacterium diphtheriae NCTC 13129	☑	□
NC_004369	Actinobacteria	Corynebacterium efficiens YS-314	☑	□
NC_003450	Actinobacteria	Corynebacterium glutamicum ATCC 13032	☑	□
NC_006958	Actinobacteria	Corynebacterium glutamicum ATCC 13032	☑	□
NC_007164	Actinobacteria	Corynebacterium jeikeium K411	☑	□
NC_007777	Actinobacteria	Frankia sp. CcI3	□	☑
NC_006087	Actinobacteria	Leifsonia xyli subsp. xyli str. CTCB07	□	☑
NC_002944	Actinobacteria	Mycobacterium avium subsp. paratuberculosis K-10	□	☑
NC_002945	Actinobacteria	Mycobacterium bovis AF2122/97	□	☑
NC_002677	Actinobacteria	Mycobacterium leprae TN	□	☑
NC_002755	Actinobacteria	Mycobacterium tuberculosis CDC1551	□	☑
NC_000962	Actinobacteria	Mycobacterium tuberculosis H37Rv	□	☑
NC_006361	Actinobacteria	Nocardia farcinica IFM 10152	□	☑
NC_006085	Actinobacteria	Propionibacterium acnes KPA171202	□	☑
NC_003155	Actinobacteria	Streptomyces avermitilis MA-4680	□	☑
NC_003888	Actinobacteria	Streptomyces coelicolor A3(2)	□	☑
NC_006177	Actinobacteria	Symbiobacterium thermophilum IAM 14863	□	☑

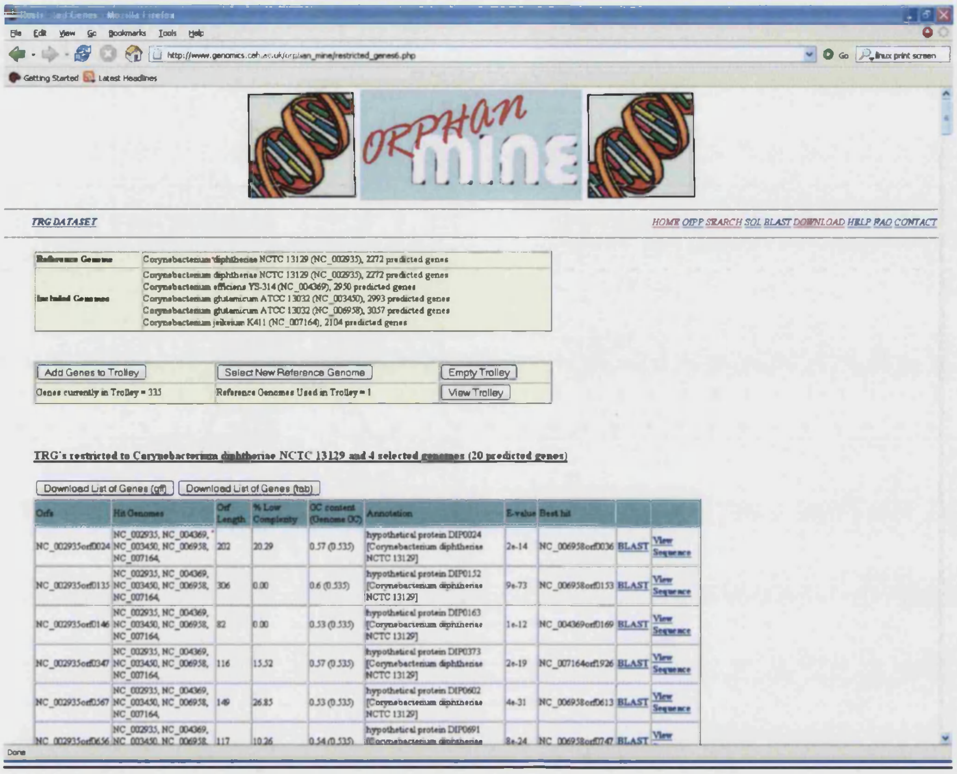
restriction_v2.php Implementation

This page uses Javascript to attempt to prevent the user from selecting a combination of boxes that will return an error message. When the user selects a reference genome from the drop down menu, the 'Shared by' checkbox associated with that genome will automatically be ticked. Javascript is also used to prevent the user from selecting both the 'Shared by' checkbox and the 'Not Present in' checkbox for the same genome. Additionally, if the user tries to submit before selecting a reference genome, a Javascript box will pop-up and prevent them from proceeding. On submission, the relevant data is passed in arrays to restricted_genes6.php.

restricted_genes6.php

- Provides a list of the genes and their associated metadata that matched the user generated query from *restriction_v2.php*.
- Provides the option to view the sequence of a gene of interest, or to BLAST the gene of interest against a selection of databases.

- Allows the user to download the list of genes in tab-delimited or GFF format.
- Provides a trolley facility. Users may add genes to their trolley, empty their trolley, view the contents of their trolley or alternatively select a new reference genome.


File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/restricted_genes6.php
Getting Started Latest Headlines
ORPHAN mine
HOME ORPP SEARCH SQL BLAST DOWNLOAD HELP FAQ CONTACT
Add Genes to Trolley
Select New Reference Genome
Empty Trolley
View Trolley
Download List of Genes (gff)
Download List of Genes (tab)
Done

TRG DATASET

Reference Genome	Corynebacterium diphtheriae NCTC 13129 (NC_002935), 2272 predicted genes
Included Genomes	Corynebacterium diphtheriae NCTC 13129 (NC_002935), 2272 predicted genes Corynebacterium efficiens YS-314 (NC_004369), 2950 predicted genes Corynebacterium glutamicum ATCC 13032 (NC_003450), 2993 predicted genes Corynebacterium glutamicum ATCC 13032 (NC_006958), 3057 predicted genes Corynebacterium jeikeium K411 (NC_007164), 2104 predicted genes

Genes currently in Trolley = 335

Reference Genomes Used in Trolley = 1

TRG's restricted to Corynebacterium diphtheriae NCTC 13129 and 4 selected genomes (20 predicted genes)

Orfs	Hit Genomes	Orf Length	% Low Complexity	GC content (Genome GC)	Annotation	E-value	Best hit		
NC_002935orf0024	NC_002935, NC_004369, NC_003450, NC_006958, NC_007164,	202	20.29	0.57 (0.535)	hypothetical protein DIP0024 [Corynebacterium diphtheriae NCTC 13129]	2e-14	NC_006958orf0036	BLAST	View Sequence
NC_002935orf0135	NC_002935, NC_004369, NC_003450, NC_006958, NC_007164,	306	0.00	0.6 (0.535)	hypothetical protein DIP0152 [Corynebacterium diphtheriae NCTC 13129]	9e-73	NC_006958orf0153	BLAST	View Sequence
NC_002935orf0146	NC_002935, NC_004369, NC_003450, NC_006958, NC_007164,	82	0.00	0.53 (0.535)	hypothetical protein DIP0163 [Corynebacterium diphtheriae NCTC 13129]	1e-12	NC_004369orf0169	BLAST	View Sequence
NC_002935orf0347	NC_002935, NC_004369, NC_003450, NC_006958, NC_007164,	116	15.52	0.57 (0.535)	hypothetical protein DIP0373 [Corynebacterium diphtheriae NCTC 13129]	2e-19	NC_007164orf1926	BLAST	View Sequence
NC_002935orf0567	NC_002935, NC_004369, NC_003450, NC_006958, NC_007164,	149	26.85	0.53 (0.535)	hypothetical protein DIP0602 [Corynebacterium diphtheriae NCTC 13129]	4e-31	NC_006958orf0613	BLAST	View Sequence
NC_002935orf0656	NC_002935, NC_004369, NC_003450, NC_006958,	117	10.26	0.54 (0.535)	hypothetical protein DIP0691 [Corynebacterium diphtheriae	8e-24	NC_006958orf0747	BLAST	View

restricted_genes6.php Implementation

The NC number of the reference genome, the genomes that have shared genes with the reference genome and those that do not have the same genes as the reference genome are passed from *restriction_v2.php* in arrays. This data is used to perform the necessary queries. The MySQL table Blast_summary is central to the functionality of this page. This table contains data showing which genomes contain matches to which genes, therefore it is quick to query and obtain the lists of genes that match the user requirements.

If the user chooses to download the list of genes, the scripts *download_trgs.php* and *download_trgs_tab.php* are called. These scripts are never seen by the user but are responsible for managing the download of the data.

In order to implement the trolley functionality, PHP session variables were utilised. This enables the tool to keep track of what genes the user is interested in, allowing the user to add more than one reference genome to the trolley. The most important session variables are trolley and ref_trolley. Trolley stores all the identifiers for the genes stored in the trolley and ref_trolley stores all the NC numbers of the reference genomes whose genes are stored in the trolley. If the user chooses to empty the trolley, the session variables are unregistered. If the user selects to view the trolley, the necessary data is passed on to *trolley.php.*

trolley.php

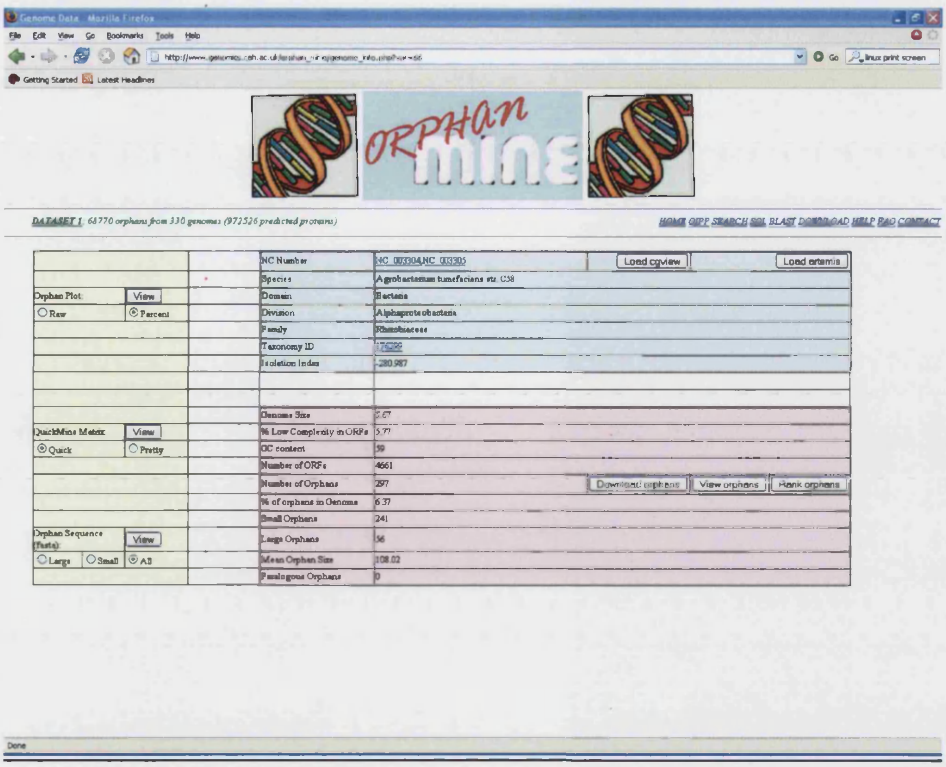
- Lists the NC number and species names of the reference genomes which have genes stored in the trolley. Also shows the number of genes from those reference genomes that are in the trolley. Provides a 'More Info' button.
- Each reference genome in the trolley has a checkbox associated with it. By selecting the checkbox and clicking on the 'Download Orphans' button, the user can download the protein sequences of the genes in their dataset.

trolley.php Implementation

This PHP script utilises the data passed in the trolley and ref_trolley session variables in order to make the necessary SQL queries. The 'More Info' button causes the *trg_info.php* page to load. By selecting to download the protein sequences of their genes of interest, the user initiates the script *downloading.php.* The header of the file generated by *downloading.php* is set so that a plain text file is produced as a file for download. The GI identifiers for each of the proteins whose sequence is required is obtained from the MySQL table Orf3. This value is used by the programme *fastacmd. Fastacmd* searches an indexed version of the BLAST database generated in QuickMine for the retrieved GI number. The output from *fastacmd* is the relevant protein sequence in FASTA format.

genome_info.php, custom_genome.php & trg_info.php

- Provides the user with a summary of the information regarding their genome of interest, specific to the dataset they are using.
- Acts as a starting point for accessing much of the genome specific data. For example the orphan plots (*genome_info.php* only), QuickMine matrix, paralogous orphans (*genome_info.php* only) and protein sequences for the genes of interest.
- Allows the user to access pages to view their genes, rank their genes and permits downloading of the genes in GFF format.
- Provides access to the Artemis Webstart and the CGView applet.
- Provides links to the GenomeBank and to NCBI taxonomy.



DATASET 1: 68770 orphans from 330 genomes (973526 predicted proteins)

Orphan Plot: View
○ Raw ● Percent

QuickMine Matrix: View
● Quick ○ Pretty

Orphan Sequence (fasta): View
○ Large ○ Small ● All

NC Number	NC_003304,NC_003305
Species	Agrobacterium tumefaciens str. C58
Domain	Bacteria
Division	Alphaproteobacteria
Family	Rhizobiaceae
Taxonomy ID	176299
Isolation Index	-280.987
Genome Size	5.67
% Low Complexity in ORFs	5.77
GC content	59
Number of ORFs	4661
Number of Orphans	297
% of orphans in Genome	6.37
Small Orphans	241
Large Orphans	56
Mean Orphan Size	108.02
Paralogous Orphans	0

genome_info.php, custom_genome.php & trg_info.php Implementation

The main function of these pages is to extract all the information describing a specific genome within a specified dataset. The method used for extracting the data differs according to the PHP page. *Genome_info.php* extracts the data from MySQL tables Genome3 and Dataset3. *Custom_genome.php* uses the values selected to generate the dataset (from *customise.php* and saved as session variables) to interrogate the Orf3 table and count the output, in addition to extracting dataset independent information from Genome3. *Trg_info.php* utilises the genes stored in the trolley session variable, and the NC numbers stored in the ref_trolley session variable, to calculate the dataset specific data. This is done using a combination of the tables Genome3 and Orf3.

The information is displayed in tables and split into categories and colour coded. The taxonomic attributes of a genome, such as Species, Taxonomy ID and Isolation Index are displayed in a green table. The data describing the genomic content, for example, number of predicted proteins and genome size are shown in a pink table. Buttons linking to output from QuickMine, such as orphan plots, are shown in a yellow table. The colours chosen to represent the groups are all pastel shades; this is so that the users do not mistake the colours for warnings or error messages which may occur with sharper shades. By using pastel shades the colours are clearly present to differentiate the categories and break the information down into more digestible slices (prevent 'information overload' (Rechenmann, 1995)).

In addition to providing users with a route for selecting numerous pages, it also provides radio buttons for the user to decide how they would like some of the output to be formatted. Users can choose to view the orphan plot using raw numbers or percentages on the axis. They can choose to view the QuickMine matrix in a text format (quick to view and download for use in e.g. Microsoft Excel) or HTML format (longer to download but easier to view online). They may also choose to view all the sequences of the genes of interest, or view just the short (<150 amino acids in length) or the long (>=150 amino acids in length) sequences. The option to download the genes contained in their chosen dataset in GFF format is also provided. By pressing this button, the script *download_gff3.php* is loaded. This is not seen by the user, but is responsible for the necessary data being written to file.

orphan.php, custom_orphans.php & trg_list.php

- Provides a list of the predicted proteins in the current dataset along with their associated metadata.
- Allows the user to view a sequence of interest or BLAST a sequence of interest against a database.
- Allows users to download the predicted proteins in GFF format.

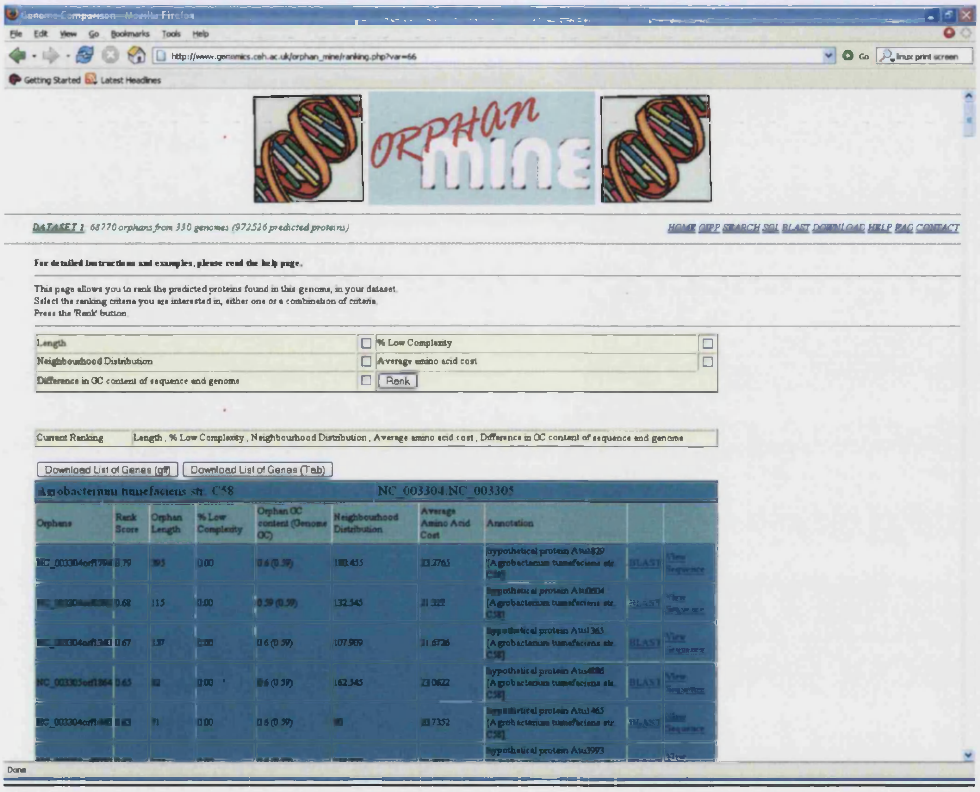
orphan.php, custom_orphans.php & trg_list.php Implementation

These pages provide a list of the predicted proteins within a specified dataset. The method used to extract the data differs according to the PHP page. *Orphan.php* obtains its data from the MySQL tables Orf3 and Orphan3. In contrast, both *custom_orphans.php* and *trg_list.php* only query Orf3, utilising the stored session variables to extract the correct data. By clicking on the 'Download' button, the script *download_gff3.php* is run. If the user elects to view the sequence, they will be directed to *fastacmd.php.* Alternatively, if they choose to BLAST, *blast.cgi* will load.

ranking.php, custom_ranking.php & trg_ranking.php

- Provides a method for ranking the predicted proteins, according to which predicted protein is more likely to be expressed and therefore be real. The score and rank is dependent on the criteria selected by the user. The criteria are selected by filling checkboxes.

- Provides a list of the predicted proteins, ordered by their rank score, in the current dataset along with their associated metadata and the rank score.
- Allows user to view a sequence of interest or BLAST a sequence of interest against a database.
- Allows users to download the predicted proteins in GFF format with additional information regarding their score and the criteria used for ranking.


Genome Comparison – Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/ranking.php?var=66
Go
linux print screen
Getting Started
Latest Headlines
ORPHAN mine
HOME OIPP SEARCH SQL BLAST DOWNLOAD HELP FAQ CONTACT
Done

DATASET 1 68770 orphans from 330 genomes (972526 predicted proteins)

For detailed instructions and examples, please read the help page.

This page allows you to rank the predicted proteins found in this genome, in your dataset.
Select the ranking criteria you are interested in, either one or a combination of criteria.
Press the 'Rank' button.

Length	□	% Low Complexity	□
Neighbourhood Distribution	□	Average amino acid cost	□
Difference in GC content of sequence and genome	□	Rank	

Current Ranking	Length , % Low Complexity , Neighbourhood Distribution , Average amino acid cost , Difference in GC content of sequence and genome

Download List of Genes (gff) Download List of Genes (Tab)

Agrobacterium tumefaciens str. C58 NC_003304.NC_003305

Orphans	Rank Score	Orphan Length	% Low Complexity	Orphan GC content (Genome GC)	Neighbourhood Distribution	Average Amino Acid Cost	Annotation		
NC_003304orf1794	0.79	393	0.00	0.6 (0.59)	180.455	23.2765	hypothetical protein Atu1829 [Agrobacterium tumefaciens str. C58]	BLAST	View Sequence
[illegible]	0.68	115	0.00	0.59 (0.59)	132.545	21.327	hypothetical protein Atu0804 [Agrobacterium tumefaciens str. C58]	BLAST	View Sequence
NC_003304orf1340	0.67	137	0.00	0.6 (0.59)	107.909	21.6726	hypothetical protein Atu1363 [Agrobacterium tumefaciens str. C58]	BLAST	View Sequence
NC_003305orf1864	0.65	82	0.00	0.6 (0.59)	162.545	23.0622	hypothetical protein Atu[illegible] [Agrobacterium tumefaciens str. C58]	BLAST	View Sequence
NC_003304orf1[illegible]	0.63	71	0.00	0.6 (0.59)	90	23.7352	hypothetical protein Atu1465 [Agrobacterium tumefaciens str. C58]	BLAST	View Sequence
							hypothetical protein Atu3993		View

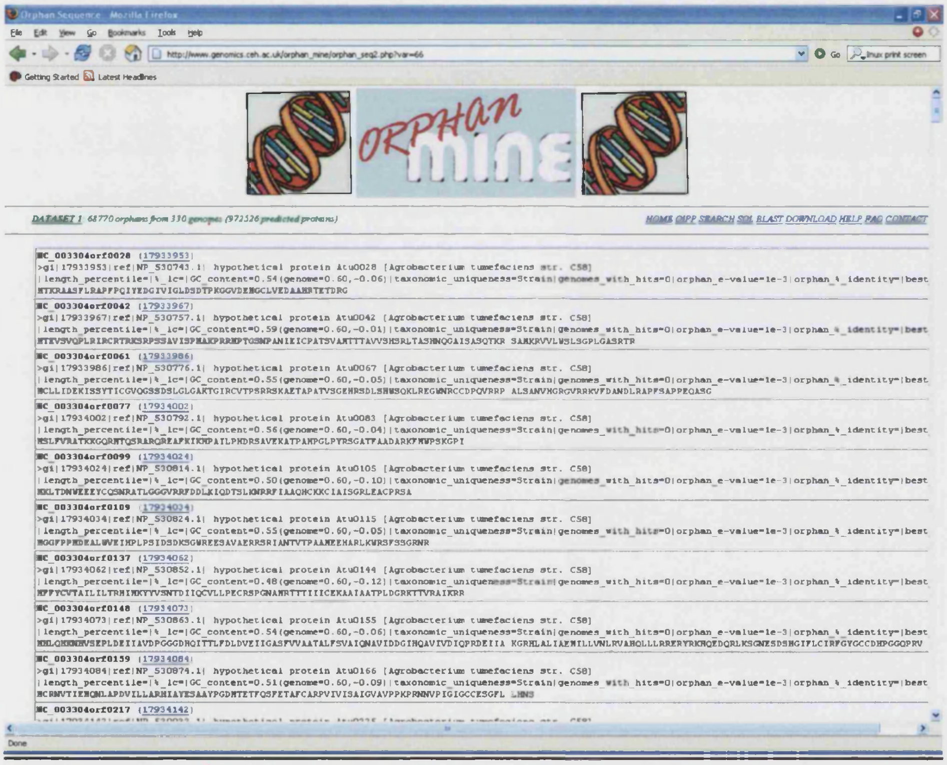
ranking.php, custom_ranking & trg_ranking.php Implementation

These pages provide a list of the predicted proteins within a specified dataset. The method used to extract the data differs according to the PHP page. *Ranking.php* obtains its data from the MySQL tables Orf3 and Orphan3. In contrast, both *custom_ranking.php* and *trg_ranking.php* only query Orf3, utilising the stored session variables to extract the correct data. When the page is loaded, it checks to determine how many criteria have been selected to rank on. Initially this is zero, so the list of predicted proteins is provided in numerical order. When the user selects to rank on a particular criteria, or combination of criteria, the page reloads. Each criterion has a corresponding column in the MySQL table Orf3. In this column, a figure between 0-100 is given. This figure, for all the criteria selected, is obtained and summed. The total is then divided by the number of criteria selected (therefore will be between 0-100 again) and divided by 100. The final score is between 0 and 1. The predicted proteins are sorted according to their score and printed to HTML in the correct order. For more information regarding the ranking method used in OrphanMine, see Chapter 5.

The output can be downloaded in GFF or tab-delimited format. These files are generated by the *download_list.php* and *download_list_tab.php* scripts. If the user elects to view the sequence, they will be directed to *fastacmd.php*. Alternatively if they choose to BLAST, *blast.cgi* will load.

fastacmd.php, orphan_seq2.php, custom_seq.php & trg_seq.php

- Displays the protein sequence or sequences of the selected protein or proteins.


Orphan Sequence - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/orphan_seq2.php?var=66
Go
linux print screen
Getting Started Latest Headlines
ORPHAN mine
DATASET 1 68770 orphans from 330 genomes (972526 predicted proteins)
HOME OIPP SEARCH SQL BLAST DOWNLOAD HELP FAQ CONTACT
Done

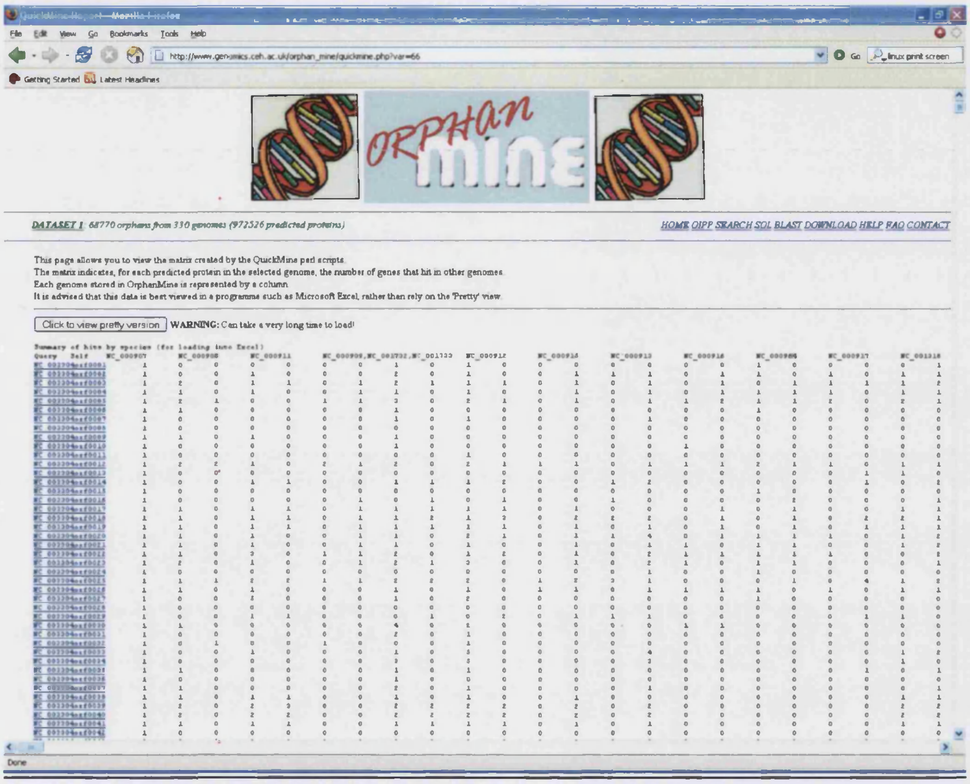
```
NC_003304orf0028 (17933953)
>gi|17933953|ref|NP_530743.1| hypothetical protein Atu0028 [Agrobacterium tumefaciens str. C58]
|length_percentile=|%_lc=|GC_content=0.54(genome=0.60,-0.06)|taxonomic_uniqueness=Strain|genomes_with_hits=0|orphan_e-value=1e-3|orphan_%_identity=|best
MTKRAASFLRAPFPQIYEDGIVIGLDSDTPKGGVDENGCLVEDAANRTETDRG
NC_003304orf0042 (17933967)
>gi|17933967|ref|NP_530757.1| hypothetical protein Atu0042 [Agrobacterium tumefaciens str. C58]
|length_percentile=|%_lc=|GC_content=0.59(genome=0.60,-0.01)|taxonomic_uniqueness=Strain|genomes_with_hits=0|orphan_e-value=1e-3|orphan_%_identity=|best
MTEVSVQPLRIRCRTRKSRPSSAVISPMAKPRRMPTGSNPANIKICPATSVANTTTAVVSHSRLTASHNQGAISASQTKR SAMKRVVLWSLSGPLGASRTR
NC_003304orf0061 (17933986)
>gi|17933986|ref|NP_530776.1| hypothetical protein Atu0067 [Agrobacterium tumefaciens str. C58]
|length_percentile=|%_lc=|GC_content=0.55(genome=0.60,-0.05)|taxonomic_uniqueness=Strain|genomes_with_hits=0|orphan_e-value=1e-3|orphan_%_identity=|best
MCLLIDEKISSYTICGVQGSSDSLGLGAKTGIRCVTPSRRSKAETAPATVSGEHRSDLSHWSQKLREGWNRCCDPQVRRP ALSANVHGRGVRRKVFDANDLRAPFSAPPEQASG
NC_003304orf0077 (17934002)
>gi|17934002|ref|NP_530792.1| hypothetical protein Atu0083 [Agrobacterium tumefaciens str. C58]
|length_percentile=|%_lc=|GC_content=0.56(genome=0.60,-0.04)|taxonomic_uniqueness=Strain|genomes_with_hits=0|orphan_e-value=1e-3|orphan_%_identity=|best
MSLFVRATKKGQRNTQSRARQREAPKIKHPAILPHDRSAVEKATPANPGLPYRSGATFAADARKFMWPSKGPI
NC_003304orf0099 (17934024)
>gi|17934024|ref|NP_530814.1| hypothetical protein Atu0105 [Agrobacterium tumefaciens str. C58]
|length_percentile=|%_lc=|GC_content=0.50(genome=0.60,-0.10)|taxonomic_uniqueness=Strain|genomes_with_hits=0|orphan_e-value=1e-3|orphan_%_identity=|best
MKLTDNWEEEYCQSNRATLGGGVRRFDDLKIQDTSLKNRRFIAAQHCKKCIAISGRLEACPRSA
NC_003304orf0109 (17934034)
>gi|17934034|ref|NP_530824.1| hypothetical protein Atu0115 [Agrobacterium tumefaciens str. C58]
|length_percentile=|%_lc=|GC_content=0.55(genome=0.60,-0.05)|taxonomic_uniqueness=Strain|genomes_with_hits=0|orphan_e-value=1e-3|orphan_%_identity=|best
MGGFPPMDEALWVEIHPLPSIDSDKSGWREESAVAERRSRIANTVTPAAMEEHARLKWRSFSSGRNR
NC_003304orf0137 (17934062)
>gi|17934062|ref|NP_530852.1| hypothetical protein Atu0144 [Agrobacterium tumefaciens str. C58]
|length_percentile=|%_lc=|GC_content=0.48(genome=0.60,-0.12)|taxonomic_uniqueness=Strain|genomes_with_hits=0|orphan_e-value=1e-3|orphan_%_identity=|best
MFFYCVTAILILTRHIMKYYVSNTDIIQCVLLPECRSPGNANRTTTIIICEKAAIAATPLDGRKTTVRAIKRR
NC_003304orf0148 (17934073)
>gi|17934073|ref|NP_530863.1| hypothetical protein Atu0155 [Agrobacterium tumefaciens str. C58]
|length_percentile=|%_lc=|GC_content=0.54(genome=0.60,-0.06)|taxonomic_uniqueness=Strain|genomes_with_hits=0|orphan_e-value=1e-3|orphan_%_identity=|best
MNLQMKMMHVSEPLDEIIAVDPGGGDHQITTLFDLDVEIIGASFVVAATALFSVAIQNAVIDDGIHQAVIVDIQPRDEIIA KGRHLALIAEMILLVNLRVAHQLLLRRERYRKHQEDQRLKSGNESDSHGIFLCIRFGYGCCDHPGGQPRV
NC_003304orf0159 (17934084)
>gi|17934084|ref|NP_530874.1| hypothetical protein Atu0166 [Agrobacterium tumefaciens str. C58]
|length_percentile=|%_lc=|GC_content=0.51(genome=0.60,-0.09)|taxonomic_uniqueness=Strain|genomes_with_hits=0|orphan_e-value=1e-3|orphan_%_identity=|best
MCRNVTIEMQNLAPDVILLARHIAYESAAYPGDNTETFQSFETAFCARPVIVISAIGVAVPPKPRNNVPIGIGCCESGFL LHNS
NC_003304orf0217 (17934142)
```

fastacmd.php, orphan_seq2.php, custom_seq.php & trg_seq.php Implementation

In these PHP scripts, the GI value from MySQL table Orf3 is obtained for each of the proteins that the user is wishing to view. The programme *fastacmd* then searches an indexed version of the BLAST database generated in QuickMine for the retrieved GI number. The output of *fastacmd*, i.e., the protein sequence with its FASTA header, is displayed in HTML.

quickmine.php

- Provides users with the opportunity to view and download the matrix on which much of OrphanMine is based.
- The data can be viewed in simple text format ready for download or in a HTML table.



File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/quickmine.php?var=66
Getting Started Latest Headlines
ORPHAN MINE
Done

DATASET 1: 68770 orphans from 330 genomes (972526 predicted proteins)

HOME OIPP SEARCH SQL BLAST DOWNLOAD HELP FAQ CONTACT

This page allows you to view the matrix created by the QuickMine perl scripts.
The matrix indicates, for each predicted protein in the selected genome, the number of genes that hit in other genomes.
Each genome stored in OrphanMine is represented by a column.
It is advised that this data is best viewed in a programme such as Microsoft Excel, rather than rely on the 'Pretty' view.

Click to view pretty version **WARNING:** Can take a very long time to load!

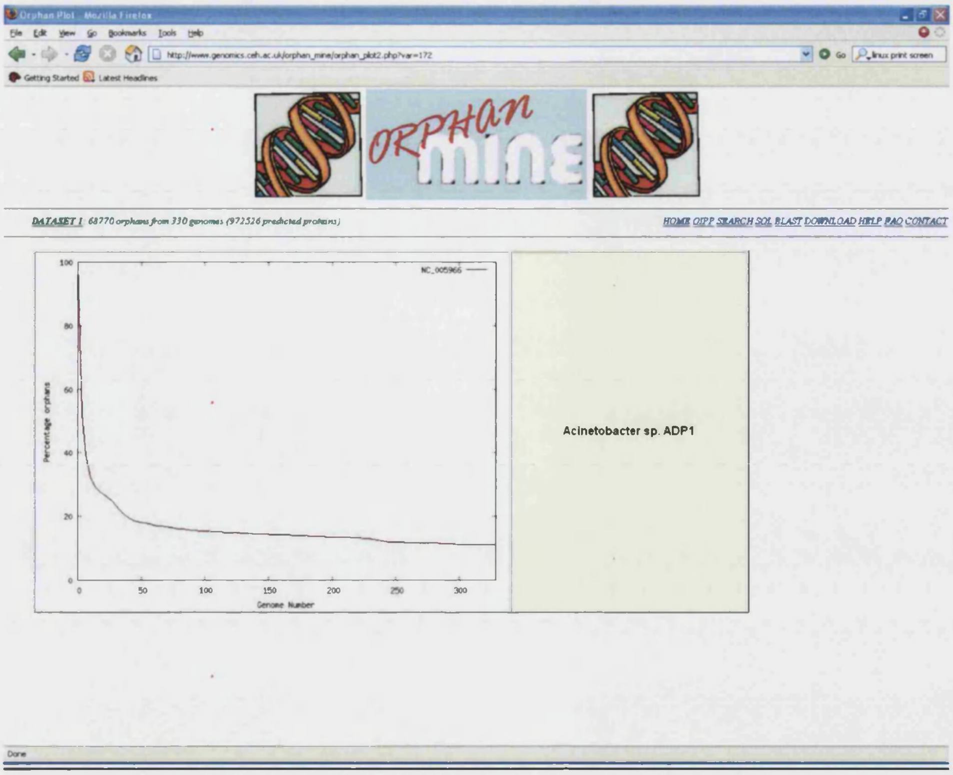
```
Summary of hits by species (for loading into Excel)
Query   Self   NC_000907        NC_000908        NC_000911        NC_000909,NC_001732,NC_001733   NC_000912        NC_000915        NC_000913        NC_000916        NC_000964        NC_000917        NC_001318
```

quickmine.php Implementation

Depending on the user's choice, a variable is passed to *quickmine.php* that determines what format the matrix will be printed in. If the user wants to view the text version, the required overview.html file is retrieved from the dbase server. If the 'pretty' view is required the overview_table.html file is retrieved from the dbase server. This file takes much longer to parse and load. The file is parsed to add a link to the *blast.cgi* page for each predicted protein.

orphan_plot2.php

- Displays a plot showing the change in orphan number over time in the selected genome for the current dataset. The user can select to view the plot in raw data form or with the data converted to percentage.


Orphan Plot - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/orphan_plot2.php?var=172
Go
linux print screen
Getting Started
Latest Headlines
ORPHAN mine
HOME OIPP SEARCH SQL BLAST DOWNLOAD HELP FAQ CONTACT
NC_005966
Percentage orphans
100
80
60
40
20
0
0
50
100
150
200
250
300
Genome Number
Done

DATASET 1: 68770 orphans from 330 genomes (972526 predicted proteins)

Acinetobacter sp. ADP1

orphan_plot2.php Implementation

When selecting to view an orphan plot, the user must choose what type of plot they wish to view by selecting the relevant radio button. A variable representing this choice is passed to *orphan_plot2.php*. *Orphan_plot2.php* is responsible for obtaining text to go alongside the plot and formatting the HTML. The plot itself is obtained by the php page *plot_link2.php*. This script is called from within the <img> tag in *orphan_plot2.php*. *plot_link2.php* is responsible for obtaining the plot file from the dbase server. This is done by querying the Paths_dataset3 MySQL table and utilising the PHP functions imagecreatefromjpeg(), imagejpeg() and imagedestroy().

true_paralogues3.php

- Displays the gene clusters within a given genome that include an orphan.
- Allows the user to view a sequence of interest or BLAST a sequence of interest against a database.

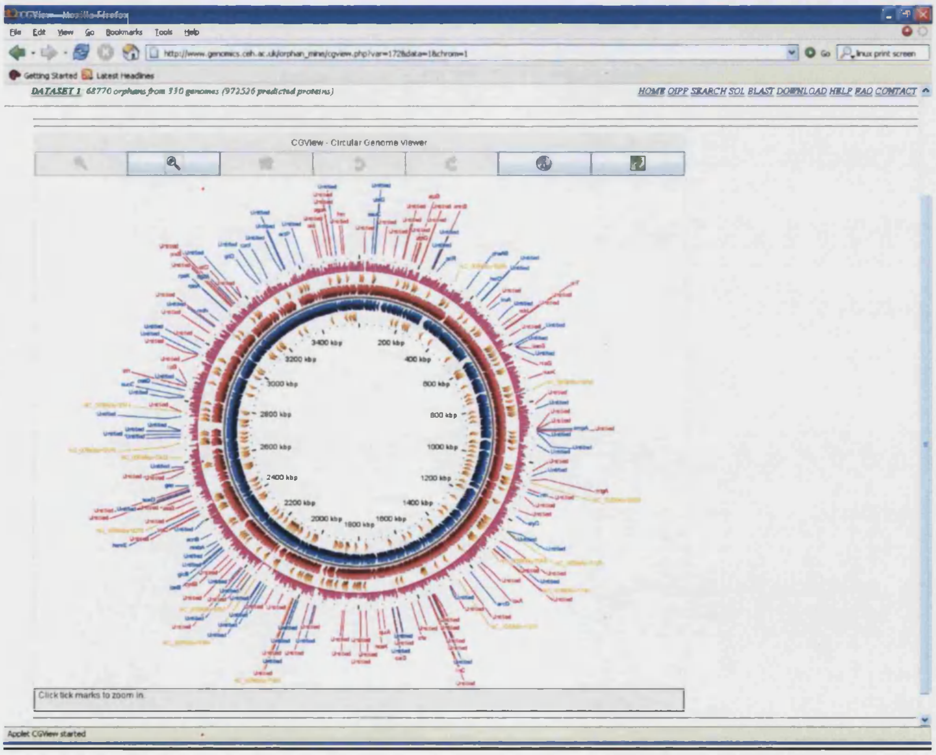
true_paralogues3.php Implementation

The page displays tables containing several genes. Each table represents a gene cluster within that genome, containing an orphan gene. The colour of the table cells is dependent on whether the gene is an orphan or not. An orphan gene will be coloured blue, a non-orphan gene will be coloured red. Therefore, if a cluster contains only genes that are unique to the genome of interest, the table will be completely blue. The tables are arranged by the size of the cluster, larger clusters will be positioned nearer the top of the page.

To generate the data necessary to produce the clusters, *true_paralogues3.php* obtains a list of orphans from the genome of interest that are found to match genes within their own genome. This data comes from the MySQL table Orphan3. The next stage is to use the identifiers of these orphans to query the MySQL table Para_blast, one at a time. Each query will return a list of the genes found to significantly match that orphan. Each of these genes is queried against Orphan3. If a match is found, that gene is also an orphan and the table cell will be coloured blue. If there is no match, the gene is queried against Orf3 to obtain its associated metadata. The table cells containing this data will be coloured red.

cgview.php & custom_cgview.php

- Displays the selected chromosome in the CGView applet (see section 4.8.2).
- Illustrates the chromosomal position of genes of interest.



CGView - Mozilla Firefox

http://www.genomics.ceh.ac.uk/orphan_mine/cgview.php?var=172&data=1&chrom=1

DATASET 1: 68770 orphans from 330 genomes (972526 predicted proteins)

CGView - Circular Genome Viewer

Click tick marks to zoom in.

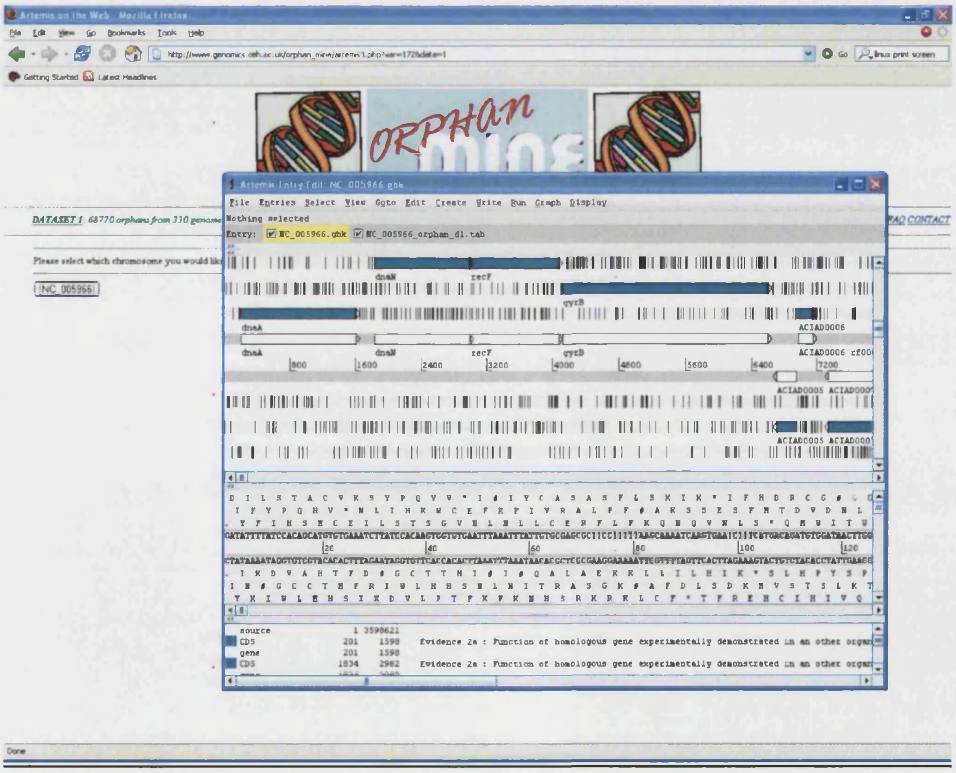
Applet CGView started

cgview.php & custom_cgview.php Implementation

The main role of these scripts is to launch the CGView applet. In order to pass the necessary data as arguments to the applet, the scripts *getcgv.cgi* and *get_custom_cgv.php* are used respectively. These scripts are initiated within the <applet> tag. The output of both *getcgv.cgi* and *get_custom.cgv.php* is in text format and in the case of *get_custom_cgv.php*, is generated on the fly. This data is read directly into the applet. When a user is investigating the pre-generated orphan gene datasets, *cgview.php* is used. Otherwise *custom_cgview.php* is used. If the genome of interest contains more than one chromosome, the file *cgview_prompt.php* is loaded. This file provides buttons for each chromosome, forcing the user to choose which chromosome they want to view. Once the choice is made, *cgview.php* or *custom_cgview.php* will load.

artemis3.php & custom_artemis2.php

- Initialises the Artemis application, allowing users to view their genes of interest in a genomic context. Additionally, users can add their own annotation files for further analysis.


Artemis on the Web - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/artemis3.php?var=172&data=1
Go
linux print screen
Getting Started
Latest Headlines
ORPHAN mine
DATASET 1 68770 orphans from 330 genomes
FAQ CONTACT
Please select which chromosome you would lik
NC_005966
Artemis Entry Edit: NC_005966.gbk
File Entries Select View Goto Edit Create Write Run Graph Display
dnaN
recF
gyrB
dnaA
ACIAD0006
rf00
ACIAD0005
Done

```
Nothing selected
Entry: ☑ NC_005966.gbk ☑ NC_005966_orphan_dl.tab
```

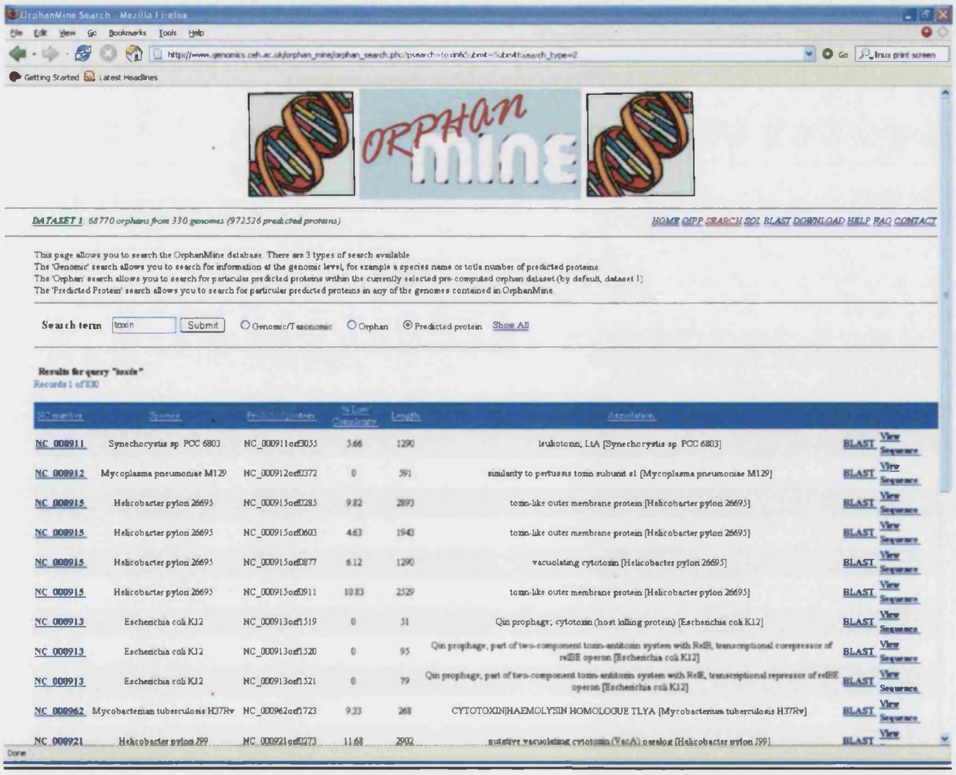
```
source        1 3598621
CDS         201    1598    Evidence 2a : Function of homologous gene experimentally demonstrated in an other organ
gene        201    1598
CDS        1834    2982    Evidence 2a : Function of homologous gene experimentally demonstrated in an other organ
```

artemis3.php & custom_artemis2.php Implementation

The role of these scripts is to initialise the Artemis application. This is done by loading *make_jnlp.php* and *make_custom_jnlp.php* respectively. These scripts generate a JNLP file containing the arguments required to load Artemis from the Sanger Centre server. In addition, they pass as arguments, the location of the relevant GenBank file, and the annotation file relevant to their dataset. The annotation files loaded by *make_jnlp.php* are pre-generated and stored on our server. The annotation files for custom datasets, loaded by *make_custom_jnlp.php,* are generated on the fly by *get_custom_tab2.php.*

orphan_search.php

- Allows users to perform a free text search of the database
- Provides three categories of search. The user may search data at the genomic level or at the level of the predicted protein. Alternatively they may choose to limit their search to the pre-generated orphans in the dataset they are currently viewing.
- On performing a search, several fields of data relevant to the type of search performed are displayed.
- If the user chooses to view more information, they can select the 'More Info' link for the genome of interest. Alternatively, if the results of the search are individual proteins, the user may view the sequence or BLAST the sequence against a database.
- Columns displayed in the results table can be sorted by ascending or descending order.


OrphanMine Search - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/orphan_search.php?psearch=toxin&Submit=Submit&search_type=2
Go
linux print screen
Getting Started
Latest Headlines
ORPHAN MINE
HOME OIPP SEARCH SQL BLAST DOWNLOAD HELP FAQ CONTACT
Done

DATASET 1 68770 orphans from 330 genomes (972526 predicted proteins)

This page allows you to search the OrphanMine database There are 3 types of search available
The 'Genomic' search allows you to search for information at the genomic level, for example a species name or totla number of predicted proteins
The 'Orphan' search allows you to search for particular predicted proteins within the currently selected pre-computed orphan dataset (by default, dataset 1)
The 'Predicted Protein' search allows you to search for particular predicted proteins in any of the genomes contained in OrphanMine

Search term toxin Submit ○ Genomic/Taxonomic ○ Orphan ● Predicted protein Show All

Results for query "*toxin*"
Records 1 of 830

NC number	Species	Predicted protein	% Low Complexity	Length	Annotation		
NC_000911	Synechocystis sp. PCC 6803	NC_000911orf3055	5.66	1290	leukotoxin, LtA [Synechocystis sp. PCC 6803]	BLAST	View Sequence
NC_000912	Mycoplasma pneumoniae M129	NC_000912orf0372	0	591	similarity to pertussis toxin subunit s1 [Mycoplasma pneumoniae M129]	BLAST	View Sequence
NC_000915	Helicobacter pylori 26695	NC_000915orf0285	9.82	2893	toxin-like outer membrane protein [Helicobacter pylori 26695]	BLAST	View Sequence
NC_000915	Helicobacter pylori 26695	NC_000915orf0603	4.63	1943	toxin-like outer membrane protein [Helicobacter pylori 26695]	BLAST	View Sequence
NC_000915	Helicobacter pylori 26695	NC_000915orf0877	6.12	1290	vacuolating cytotoxin [Helicobacter pylori 26695]	BLAST	View Sequence
NC_000915	Helicobacter pylori 26695	NC_000915orf0911	10.83	2529	toxin-like outer membrane protein [Helicobacter pylori 26695]	BLAST	View Sequence
NC_000913	Escherichia coli K12	NC_000913orf1519	0	51	Qin prophage; cytotoxin (host killing protein) [Escherichia coli K12]	BLAST	View Sequence
NC_000913	Escherichia coli K12	NC_000913orf1520	0	95	Qin prophage; part of two-component toxin-antitoxin system with RelB, transcriptional corepressor of relBE operon [Escherichia coli K12]	BLAST	View Sequence
NC_000913	Escherichia coli K12	NC_000913orf1521	0	79	Qin prophage; part of two-component toxin-antitoxin system with RelE, transcriptional repressor of relBE operon [Escherichia coli K12]	BLAST	View Sequence
NC_000962	Mycobacterium tuberculosis H37Rv	NC_000962orf1723	9.33	268	CYTOTOXIN|HAEMOLYSIN HOMOLOGUE TLYA [Mycobacterium tuberculosis H37Rv]	BLAST	View Sequence
NC_000921	Helicobacter pylori J99	NC_000921orf0273	11.68	2902	putative vacuolating cytotoxin (VacA) paralog [Helicobacter pylori J99]	BLAST	View

orphan_search.php Implementation

orphan_search.php is the main search page for the OrphanMine system. It is accessed from the ‘SEARCH’ option in the navigation bar. The search page allows users to search for data stored in OrphanMine using free text. The text entered will be used to search the majority of fields in the database. When search results are displayed, the query remains in the relevant text box, plus the term ‘Results for query “query”’ is printed above the results, in addition to the number of records found. The page displays 20 records at a time. A navigation bar is present at the bottom of the results table that allows the user to select the results they wish to view. The columns displayed in the results table can be sorted by ascending or descending order, this is achieved by clicking on the column headings. The section of the PHP script that is involved with the mechanisms of the search was generated by a tool called PHPMaker. The output from this programme was heavily modified to fit the requirements of OrphanMine.

When the user enters a search query the PHP script generates an SQL SELECT statement. The user query is utilised as the WHERE argument. An example of this is shown below, where ‘toxin’ was the query in a ‘Predicted protein’ search:

```
$user_query = ‘toxin’
$dbwhere = (`Orf` LIKE '%$user_query%' OR `Gi` LIKE
'%$user_query%' OR `Low_complexity` LIKE
'%$user_query%' OR `Length` LIKE '%$user_query%' OR
`Description` LIKE '%$user_query%' OR `Species` LIKE
'%$user_query%');

$strsql = "SELECT * FROM `genome3`, ‘orf3’ WHERE
genome3.Genome_id = orf3.Genome_id";

if ($dbwhere != "")
{
      $strsql .= " AND ".$dbwhere;
}
$rs = mysql_query($strsql, $conn)
      or die(mysql_error());
```

The same basic search mechanism is used for the 'Genomic/Taxonomic' and the 'Orphan' searches.

sequence download.php

- Allows the user to download protein sequences in FASTA format.
- The user can select the genomes and dataset from which the sequences should be obtained.
- Allows the user to download the protein sequences of more than one genome.

sequence download.php Implementation

This script lists the genomes held in OrphanMine. Next to each genome is a checkbox. By selecting these checkboxes, the user is selecting which genomes they wish to have sequences downloaded from. In addition, the user needs to select which dataset they wish to use, by selecting the relevant radio button. By clicking on the name of the dataset, a pop-up box appears describing that dataset. The exception to this is 'Custom'. By clicking on 'Custom', a box appears displaying what the current custom dataset is, this is performed by *customise_check.php.* The sequences are downloaded using the script *downloading.php*, this is initiated when the 'Download Orphans' button is pressed.

faq.php

- Provides the user with answers to common OrphanMine related queries.

faq.php Implementation

At the top of the page is a list of the questions answered in the FAQ section. These questions are linked to the section of the page in which they are answered. The questions that make up the FAQ's can be updated depending on user response.

contacts.php

- Provides contact details of people involved in the maintenance of OrphanMine.
- Initiates default e-mail editor to construct an e-mail to selected contact.

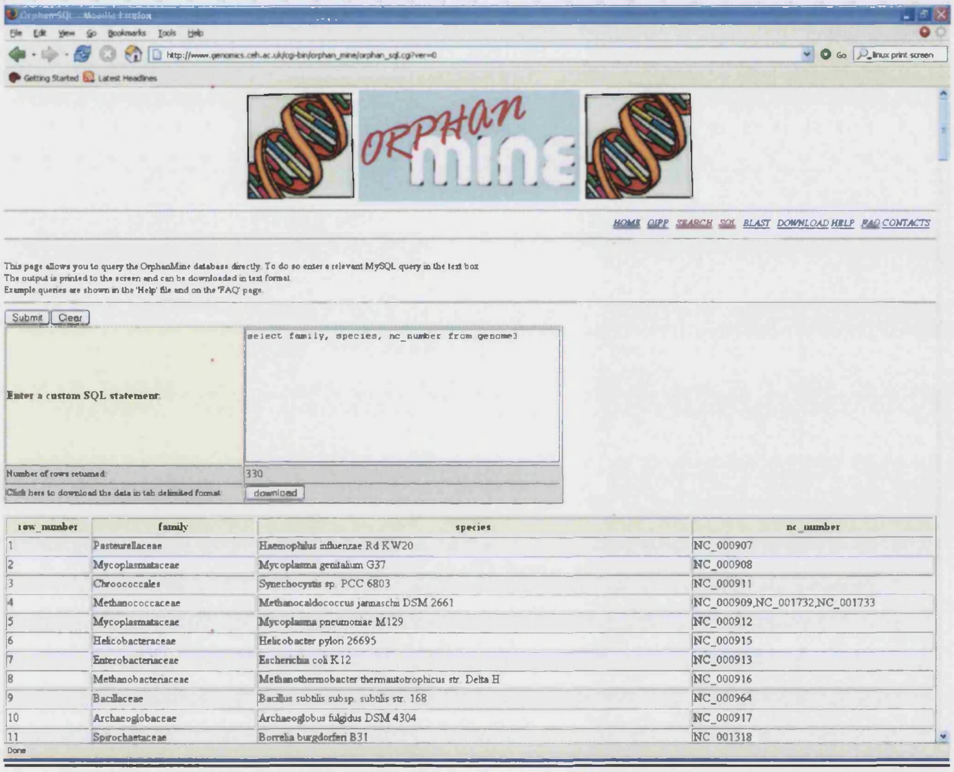
contacts.php Implementation

The contacts page provides users with contact details of those involved in the maintenance of the site. The page was initially generated by PHPMaker and was later modified for use in OrphanMine.

Perl CGI Page Descriptions

orphan_sql.cgi

- Provides a text box for users to enter their own SQL queries, therefore allowing users to query the database directly.
- Prints the results of the query to screen in an HTML table.
- Allows users to download the output in tab delimited format to a text file.


Orphan-SQL - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/cgi-bin/orphan_mine/orphan_sql.cgi?ver=0
Go
linux print screen
Getting Started
Latest Headlines
ORPHAN MINE
HOME OIPP SEARCH SQL BLAST DOWNLOAD HELP FAQ CONTACTS
Submit
Clear
download
Done

This page allows you to query the OrphanMine database directly. To do so enter a relevant MySQL query in the text box
The output is printed to the screen and can be downloaded in text format
Example queries are shown in the 'Help' file and on the 'FAQ' page

Enter a custom SQL statement:

```
select family, species, nc_number from genome3
```

Number of rows returned: 330

Click here to download the data in tab delimited format

row_number	family	species	nc_number
1	Pasteurellaceae	Haemophilus influenzae Rd KW20	NC_000907
2	Mycoplasmataceae	Mycoplasma genitalium G37	NC_000908
3	Chroococcales	Synechocystis sp. PCC 6803	NC_000911
4	Methanococcaceae	Methanocaldococcus jannaschii DSM 2661	NC_000909,NC_001732,NC_001733
5	Mycoplasmataceae	Mycoplasma pneumoniae M129	NC_000912
6	Helicobacteraceae	Helicobacter pylori 26695	NC_000915
7	Enterobacteriaceae	Escherichia coli K12	NC_000913
8	Methanobacteriaceae	Methanothermobacter thermautotrophicus str. Delta H	NC_000916
9	Bacillaceae	Bacillus subtilis subsp. subtilis str. 168	NC_000964
10	Archaeoglobaceae	Archaeoglobus fulgidus DSM 4304	NC_000917
11	Spirochaetaceae	Borrelia burgdorferi B31	NC_001318

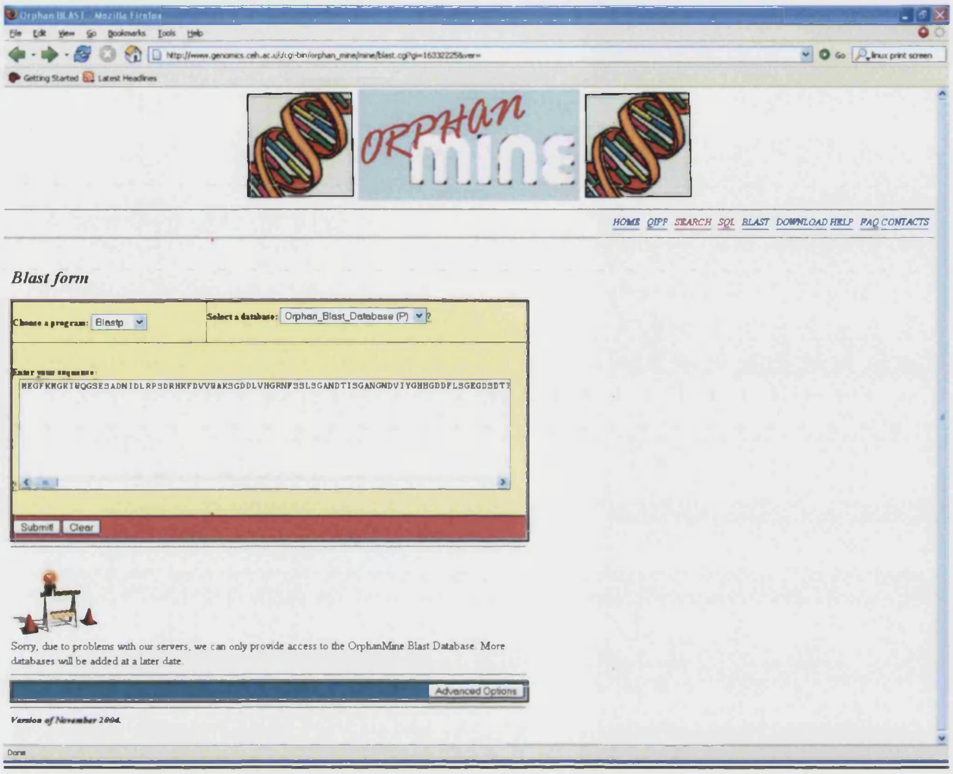
orphan_sql.cgi Implementation

The script was originally written by Dr Milo Thurston and was later modified for use in OrphanMine. Only SELECT queries are permitted. If a user attempted to submit a query that would alter the contents of the database, for example 'DROP table Orf3;', an error message would be returned. The database schema is available in the FAQs and in the Help page, to assist users to form meaningful queries. The FAQ page also provides some example queries.

blast.cgi

- Provides a text box for users to enter their sequence. Alternatively, if the user has arrived at the page by selecting the BLAST link associated with a particular protein, the sequence of that protein will be pre-loaded in the box.

- Provides a choice of BLAST programmes and BLAST databases.
- Allows the user to configure their BLAST search by providing them with advanced options.


Orphan BLAST - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/cgi-bin/orphan_mine/mine/blast.cgi?gi=16332225&ver=
Go
linux print screen
Getting Started
Latest Headlines
ORPHAN MINE
HOME QIPP SEARCH SQL BLAST DOWNLOAD HELP FAQ CONTACTS
Submit
Clear
Advanced Options
Done

Blast form

Choose a program: Blastp

Select a database: Orphan_Blast_Database (P) ?

Enter your sequence:

```
MEGFKMGKIWQGSESADNIDLRPSDRHKFDVVWAKSGDDLVHGRNFSSLSGANDTISGANGNDVIYGHHGDDFLSGEGDSDT
```

Sorry, due to problems with our servers, we can only provide access to the OrphanMine Blast Database. More databases will be added at a later date.

Version of November 2004.

blast.cgi Implementation

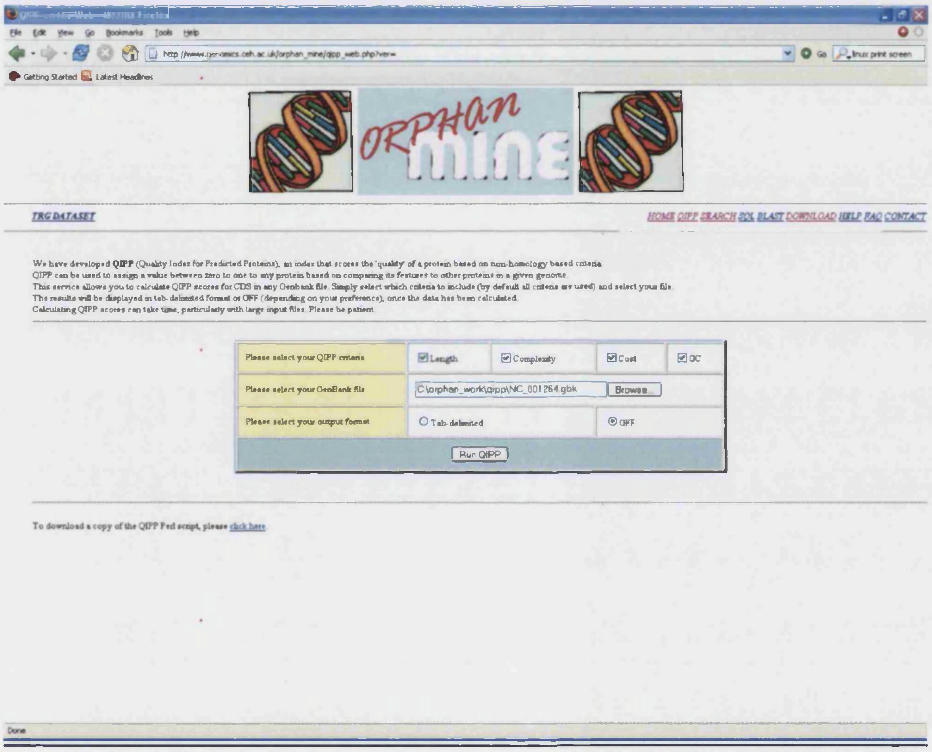
This script was originally written by John Peden (OUBC) and was later modified for use in OrphanMine. When the script is loaded, a check is performed to determine if a GI identifier of a protein has been passed as a parameter. This would occur if the user was following a link from a particular protein, rather than using the link in the page header. If a GI identifier is found, the programme *fastacmd* is used to obtain the sequence for the relevant protein. This sequence is then printed to screen in the text box, ready to be BLASTed.

Drop down menus provide the user with a choice of BLAST programmes and BLAST databases. If the user wants to view the advanced options available to them, they should click on the 'Advanced Options' button. This will load an additional table containing further options. When the user selects the 'Submit' button, the form element values are passed as parameters to the blastall command. Blastall performs the BLAST and returns the results to the user. If there is a problem with the input, an error message will be displayed explaining the problem.

WebQIPP

qipp_web.php

- The page provides an easy-to-use interface to access the *qipp.pl* script.
- Allows the user to select a GenBank file from their directory and select the criteria by which they want to score the sequences in the file.
- Provides the option to produce the output as a tab-delimited text file, ranked by QIPP score or as a GFF file suitable for use in Artemis.
- Provides a link to download the *qipp.pl* file for use locally.



QIPP on the Web - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/qipp_web.php?ver=
Go
linux print screen
Getting Started Latest Headlines
ORPHAN MINE
TRG DATASET
HOME QIPP SEARCH SQL BLAST DOWNLOAD HELP FAQ CONTACT
Done

We have developed **QIPP** (Quality Index for Predicted Proteins), an index that scores the 'quality' of a protein based on non-homology based criteria.
QIPP can be used to assign a value between zero to one to any protein based on comparing its features to other proteins in a given genome.
This service allows you to calculate QIPP scores for CDS in any Genbank file. Simply select which criteria to include (by default all criteria are used) and select your file.
The results will be displayed in tab-delimited format or GFF (depending on your preference), once the data has been calculated.
Calculating QIPP scores can take time, particularly with large input files. Please be patient.

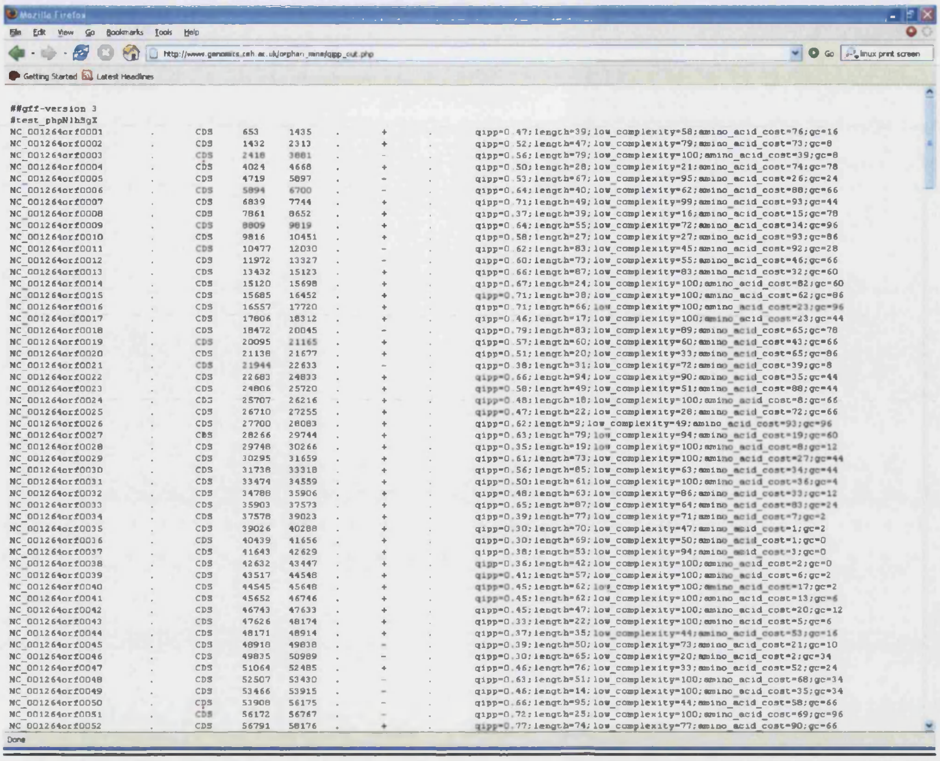
Please select your QIPP criteria	☑ Length	☑ Complexity	☑ Cost	☑ GC
Please select your GenBank file	C:\orphan_work\qipp\NC_001264.gbk Browse...			
Please select your output format	○ Tab-delimited		● GFF	
Run QIPP				

To download a copy of the QIPP Perl script, please click here.

qipp_web.php Implementation

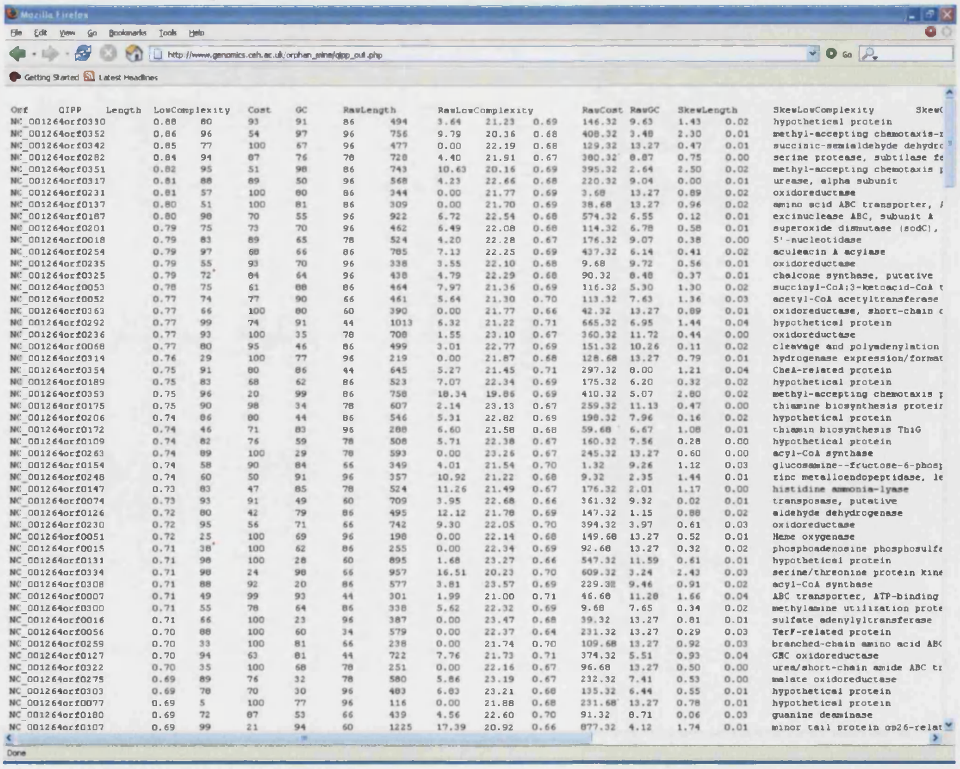
The form elements capture the required information from the user. When the 'Run QIPP' button is pressed, *qipp_out.php* is launched. This script obtains the parameters passed by *qipp_web.php*. Numerous checks are carried out to determine the authenticity of the uploaded file. Once the checks are complete, *qipp.pl* is launched. The output from this script is printed directly to screen. Finally the uploaded GenBank file is deleted from the server.

WebQIPP Output - GFF


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/qipp_out.php
Go
linux print screen
Getting Started
Latest Headlines
Done

```
##gff-version 3
#test_phpNlh5gX
NC_001264orf0001    .    CDS    653      1435     .    +    .    qipp=0.47;length=39;low_complexity=58;amino_acid_cost=76;gc=16
NC_001264orf0002    .    CDS    1432     2313     .    +    .    qipp=0.52;length=47;low_complexity=79;amino_acid_cost=73;gc=8
NC_001264orf0003    .    CDS    2418     3881     .    -    .    qipp=0.56;length=79;low_complexity=100;amino_acid_cost=39;gc=8
NC_001264orf0004    .    CDS    4024     4668     .    +    .    qipp=0.50;length=28;low_complexity=21;amino_acid_cost=74;gc=78
NC_001264orf0005    .    CDS    4719     5897     .    -    .    qipp=0.53;length=67;low_complexity=95;amino_acid_cost=26;gc=24
NC_001264orf0006    .    CDS    5894     6700     .    -    .    qipp=0.64;length=40;low_complexity=62;amino_acid_cost=88;gc=66
NC_001264orf0007    .    CDS    6839     7744     .    +    .    qipp=0.71;length=49;low_complexity=99;amino_acid_cost=93;gc=44
NC_001264orf0008    .    CDS    7861     8652     .    +    .    qipp=0.37;length=39;low_complexity=16;amino_acid_cost=15;gc=78
NC_001264orf0009    .    CDS    8809     9819     .    +    .    qipp=0.64;length=55;low_complexity=72;amino_acid_cost=34;gc=96
NC_001264orf0010    .    CDS    9816     10451    .    +    .    qipp=0.58;length=27;low_complexity=27;amino_acid_cost=93;gc=86
NC_001264orf0011    .    CDS    10477    12030    .    -    .    qipp=0.62;length=83;low_complexity=45;amino_acid_cost=92;gc=28
NC_001264orf0012    .    CDS    11972    13327    .    -    .    qipp=0.60;length=73;low_complexity=55;amino_acid_cost=46;gc=66
NC_001264orf0013    .    CDS    13432    15123    .    +    .    qipp=0.66;length=87;low_complexity=83;amino_acid_cost=32;gc=60
NC_001264orf0014    .    CDS    15120    15698    .    +    .    qipp=0.67;length=24;low_complexity=100;amino_acid_cost=82;gc=60
NC_001264orf0015    .    CDS    15685    16452    .    +    .    qipp=0.71;length=38;low_complexity=100;amino_acid_cost=62;gc=86
NC_001264orf0016    .    CDS    16557    17720    .    +    .    qipp=0.71;length=66;low_complexity=100;amino_acid_cost=23;gc=96
NC_001264orf0017    .    CDS    17806    18312    .    +    .    qipp=0.46;length=17;low_complexity=100;amino_acid_cost=23;gc=44
NC_001264orf0018    .    CDS    18472    20045    .    -    .    qipp=0.79;length=83;low_complexity=89;amino_acid_cost=65;gc=78
NC_001264orf0019    .    CDS    20095    21165    .    +    .    qipp=0.57;length=60;low_complexity=60;amino_acid_cost=43;gc=66
NC_001264orf0020    .    CDS    21138    21677    .    +    .    qipp=0.51;length=20;low_complexity=33;amino_acid_cost=65;gc=86
NC_001264orf0021    .    CDS    21944    22633    .    -    .    qipp=0.38;length=31;low_complexity=72;amino_acid_cost=39;gc=8
NC_001264orf0022    .    CDS    22683    24833    .    +    .    qipp=0.66;length=94;low_complexity=90;amino_acid_cost=35;gc=44
NC_001264orf0023    .    CDS    24806    25720    .    +    .    qipp=0.58;length=49;low_complexity=51;amino_acid_cost=88;gc=44
NC_001264orf0024    .    CDS    25707    26216    .    +    .    qipp=0.48;length=18;low_complexity=100;amino_acid_cost=8;gc=66
NC_001264orf0025    .    CDS    26710    27255    .    +    .    qipp=0.47;length=22;low_complexity=28;amino_acid_cost=72;gc=66
NC_001264orf0026    .    CDS    27700    28083    .    +    .    qipp=0.62;length=9;low_complexity=49;amino_acid_cost=93;gc=96
NC_001264orf0027    .    CDS    28266    29744    .    +    .    qipp=0.63;length=79;low_complexity=94;amino_acid_cost=19;gc=60
NC_001264orf0028    .    CDS    29748    30266    .    +    .    qipp=0.35;length=19;low_complexity=100;amino_acid_cost=8;gc=12
NC_001264orf0029    .    CDS    30295    31659    .    +    .    qipp=0.61;length=73;low_complexity=100;amino_acid_cost=27;gc=44
NC_001264orf0030    .    CDS    31738    33318    .    +    .    qipp=0.56;length=85;low_complexity=63;amino_acid_cost=34;gc=44
NC_001264orf0031    .    CDS    33474    34559    .    +    .    qipp=0.50;length=61;low_complexity=100;amino_acid_cost=36;gc=4
NC_001264orf0032    .    CDS    34788    35906    .    +    .    qipp=0.48;length=63;low_complexity=86;amino_acid_cost=33;gc=12
NC_001264orf0033    .    CDS    35903    37573    .    +    .    qipp=0.65;length=87;low_complexity=64;amino_acid_cost=83;gc=24
NC_001264orf0034    .    CDS    37578    39023    .    +    .    qipp=0.39;length=77;low_complexity=71;amino_acid_cost=7;gc=2
NC_001264orf0035    .    CDS    39026    40288    .    +    .    qipp=0.30;length=70;low_complexity=47;amino_acid_cost=1;gc=2
NC_001264orf0036    .    CDS    40439    41656    .    +    .    qipp=0.30;length=69;low_complexity=50;amino_acid_cost=1;gc=0
NC_001264orf0037    .    CDS    41643    42629    .    +    .    qipp=0.38;length=53;low_complexity=94;amino_acid_cost=3;gc=0
NC_001264orf0038    .    CDS    42632    43447    .    +    .    qipp=0.36;length=42;low_complexity=100;amino_acid_cost=2;gc=0
NC_001264orf0039    .    CDS    43517    44548    .    +    .    qipp=0.41;length=57;low_complexity=100;amino_acid_cost=6;gc=2
NC_001264orf0040    .    CDS    44545    45648    .    +    .    qipp=0.45;length=62;low_complexity=100;amino_acid_cost=17;gc=2
NC_001264orf0041    .    CDS    45652    46746    .    +    .    qipp=0.45;length=62;low_complexity=100;amino_acid_cost=13;gc=6
NC_001264orf0042    .    CDS    46743    47633    .    +    .    qipp=0.45;length=47;low_complexity=100;amino_acid_cost=20;gc=12
NC_001264orf0043    .    CDS    47626    48174    .    +    .    qipp=0.33;length=22;low_complexity=100;amino_acid_cost=5;gc=6
NC_001264orf0044    .    CDS    48171    48914    .    +    .    qipp=0.37;length=35;low_complexity=44;amino_acid_cost=53;gc=16
NC_001264orf0045    .    CDS    48918    49838    .    -    .    qipp=0.39;length=50;low_complexity=73;amino_acid_cost=21;gc=10
NC_001264orf0046    .    CDS    49835    50989    .    -    .    qipp=0.30;length=65;low_complexity=20;amino_acid_cost=2;gc=34
NC_001264orf0047    .    CDS    51064    52485    .    +    .    qipp=0.46;length=76;low_complexity=33;amino_acid_cost=52;gc=24
NC_001264orf0048    .    CDS    52507    53430    .    -    .    qipp=0.63;length=51;low_complexity=100;amino_acid_cost=68;gc=34
NC_001264orf0049    .    CDS    53466    53915    .    -    .    qipp=0.46;length=14;low_complexity=100;amino_acid_cost=35;gc=34
NC_001264orf0050    .    CDS    53908    56175    .    -    .    qipp=0.66;length=95;low_complexity=44;amino_acid_cost=58;gc=66
NC_001264orf0051    .    CDS    56172    56767    .    -    .    qipp=0.72;length=25;low_complexity=100;amino_acid_cost=69;gc=96
NC_001264orf0052    .    CDS    56791    58176    .    +    .    qipp=0.77;length=74;low_complexity=77;amino_acid_cost=90;gc=66
```

WebQIPP Output – Tab-delimited


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://www.genomics.ceh.ac.uk/orphan_mine/qipp_out.php
Go
Getting Started
Latest Headlines
Done

```
Orf          QIPP     Length  LowComplexity     Cost   GC     RawLength     RawLowComplexity         RawCost  RawGC   SkewLength      SkewLowComplexity        Skew(
NC_001264orf0330      0.88   80    93     91     86    494    3.64    21.23   0.69    146.32  9.63    1.43    0.02    hypothetical protein
NC_001264orf0352      0.86   96    54     97     96    756    9.79    20.36   0.68    408.32  3.48    2.30    0.01    methyl-accepting chemotaxis-r
NC_001264orf0342      0.85   77    100    67     96    477    0.00    22.19   0.68    129.32  13.27   0.47    0.01    succinic-semialdehyde dehydrc
NC_001264orf0282      0.84   94    87     76     78    728    4.40    21.91   0.67    380.32  8.87    0.75    0.00    serine protease, subtilase fe
NC_001264orf0351      0.82   95    51     98     86    743    10.63   20.16   0.69    395.32  2.64    2.50    0.02    methyl-accepting chemotaxis p
NC_001264orf0317      0.81   88    89     50     96    568    4.23    22.66   0.68    220.32  9.04    0.00    0.01    urease, alpha subunit
NC_001264orf0231      0.81   57    100    80     86    344    0.00    21.77   0.69    3.68    13.27   0.89    0.02    oxidoreductase
NC_001264orf0137      0.80   51    100    81     86    309    0.00    21.70   0.69    38.68   13.27   0.96    0.02    amino acid ABC transporter, A
NC_001264orf0187      0.80   98    70     55     96    922    6.72    22.54   0.68    574.32  6.55    0.12    0.01    excinuclease ABC, subunit A
NC_001264orf0201      0.79   75    73     70     96    462    6.49    22.08   0.68    114.32  6.78    0.58    0.01    superoxide dismutase (sodC),
NC_001264orf0018      0.79   83    89     65     78    524    4.20    22.28   0.67    176.32  9.07    0.38    0.00    5'-nucleotidase
NC_001264orf0254      0.79   97    68     66     86    785    7.13    22.25   0.69    437.32  6.14    0.41    0.02    aculeacin A acylase
NC_001264orf0235      0.79   55    93     70     96    338    3.55    22.10   0.68    9.68    9.72    0.56    0.01    oxidoreductase
NC_001264orf0325      0.79   72    84     64     96    438    4.79    22.29   0.68    90.32   8.48    0.37    0.01    chalcone synthase, putative
NC_001264orf0053      0.78   75    61     88     86    464    7.97    21.36   0.69    116.32  5.30    1.30    0.02    succinyl-CoA:3-ketoacid-CoA t
NC_001264orf0052      0.77   74    77     90     66    461    5.64    21.30   0.70    113.32  7.63    1.36    0.03    acetyl-CoA acetyltransferase
NC_001264orf0363      0.77   66    100    80     60    390    0.00    21.77   0.66    42.32   13.27   0.89    0.01    oxidoreductase, short-chain c
NC_001264orf0292      0.77   99    74     91     44    1013   6.32    21.22   0.71    665.32  6.95    1.44    0.04    hypothetical protein
NC_001264orf0236      0.77   93    100    35     78    708    1.55    23.10   0.67    360.32  11.72   0.44    0.00    oxidoreductase
NC_001264orf0068      0.77   80    95     46     86    499    3.01    22.77   0.69    151.32  10.26   0.11    0.02    cleavage and polyadenylation
NC_001264orf0314      0.76   29    100    77     96    219    0.00    21.87   0.68    128.68  13.27   0.79    0.01    hydrogenase expression/format
NC_001264orf0354      0.75   91    80     86     44    645    5.27    21.45   0.71    297.32  8.00    1.21    0.04    CheA-related protein
NC_001264orf0189      0.75   83    68     62     86    523    7.07    22.34   0.69    175.32  6.20    0.32    0.02    hypothetical protein
NC_001264orf0353      0.75   96    20     99     86    758    18.34   19.86   0.69    410.32  5.07    2.80    0.02    methyl-accepting chemotaxis p
NC_001264orf0175      0.75   90    98     34     78    607    2.14    23.13   0.67    259.32  11.13   0.47    0.00    thiamine biosynthesis proteir
NC_001264orf0206      0.74   86    80     44     86    546    5.31    22.82   0.69    198.32  7.96    0.16    0.02    hypothetical protein
NC_001264orf0172      0.74   46    71     83     96    288    6.60    21.58   0.68    59.68   6.67    1.08    0.01    thiamin biosynthesis ThiG
NC_001264orf0109      0.74   82    76     59     78    508    5.71    22.38   0.67    160.32  7.56    0.28    0.00    hypothetical protein
NC_001264orf0263      0.74   89    100    29     78    593    0.00    23.26   0.67    245.32  13.27   0.60    0.00    acyl-CoA synthase
NC_001264orf0154      0.74   58    90     84     66    349    4.01    21.54   0.70    1.32    9.26    1.12    0.03    glucosamine--fructose-6-phosp
NC_001264orf0248      0.74   60    50     91     96    357    10.92   21.22   0.68    9.32    2.35    1.44    0.01    zinc metalloendopeptidase, le
NC_001264orf0147      0.73   83    47     85     78    524    11.26   21.49   0.67    176.32  2.01    1.17    0.00    histidine ammonia-lyase
NC_001264orf0074      0.73   93    91     49     60    709    3.95    22.68   0.66    361.32  9.32    0.02    0.01    transposase, putative
NC_001264orf0126      0.72   80    42     79     86    495    12.12   21.78   0.69    147.32  1.15    0.88    0.02    aldehyde dehydrogenase
NC_001264orf0230      0.72   95    56     71     66    742    9.30    22.05   0.70    394.32  3.97    0.61    0.03    oxidoreductase
NC_001264orf0051      0.72   25    100    69     96    198    0.00    22.14   0.68    149.68  13.27   0.52    0.01    Heme oxygenase
NC_001264orf0015      0.71   38    100    62     86    255    0.00    22.34   0.69    92.68   13.27   0.32    0.02    phosphoadenosine phosphosulfe
NC_001264orf0131      0.71   98    100    28     60    895    1.68    23.27   0.66    547.32  11.59   0.61    0.01    hypothetical protein
NC_001264orf0334      0.71   98    24     98     66    957    16.51   20.23   0.70    609.32  3.24    2.43    0.03    serine/threonine protein kine
NC_001264orf0308      0.71   88    92     20     86    577    3.81    23.57   0.69    229.32  9.46    0.91    0.02    acyl-CoA synthase
NC_001264orf0007      0.71   49    99     93     44    301    1.99    21.00   0.71    46.68   11.28   1.66    0.04    ABC transporter, ATP-binding
NC_001264orf0300      0.71   55    78     64     86    338    5.62    22.32   0.69    9.68    7.65    0.34    0.02    methylamine utilization prote
NC_001264orf0016      0.71   66    100    23     96    387    0.00    23.47   0.68    39.32   13.27   0.81    0.01    sulfate adenylyltransferase
NC_001264orf0056      0.70   88    100    60     34    579    0.00    22.37   0.64    231.32  13.27   0.29    0.03    TerF-related protein
NC_001264orf0259      0.70   33    100    81     66    238    0.00    21.74   0.70    109.68  13.27   0.92    0.03    branched-chain amino acid ABC
NC_001264orf0127      0.70   94    63     81     44    722    7.76    21.73   0.71    374.32  5.51    0.93    0.04    GMC oxidoreductase
NC_001264orf0322      0.70   35    100    68     78    251    0.00    22.16   0.67    96.68   13.27   0.50    0.00    urea/short-chain amide ABC tr
NC_001264orf0275      0.69   89    76     32     78    580    5.86    23.19   0.67    232.32  7.41    0.53    0.00    malate oxidoreductase
NC_001264orf0303      0.69   78    70     30     96    483    6.83    23.21   0.68    135.32  6.44    0.55    0.01    hypothetical protein
NC_001264orf0077      0.69   5     100    77     96    116    0.00    21.88   0.68    231.68  13.27   0.78    0.01    hypothetical protein
NC_001264orf0180      0.69   72    87     53     66    439    4.56    22.60   0.70    91.32   8.71    0.06    0.03    guanine deaminase
NC_001264orf0107      0.69   99    21     94     60    1225   17.39   20.92   0.66    877.32  4.12    1.74    0.01    minor tail protein gp26-relat
```

Appendix 4.3 – Design Evaluation

Below are ten heuristics used as a guide to evaluate the usability of OrphanMine. These heuristics were obtained from the book 'Human-Computer Interaction' by Dix *et al* (1993).

1. **Visibility of system status –** does the system always keep users informed about what is going on, through appropriate feedback, within reasonable time?

2. **Match between system and the real world** – does the system speak the user's language, with words, phrases and concepts familiar to the user, rather than system-oriented terms? Does the system follow real conventions, making information appear in a natural and logical order?

3. **User control and freedom** – users often choose system functions by mistake and will need a clearly marked 'emergency exit' to leave the unwanted state, without having to go through an extended dialogue. Does the system support undo and redo?

4. **Consistency and standards** - users should not have to wonder whether different words, situations or actions mean the same thing. Does the system follow platform conventions?

5. **Error prevention** – better than having good error messages, is a careful design which prevents a problem from occurring in the first place. Has the system created any problems?

6. **Recognition rather than recall** – does the system make objects, actions and options visible? Do you have to remember information from one part of the dialogue to another? Are instructions for use of the system visible or easily retrievable, whenever appropriate?

7. **Flexibility and efficiency of use** – Do you think the system would be easy to use by both inexperienced and experienced users? Does the system allow users to tailor frequent actions?

8. **Aesthetic and minimalist design** – Does the system dialogue contain information which is irrelevant or rarely needed?

9. **Help users recognise, diagnose and recover from errors** – are error messages expressed in plain language? Do they indicate the problem, precisely, and constructively suggest a solution?

10. **Help and documentation** – it may be necessary to provide help and documentation in such systems. Is such information easy to search? Is it focused on the user's task? Does it list concrete steps to be carried out?

Appendix 4.4 – Implementation Evaluation

The aim of this evaluation process is to test the functionality of the OrphanMine database. By obtaining your feedback, I hope to identify any problems and design a better system.

This type of co-operative evaluation involves active participation by the users of the system, i.e., you. I would like to observe you interacting with the system by completing a set of tasks, which should last about 20 minutes. In this time, I will watch and record your actions. I would also like you to elaborate your actions by 'thinking aloud' to tell me what you think is happening, and what you are trying to do with each action. This will help to provide useful insight into problems with the interface and allow me to observe how the system is actually used.

I may ask you questions throughout the process and I would like you to raise any problems or suggestions you may have. Please feel free to criticise (or praise!) the system.

Tasks

Orphans

1. Please go to the OrphanMine home page.
2. You will be presented with several choices. To begin with, I would like you to explore the pre-generated orphan datasets. You are particularly interested in finding out more about the first bacterial genome to have its genome completely sequenced. What species does this genome represent?
3. Please find more information about this genome, for example, how many orphans does it contain?
4. To get a better idea of how the number of orphans in this genome has changed as more genomes are sequenced, take a look at the plot.
5. Rank the orphans according to length and average amino acid cost.
6. Download the ranked gene list in GFF format.

Search

1. Determine which genome has the most orphan genes.

2. Search for the word 'virulence' in all the predicted proteins in OrphanMine. How many proteins match this keyword?
3. Of these proteins, BLAST the longest against the Orphan_Blast_Database using default settings. What, apart from self, is the top hit?

Custom

1. Return to the home page.
2. Could you proceed to create your own 'Custom Dataset'. You want to view genes that have significant matches to genes in all other genomes in the database. Look for more information about the *Escherichia coli* K12 genome.
3. View these genes in Artemis.
4. If you have a problem loading Artemis, how would you go about reporting the fault?

TRGs

1. Return to the home page.
2. You are interested in finding out which genes in *Bacillus anthracis* str. Sterne are only found in other *Bacillus anthracis* genomes. Please create this dataset.
3. Once the results have been generated, add the genes to your trolley and view your trolley.
4. Download the taxonomically restricted genes in your trolley and then view more information.
5. Take a look at the genome using CGView. Zoom in on one of the genes in your dataset and find its ID (NC_??????orf????).
6. What does the pink band represent?

Appendix 5.1 – Chapter 5 Table S1

The number of predicted proteins, orphans, percentage orphans, isolation index and taxonomic uniqueness for each of the 122 bacterial genomes used in this analysis

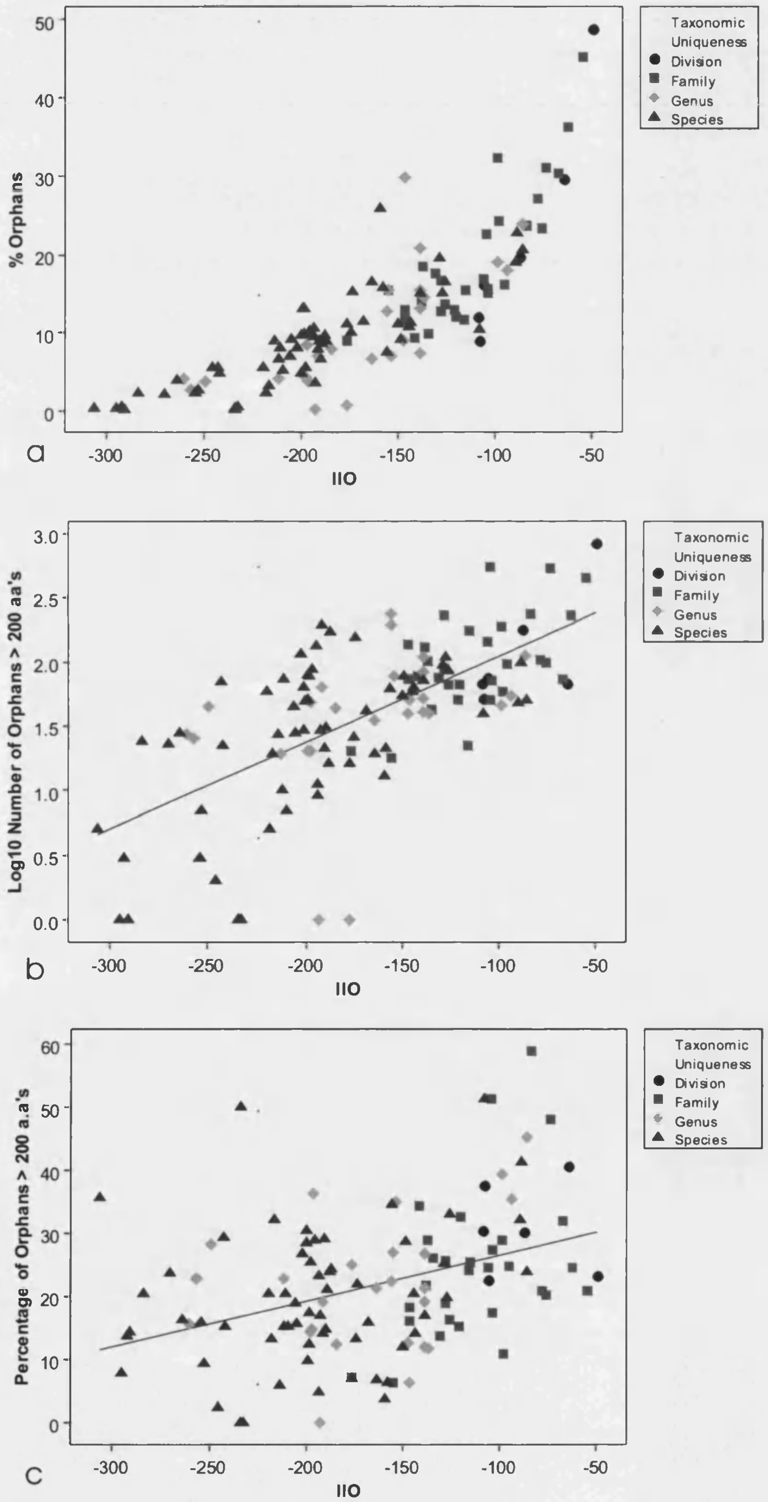
Species	Predicted Proteins	Orphans	% Orphans	IIO	Taxonomic Uniqueness
Haemophilus influenzae Rd	1657	38	2.29	-218.02	Species
Mycoplasma genitalium	484	2	0.41	-233	Species
Synechocystis sp PCC6803	3167	223	7.04	-153.99	Genus
Methanococcus jannaschii	1729	259	14.98	-103.83	Family
Mycoplasma pneumoniae	689	67	9.87	-188.09	Species
Helicobacter pylori 26695	1576	261	16.56	-125.97	Species
Escherichia coli K12	4311	174	4.04	-259.98	Genus
Methanobacterium thermoautotrophicum Delta H	1873	293	15.64	-104.14	Family
Bacillus subtilis,subsp. subtilis str. 168	4112	460	11.19	-150.05	Species
Archaeoglobus fulgidus	2420	391	16.16	-95.47	Family
Borrelia burgdorferi	851	152	17.86	-93.49	Genus
Aquifex aeolicus	1529	138	9.03	-108.15	Division
Pyrococcus horikoshii	1956	180	9.2	-205.47	Species
Mycobacterium tuberculosis H27Rv	3927	13	0.33	-294.93	Species
Treponema pallidum	1036	248	23.94	-86.01	Genus
Chlamydia trachomatis D/UW-3/CX	895	46	5.14	-209.85	Species
Rickettsia prowazekii	835	19	2.28	-253.82	Species
Chlamydophila pneumoniae CWL029	1054	102	9.68	-200.75	Species
Aeropyrum pernix	1841	499	27.1	-78.21	Family
Thermotoga maritima	1858	224	12.06	-108.7	Division
Deinococcus radiodurans	2997	591	19.72	-87.47	Division
Campylobacter jejuni	1634	161	9.85	-134.52	Family
Neisseria meningitidis MC58	2079	434	20.88	-138.93	Genus
Bacillus halodurans	4066	456	11.22	-143.81	Species
Xylella fastidiosa	2766	824	29.79	-146.52	Genus
Vibrio cholerae	3835	504	13.14	-200.1	Species
Buchnera sp. APS	564	2	0.35	-234	Species
Thermoplasma acidophilus	1482	55	3.71	-197.13	Genus
Pseudomonas aeruginosa PAO1	5567	306	5.5	-198.07	Species
Ureaplasma urealyticum	614	117	19.06	-98.92	Genus
Halobacterium sp. NRC-1	2075	485	23.37	-75.64	Family
Mesorhizobium loti	6746	864	12.81	-146.7	Family
Thermoplasma volcanium	1499	53	3.54	-193.65	Species
Mycobacterium leprae	1605	89	5.55	-245.57	Species
Pasteurella multocida Pm70	2015	83	4.12	-212.23	Genus
Streptococcus pyogenes M1GAS	1697	147	8.66	-189.56	Species
Staphylococcus aureus subsp. aureus N315	2593	205	7.91	-192.07	Species
Lactococcus lactis subsp. lactis	2321	336	14.48	-136.45	Genus
Mycoplasma pulmonis	782	149	19.05	-89.79	Species
Caulobacter crescentus	3737	475	12.71	-128.28	Family

Sulfolobus solfataricus	2977	271	9.1	-148.85	Species
Streptococcus pneumoniae TIGR4	2094	330	15.76	-158.42	Species
Sinorhizobium meliloti	3341	140	4.19	-198.6	Genus
Clostridium acetobutylicum	3672	556	15.14	-127.15	Species
Sulfolobus tokodaii	2826	426	15.07	-138.95	Species
Rickettsia conorii Malish 7	1374	354	25.76	-159.38	Species
Yersinia pestis C092	3885	325	8.37	-197.25	Genus
Salmonella enterica subsp. enterica serovar Typhi	4395	173	3.94	-263.64	Species
Salmonella typhimurium LT2	4451	97	2.18	-269.58	Species
Listeria innocua	2968	145	4.89	-242.27	Species
Listeria monocytogenes	2846	76	2.67	-252.97	Species
Nostoc sp PCC7120 (Cyanobacteria)	5366	892	16.62	-128.15	Family
Agrobacterium tumefaciens (C58 Cereon)	4554	356	7.82	-184.73	Genus
Brucella melitensis 16M	3198	284	8.88	-176.83	Family
Clostridium perfringens 13	2660	285	10.71	-144.81	Species
Pyrobaculum aerophilum	2605	945	36.28	-62.39	Family
Ralstonia solanacearum GM1000	3440	399	11.6	-146.95	Family
Pyrococcus furiosus DSM 3638	2125	142	6.68	-190.17	Species
Pyrococcus abyssi	1896	59	3.11	-217.21	Species
Corynebacterium glutamicum ATCC 13032	2993	304	10.16	-196.51	Species
Methanopyrus kandleri AV19	1687	399	23.65	-83.63	Family
Fusobacterium nucleatum subsp. nucleatum ATCC 25586	2067	335	16.21	-106.12	Division
Methanosarcina acetivorans str. C2A	4540	695	15.31	-174.12	Species
Thermoanaerobacter tengcongensis	2588	335	12.94	-120.74	Family
Streptomyces coelicolor A3(2)	7769	700	9.01	-187.56	Species
Xanthomonas campestris pv. campestris str. ATCC 33913	4181	159	3.8	-249.12	Species
Xanthomonas axonopodis pv. citri str. 306	4312	239	5.54	-242.74	Species
Buchnera aphidicola str. Sg (Schizaphis graminum)	546	1	0.18	-233.82	Species
Chlorobium tepidum TLS	2252	545	24.2	-97.83	Family
Methanosarcina mazei Goe1	3371	237	7.03	-206.07	Species
Thermosynechococcus elongatus BP-1	2475	165	6.67	-163.89	Genus
Streptococcus agalactiae 2603V/R	2124	235	11.06	-177.09	Species
Oceanobacillus iheyensis HTE 831	3500	310	8.86	-147.23	Genus
Shewanella oneidensis MR-1	4324	602	13.92	-138.47	Family
Shigella flexneri 2a str. 301	4180	110	2.63	-256.23	Genus
Wigglesworthia brevipalpis	611	4	0.65	-177.01	Genus
Bifidobacterium longum NCC2705	1727	206	11.93	-120.13	Family
Streptococcus mutans UA159	1960	198	10.1	-174.91	Species
Mycoplasma penetrans HF-2	1037	237	22.85	-88.72	Species
Pseudomonas putida KT2440	5350	428	8	-202.63	Species
Vibrio vulnificus CMCP6	4537	363	8	-210.89	Species
Bradyrhizobium japonicum USDA 110	8317	1060	12.75	-155.74	Genus
Staphylococcus epidermis ATCC 12228	2419	228	9.43	-193.8	Species
Chlostridium tetani Massachusetts E88	2373	176	7.42	-156.36	Species
Lactobacillus plantarum WCFS1	3009	408	13.56	-126.17	Family
Tropheryma whipplei TW08/27	783	91	11.62	-116.16	Family

Vibrio parahaemolyticus RIMD 2210633	4832	634	13.12	-199.02	Species
Bacteroides thetaiotaomicron VPI-5482	4778	1082	22.65	-104.35	Family
Enterococcus faecalis V583	3113	546	17.54	-130.73	Family
Streptomyces avermitilis MA-4680	7575	671	8.86	-191.34	Species
Chlamydophila caviae GPIC	998	66	6.61	-211.69	Species
Leptospira interrogans serovar lai str. 56601	4727	2138	45.23	-54.55	Family
Coxiella burnetii RSA 493	2009	649	32.3	-98.66	Family
Nitrosomonas europaea ATCC 19718	2461	227	9.22	-141.87	Family
Bacillus cereus ATCC 14579	5234	288	5.5	-219.54	Species
Bacillus anthracis Ames	5311	471	8.87	-213.99	Species
Mycobacterium bovis AF2122/97 (spoligotype 9)	3920	22	0.56	-292.39	Species
Helicobacter hepaticus ATCC51449	1875	368	19.63	-129.07	Species
Corynebacterium efficiens YS-314T	2950	289	9.8	-198.89	Species
Pirellula sp. 1	7325	3576	48.82	-49.79	Division
Haemophilus ducreyi 35000HP	1717	284	16.54	-164	Species
Candidatus Blochmannia floridanus	583	1	0.17	-193.26	Genus
Bordetella pertussis Tohama I NCTC-13251	3447	7	0.2	-290.4	Species
Bordetella parapertussis 12822 NCTC-13253	4185	14	0.33	-306.04	Species
Bordetella bronchiseptica RB50 NCTC-13252	4994	117	2.34	-283.52	Species
Prochlorococcus marinus CCMP1375(SS120)	1882	291	15.46	-155.63	Family
Synechococcus sp.WH8102	2517	388	15.42	-139.08	Genus
Mycoplasma gallisepticum R	726	76	10.47	-108.29	Species
Pseudomonas syringae pv. Tomato DC3000	5471	573	10.47	-194.24	Species
Porphyromonas gingivalis W83	1909	352	18.44	-137.14	Family
Chromobacterium violaceum ATCC 12472	4407	577	13.09	-138.63	Genus
Wolinella succinogenes	2044	150	7.34	-138.64	Genus
Photorhabdus luminescens laumondii TT01	4683	719	15.35	-155.29	Genus
Gloeobacter violaceus PCC7421	4430	682	15.4	-115.08	Family
Nanoarchaeum equitans Kin4-M	563	167	29.66	-64.64	Division
Corynebacterium diphtheriae gravis NCTC13129	2272	259	11.4	-168.21	Species
Geobacter sulfurreducens PCA	3445	580	16.84	-106.03	Family
Rhodopseudomonas palustris CGA009	4814	336	6.98	-191.98	Genus
Phytoplasma asteris OY	754	229	30.37	-67.1	Family
Bdellovibrio bacteriovorus HD100	3583	1113	31.06	-73.66	Family
Mycobacterium avium,subsp.:paratuberculosis K-10	4350	211	4.85	-200.77	Species
Mycoplasma mycoides,subsp.mycoides SC	1016	209	20.57	-85.65	Species

Appendix 5.2 – Chapter 5 Figure S1

Relationship between the numbers of orphans and Isolation Index of an Organism. The IIO for each genome in our dataset (full list of genomes given in Appendix 5.1) is plotted against (a) percentage of orphans, (b) the number of orphans greater than 200 a.a's and (c) the percentage of total orphans greater than 200 a.a's in length. In addition, each genome is classed according to the taxonomic level at which it is the only sequenced representative.



REFERENCES

1. Abril, J.F. and Guigo, R. (2000) gff2ps: visualizing genomic annotations, *Bioinformatics (Oxford, England)*, **16**, 743-744.

2. Akashi, H. and Gojobori, T. (2002) Metabolic efficiency and amino acid composition in the proteomes of Escherichia coli and Bacillus subtilis, *Proc Natl Acad Sci U S A*, **99**, 3695-3700.

3. Alimi, J.P., Poirot, O., Lopez, F. and Claverie, J.M. (2000) Reverse transcriptase-polymerase chain reaction validation of 25 "orphan" genes from Escherichia coli K-12 MG1655, *Genome Research*, **10**, 959-966.

4. Alm, E.J., Huang, K.H., Price, M.N., Koche, R.P., Keller, K., Dubchak, I.L. and Arkin, A.P. (2005) The MicrobesOnline Web site for comparative genomics, *Genome Res*, **15**, 1015-1022.

5. Altschul, S.F., Boguski, M.S., Gish, W. and Wootton, J.C. (1994) Issues in searching molecular sequence databases, *Nat Genet*, **6**, 119-129.

6. Altschul, S.F., Gish, W., Miller, W., Myers, E.W. and Lipman, D.J. (1990) Basic local alignment search tool, *J Mol Biol*, **215**, 403-410.

7. Amiri, H., Davids, W. and Andersson, S.G. (2003) Birth and death of orphan genes in Rickettsia, *Mol Biol Evol.*, **20**, 1575-1587. Epub 2003 Jun 1527.

8. Andrade, M.A., Brown, N.P., Leroy, C., Hoersch, S., de Daruvar, A., Reich, C., Franchini, A., Tamames, J., Valencia, A., Ouzounis, C. and Sander, C. (1999) Automated genome sequence analysis and annotation, *Bioinformatics (Oxford, England)*, **15**, 391-412.

9. Apweiler, R., Bairoch, A., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M.J., Natale, D.A., O'Donovan, C., Redaschi, N. and Yeh, L.S. (2004) UniProt: the Universal Protein knowledgebase, *Nucleic acids research*, **32**, D115-119.

10. Avison, D. and Fitzgerald, G. (2003) *Information Systems Development: Methodologies, Techniques and Tools.* McGraw-Hill Publishing Company.

11. Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L., Studholme, D.J., Yeats, C. and Eddy, S.R. (2004) The Pfam protein families database, *Nucleic acids research*, **32**, D138-141.

12. Baxevanis, A.D. and Ouellette, B.F.F. (2005) *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. John Wiley & Sons.

13. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J. and Wheeler, D.L. (2006) GenBank, *Nucleic acids research*, **34**, D16-20.

14. Beres, S.B., Sylva, G.L., Barbian, K.D., Lei, B., Hoff, J.S., Mammarella, N.D., Liu, M.Y., Smoot, J.C., Porcella, S.F., Parkins, L.D., Campbell, D.S., Smith, T.M., McCormick, J.K., Leung, D.Y., Schlievert, P.M. and Musser, J.M. (2002) Genome sequence of a serotype M3 strain of group A Streptococcus: phage-encoded toxins, the high-virulence phenotype, and clone emergence, *Proc Natl Acad Sci U S A*, **99**, 10078-10083.

15. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N. and Bourne, P.E. (2000) The Protein Data Bank, *Nucleic acids research*, **28**, 235-242.

16. Besemer, J. and Borodovsky, M. (2005) GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses, *Nucleic acids research*, **33**, W451-454.

17. Besemer, J., Lomsadze, A. and Borodovsky, M. (2001) GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions, *Nucleic acids research*, **29**, 2607-2618.

18. Binnewies, T.T., Motro, Y., Hallin, P.F., Lund, O., Dunn, D., La, T., Hampson, D.J., Bellgard, M., Wassenaar, T.M. and Ussery, D.W. (2006) Ten years of bacterial genome sequencing: comparative-genomics-based discoveries, *Funct Integr Genomics*, **6**, 165-185.

19. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.C., Estreicher, A., Gasteiger, E., Martin, M.J., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S.

and Schneider, M. (2003) The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003, *Nucleic acids research*, **31**, 365-370.

20. Bookman, C. (2002) *Linux Clustering: Building and Maintaining Linux Clusters.* Sams.

21. Borodovsky, M., McIninch, J.D., Koonin, E.V., Rudd, K.E., Medigue, C. and Danchin, A. (1995) Detection of new genes in a bacterial genome using Markov models for three gene classes, *Nucleic acids research*, **23**, 3554-3562.

22. Cai, J.J., Woo, P.C., Lau, S.K., Smith, D.K. and Yuen, K.Y. (2006) Accelerated evolutionary rate may be responsible for the emergence of lineage-specific genes in ascomycota, *J Mol Evol*, **63**, 1-11.

23. Cai, Y.D. and Doig, A.J. (2004) Prediction of Saccharomyces cerevisiae protein functional class from functional domain composition, *Bioinformatics (Oxford, England)*, **20**, 1292-1300.

24. Canchaya, C., Fournous, G. and Brussow, H. (2004) The impact of prophages on bacterial chromosomes, *Mol Microbiol*, **53**, 9-18.

25. Carattoli, A., Filetici, E., Villa, L., Dionisi, A.M., Ricci, A. and Luzzi, I. (2002) Antibiotic resistance genes and Salmonella genomic island 1 in Salmonella enterica serovar Typhimurium isolated in Italy, *Antimicrob Agents Chemother*, **46**, 2821-2828.

26. Casjens, S. (2003) Prophages and bacterial genomics: what have we learned so far?, *Mol Microbiol*, **49**, 277-300.

27. Champion, O.L., Gaunt, M.W., Gundogdu, O., Elmi, A., Witney, A.A., Hinds, J., Dorrell, N. and Wren, B.W. (2005) Comparative phylogenomics of the food-borne pathogen Campylobacter jejuni reveals genetic markers predictive of infection source, *Proc Natl Acad Sci U S A*, **102**, 16043-16048.

28. Charlebois, R.L., Clarke, G.D., Beiko, R.G. and St Jean, A. (2003) Characterization of species-specific genes using a flexible, web-based querying system, *FEMS Microbiol Lett*, **225**, 213-220.

29. Charlebois, R.L. and Doolittle, W.F. (2004) Computing prokaryotic gene ubiquity: rescuing the core from extinction, *Genome Res*, **14**, 2469-2477.

30. Chen, S.L., Hung, C.S., Xu, J., Reigstad, C.S., Magrini, V., Sabo, A., Blasiar, D., Bieri, T., Meyer, R.R., Ozersky, P., Armstrong, J.R., Fulton, R.S., Latreille, J.P., Spieth, J., Hooton, T.M., Mardis, E.R., Hultgren, S.J. and Gordon, J.I. (2006) Identification of genes subject to positive selection in uropathogenic strains of Escherichia coli: a comparative genomics approach, *Proc Natl Acad Sci U S A*, **103**, 5977-5982.

31. Chiapello, H., Bourgait, I., Sourivong, F., Heuclin, G., Gendrault-Jacquemard, A., Petit, M.A. and El Karoui, M. (2005) Systematic determination of the mosaic structure of bacterial genomes: species backbone versus strain-specific loops, *BMC Bioinformatics*, **6**, 171.

32. Ciccarelli, F.D., Doerks, T., von Mering, C., Creevey, C.J., Snel, B. and Bork, P. (2006) Toward automatic reconstruction of a highly resolved tree of life, *Science*, **311**, 1283-1287.

33. Cole, S.T., Eiglmeier, K., Parkhill, J., James, K.D., Thomson, N.R., Wheeler, P.R., Honore, N., Garnier, T., Churcher, C., Harris, D., Mungall, K., Basham, D., Brown, D., Chillingworth, T., Connor, R., Davies, R.M., Devlin, K., Duthoy, S., Feltwell, T., Fraser, A., Hamlin, N., Holroyd, S., Hornsby, T., Jagels, K., Lacroix, C., Maclean, J., Moule, S., Murphy, L., Oliver, K., Quail, M.A., Rajandream, M.A., Rutherford, K.M., Rutter, S., Seeger, K., Simon, S., Simmonds, M., Skelton, J., Squares, R., Squares, S., Stevens, K., Taylor, K., Whitehead, S., Woodward, J.R. and Barrell, B.G. (2001) Massive gene decay in the leprosy bacillus, *Nature*, **409**, 1007-1011.

34. Corbin, R.W., Paliy, O., Yang, F., Shabanowitz, J., Platt, M., Lyons, C.E., Jr., Root, K., McAuliffe, J., Jordan, M.I., Kustu, S., Soupene, E. and Hunt, D.F. (2003) Toward a protein profile of Escherichia coli: comparison to its transcription profile, *Proc Natl Acad Sci U S A*, **100**, 9232-9237.

35. Courcelle, J., Khodursky, A., Peter, B., Brown, P.O. and Hanawalt, P.C. (2001) Comparative gene expression profiles following UV exposure in wild-type and SOS-deficient Escherichia coli, *Genetics*, **158**, 41-64.

36. Daubin, V., Lerat, E. and Perriere, G. (2003) The source of laterally transferred genes in bacterial genomes, *Genome Biol*, **4**, R57.

37. Daubin, V. and Ochman, H. (2004) Bacterial Genomes as new gene homes: The genealogy of ORFans in E-coli, *Genome Research*, **14**, 1036-1042.

38. Daubin, V. and Ochman, H. (2004) Start-up entities in the origin of new genes, *Curr Opin Genet Dev*, **14**, 616-619.

39. Delcher, A.L., Harmon, D., Kasif, S., White, O. and Salzberg, S.L. (1999) Improved microbial gene identification with GLIMMER, *Nucleic acids research*, **27**, 4636-4641.

40. Deng, W., Burland, V., Plunkett, G., 3rd, Boutin, A., Mayhew, G.F., Liss, P., Perna, N.T., Rose, D.J., Mau, B., Zhou, S., Schwartz, D.C., Fetherston, J.D., Lindler, L.E., Brubaker, R.R., Plano, G.V., Straley, S.C., McDonough, K.A., Nilles, M.L., Matson, J.S., Blattner, F.R. and Perry, R.D. (2002) Genome sequence of Yersinia pestis KIM, *J Bacteriol*, **184**, 4601-4611.

41. Diep, B.A., Gill, S.R., Chang, R.F., Phan, T.H., Chen, J.H., Davidson, M.G., Lin, F., Lin, J., Carleton, H.A., Mongodin, E.F., Sensabaugh, G.F. and Perdreau-Remington, F. (2006) Complete genome sequence of USA300, an epidemic clone of community-acquired meticillin-resistant Staphylococcus aureus, *Lancet*, **367**, 731-739.

42. Dix, A., Finlay, J., Abowd, G. and Beale, R. (1993) *Human-Computer Interaction.* Prentice Hall.

43. Doerks, T., von Mering, C. and Bork, P. (2004) Functional clues for hypothetical proteins based on genomic context analysis in prokaryotes, *Nucleic acids research*, **32**, 6321-6326.

44. Domazet-Loso, T. and Tautz, D. (2003) An evolutionary analysis of orphan genes in Drosophila, *Genome Research*, **13**, 2213-2219.

45. Doolittle, R.F. (2002) Biodiversity: Microbial genomes multiply, *Nature*, **416**, 697-700.

46. Eisen, J.A. (2007) Environmental Shotgun Sequencing: Its Potential and Challenges for Studying the Hidden World of Microbes, *PLoS biology*, **5**, e82.

47. Eisen, J.A. and Fraser, C.M. (2003) Phylogenomics: intersection of evolution and genomics, *Science*, **300**, 1706-1707.

48. Elias, D.A., Monroe, M.E., Smith, R.D., Fredrickson, J.K. and Lipton, M.S. (2006) Confirmation of the expression of a large set of conserved hypothetical proteins in Shewanella oneidensis MR-1, *J Microbiol Methods*, **66**, 223-233.

49. Enault, F., Suhre, K. and Claverie, J.M. (2005) Phydbac "Gene Function Predictor": a gene annotation tool based on genomic context analysis, *BMC Bioinformatics*, **6**, 247.

50. Falkowski, P.G. and de Vargas, C. (2004) Genomics and evolution. Shotgun sequencing in the sea: a blast from the past?, *Science*, **304**, 58-60.

51. Faruque, S.M., Chowdhury, N., Kamruzzaman, M., Dziejman, M., Rahman, M.H., Sack, D.A., Nair, G.B. and Mekalanos, J.J. (2004) Genetic diversity and virulence potential of environmental Vibrio cholerae population in a cholera-endemic area, *Proc Natl Acad Sci U S A*, **101**, 2123-2128.

52. Ferreira, L., Berstis, V., Armstrong, J., Kendzierski, M., Neukoetter, A., Takagi, M., Bing-Wo, R., Amir, A., Murakawa, R., Hernandez, O., Magowan, J. and Bieberstein, N. (2003) *Introduction to Grid Computing with Globus*. IBM.

53. Field, D., Feil, E. and Wilson, G.A. (2005) Analyzing the evolution of infectious bacteria. In, *Genomes to Therapies.*

54. Field, D., Feil, E.J. and Wilson, G.A. (2005) Databases and software for the comparison of prokaryotic genomes, *Microbiology*, **151**, 2125-2132.

55. Field, D., Garrity, G., Gray, T., Selengut, J., Sterk, P., Thomson, N., Tatusova, T., Cochrane, G., Glockner, F.O., Kottmann, R., Lister, A.L., Tateno, Y. and Vaughan, R. (2007 (in press)) eGenomics: Cataloguing Our Complete Genome Collection III., *Comparative and Functional Genomics.*

56. Field, D., Garrity, G.M., Gray, T., Morrison, N., Selengut, J.D., Sterk, P., Tatusova, T., Thomson, N., Allen, M.J., Ashburner, M., Baldauf, S., Ballard, S., Boore, J., Cochrane, G., Cole, J., de Pamphilis, C., Edwards, R., Faruque, N., Feldman, R., Glockner, F.O., Haft, D.H., Hancock, D., Hermjakob, H., Hertz-Fowler, C., Hugenholtz, P., Joint, I., Kane, M., Kennedy, J., Kowalchuk, G., Kottmann, R., Kolker, E., Kyrpides, N.C., Leebens-Mack, J., Lewis, S.E., Lister, A.L., Lord, P., Maltsev, N., Markowitz, V.M., Martiny, J.B.H., Methe, B., Moxon, R., Nelson, K.E., Parkhill, J., Sansone, S., Spiers, A.J., Stevens, R., Swift, P., Taylor, C., Tateno, Y., Tett, A., Turner, S.L., Ussery, D., Vaughan, R., Ward, N.L., Whetzel, T., Wilson, G.A. and Wipat, A. (2007) Towards a richer description of our complete collection of genomes and metagenomes: the "Minimal Information about a Genome Sequence", *Nature Biotechnology*, **(in press)**.

57. Field, D., Tiwari, B., Booth, T., Houten, S., Swan, D., Bertrand, N. and Thurston, M. (2006) Open software for biologists: from famine to feast, *Nat Biotechnol*, **24**, 801-803.

58. Field, D., Wilson, G. and van der Gast, C. (2006) How do we compare hundreds of bacterial genomes?, *Curr Opin Microbiol*, **9**, 499-504.

59. Floyd, M.M., Tang, J., Kane, M. and Emerson, D. (2005) Captured diversity in a culture collection: case study of the geographic and habitat distributions of environmental isolates held at the american type culture collection, *Appl Environ Microbiol*, **71**, 2813-2823.

60. Foerstner, K.U., von Mering, C., Hooper, S.D. and Bork, P. (2005) Environments shape the nucleotide composition of genomes, *EMBO Rep.*, **6**, 1208-1213.

61. Fuhrman, J. (2003) Genome sequences from the sea, *Nature*, **424**, 1001-1002.

62. Fukuchi, S. and Nishikawa, K. (2004) Estimation of the number of authentic orphan genes in bacterial genomes, *DNA Res*, **11**, 219-231, 311-313.

63. Galperin, M.Y. and Kolker, E. (2006) New metrics for comparative genomics, *Curr Opin Biotechnol.*, **17**, 440-447. Epub 2006 Sep 2015.

64. Galperin, M.Y. and Koonin, E.V. (2004) 'Conserved hypothetical' proteins: prioritization of targets for experimental study, *Nucleic Acids Res.*, **32**, 5452-5463. Print 2004.

65. Garcia Martin, H., Ivanova, N., Kunin, V., Warnecke, F., Barry, K.W., McHardy, A.C., Yeates, C., He, S., Salamov, A.A., Szeto, E., Dalin, E., Putnam, N.H., Shapiro, H.J., Pangilinan, J.L., Rigoutsos, I., Kyrpides, N.C., Blackall, L.L., McMahon, K.D. and Hugenholtz, P. (2006) Metagenomic analysis of two enhanced biological phosphorus removal (EBPR) sludge communities, *Nat Biotechnol*, **24**, 1263-1269.

66. Garrity, G.M. (2001) *Bergey's Manual of Systematic Bacteriology.* Springer-Verlag, New York.

67. Gevaert, K., Van Damme, J., Goethals, M., Thomas, G.R., Hoorelbeke, B., Demol, H., Martens, L., Puype, M., Staes, A. and Vandekerckhove, J. (2002) Chromatographic isolation of methionine-containing peptides for gel-free proteome analysis: identification of more than 800 Escherichia coli proteins, *Mol Cell Proteomics*, **1**, 896-903.

68. Gilks, W.R., Audit, B., de Angelis, D., Tsoka, S. and Ouzounis, C.A. (2005) Percolation of annotation errors through hierarchically structured protein sequence databases, *Math Biosci*, **193**, 223-234.

69. Gill, S.R., Pop, M., Deboy, R.T., Eckburg, P.B., Turnbaugh, P.J., Samuel, B.S., Gordon, J.I., Relman, D.A., Fraser-Liggett, C.M. and Nelson, K.E. (2006) Metagenomic analysis of the human distal gut microbiome, *Science*, **312**, 1355-1359.

70. Goodner, B., Hinkle, G., Gattung, S., Miller, N., Blanchard, M., Qurollo, B., Goldman, B.S., Cao, Y., Askenazi, M., Halling, C., Mullin, L., Houmiel, K., Gordon, J., Vaudin, M., Iartchouk, O., Epp, A., Liu, F., Wollam, C., Allinger, M., Doughty, D., Scott, C., Lappas, C., Markelz, B., Flanagan, C., Crowell, C., Gurson, J., Lomo, C., Sear, C., Strub, G., Cielo, C. and Slater, S. (2001) Genome sequence of the plant pathogen and biotechnology agent Agrobacterium tumefaciens C58, *Science*, **294**, 2323-2328.

71. Graham, D.E., Overbeek, R., Olsen, G.J. and Woese, C.R. (2000) An archaeal genomic signature, *Proc Natl Acad Sci U S A*, **97**, 3304-3308.

72. Greenspan, J. and Bulger, B. (2001) *MySQL/PHP Database Applications.* IDG Books.

73. Gutierrez-Rios, R.M., Rosenblueth, D.A., Loza, J.A., Huerta, A.M., Glasner, J.D., Blattner, F.R. and Collado-Vides, J. (2003) Regulatory network of Escherichia coli: consistency between literature knowledge and microarray profiles, *Genome Res*, **13**, 2435-2443.

74. Hallin, P.F. and Ussery, D.W. (2004) CBS genome atlas database: a dynamic storage for bioinformatic results and sequence data, *Bioinformatics (Oxford, England).*

75. Harrison, P.M., Carriero, N., Liu, Y. and Gerstein, M. (2003) A "polyORFomic" analysis of prokaryote genomes using disabled-homology filtering reveals conserved but undiscovered short ORFs, *J Mol Biol*, **333**, 885-892.

76. Harrison, P.M. and Gerstein, M. (2002) Studying genomes through the aeons: protein families, pseudogenes and proteome evolution, *J Mol Biol*, **318**, 1155-1174.

77. Hayashi, T., Makino, K., Ohnishi, M., Kurokawa, K., Ishii, K., Yokoyama, K., Han, C.G., Ohtsubo, E., Nakayama, K., Murata, T., Tanaka, M., Tobe, T., Iida, T., Takami, H., Honda, T., Sasakawa, C., Ogasawara, N., Yasunaga, T., Kuhara, S., Shiba, T., Hattori, M. and Shinagawa, H. (2001) Complete genome sequence of enterohemorrhagic Escherichia coli O157:H7 and genomic comparison with a laboratory strain K-12, *DNA Res*, **8**, 11-22.

78. Heidelberg, J.F., Eisen, J.A., Nelson, W.C., Clayton, R.A., Gwinn, M.L., Dodson, R.J., Haft, D.H., Hickey, E.K., Peterson, J.D., Umayam, L., Gill, S.R., Nelson, K.E., Read, T.D., Tettelin, H., Richardson, D., Ermolaeva, M.D., Vamathevan, J., Bass, S., Qin, H., Dragoi, I., Sellers, P., McDonald, L., Utterback, T., Fleishmann, R.D., Nierman, W.C., White, O., Salzberg, S.L., Smith, H.O., Colwell, R.R., Mekalanos, J.J., Venter, J.C. and Fraser, C.M. (2000) DNA sequence of both chromosomes of the cholera pathogen Vibrio cholerae, *Nature*, **406**, 477-483.

79. Heizer, E.M., Jr., Raiford, D.W., Raymer, M.L., Doom, T.E., Miller, R.V. and Krane, D.E. (2006) Amino acid cost and codon-usage biases in 6 prokaryotic genomes: a whole-genome analysis, *Mol Biol Evol*, **23**, 1670-1680.

80. Henikoff, S. and Henikoff, J.G. (1992) Amino acid substitution matrices from protein blocks, *Proc Natl Acad Sci U S A*, **89**, 10915-10919.

81. Hirsh, A.E. and Fraser, H.B. (2001) Protein dispensability and rate of evolution, *Nature*, **411**, 1046-1049.

82. Holden, M., Rajandream, M.A. and Bentley, S. (2005) Food for thought, *Nat Rev Microbiol*, **3**, 912-913.

83. Hsiao, W.W., Ung, K., Aeschliman, D., Bryan, J., Finlay, B.B. and Brinkman, F.S. (2005) Evidence of a large novel gene pool associated with prokaryotic genomic islands, *PLoS Genet*, **1**, e62.

84. Hubbard, T. and Birney, E. (2000) Open annotation offers a democratic solution to genome sequencing., *Nature*, **403**, 825.

85. Hurst, L.D., Feil, E.J. and Rocha, E.P. (2006) Protein evolution: causes of trends in amino-acid gain and loss, *Nature.*, **442**, E11-12; discussion E12.

86. Jensen, L.J., Skovgaard, M., Sicheritz-Ponten, T., Jorgensen, M.K., Lundegaard, C., Pedersen, C.C., Petersen, N. and Ussery, D. (2003) Analysis of two large functionally uncharacterized regions in the Methanopyrus kandleri AV19 genome, *Bmc Genomics*, **4**, art. no.-12.

87. Joseph, J. and Fellenstein, C. (2003) *Grid Computing.* IBM Press.

88. Kang, Y., Weber, K.D., Qiu, Y., Kiley, P.J. and Blattner, F.R. (2005) Genome-wide expression analysis indicates that FNR of Escherichia coli K-12 regulates a large number of genes of unknown function, *J Bacteriol*, **187**, 1135-1160.

89. Karp, P.D. (2004) Call for an enzyme genomics initiative, *Genome Biol*, **5**, 401.

90. Kawarabayasi, Y., Hino, Y., Horikawa, H., Yamazaki, S., Haikawa, Y., Jin-no, K., Takahashi, M., Sekine, M., Baba, S., Ankai, A., Kosugi, H., Hosoyama, A.,

Fukui, S., Nagai, Y., Nishijima, K., Nakazawa, H., Takamiya, M., Masuda, S., Funahashi, T., Tanaka, T., Kudoh, Y., Yamazaki, J., Kushida, N., Oguchi, A., Kikuchi, H. and et al. (1999) Complete genome sequence of an aerobic hyper-thermophilic crenarchaeon, Aeropyrum pernix K1, *DNA Res*, **6**, 83-101, 145-152.

91. Khodursky, A.B., Peter, B.J., Cozzarelli, N.R., Botstein, D., Brown, P.O. and Yanofsky, C. (2000) DNA microarray analysis of gene expression in response to physiological and genetic changes that affect tryptophan metabolism in Escherichia coli, *Proc Natl Acad Sci U S A*, **97**, 12170-12175.

92. Kolker, E., Makarova, K.S., Shabalina, S., Picone, A.F., Purvine, S., Holzman, T., Cherny, T., Armbruster, D., Munson, R.S., Jr., Kolesov, G., Frishman, D. and Galperin, M.Y. (2004) Identification and functional analysis of 'hypothetical' genes expressed in Haemophilus influenzae, *Nucleic acids research*, **32**, 2353-2361.

93. Kolker, E., Picone, A.F., Galperin, M.Y., Romine, M.F., Higdon, R., Makarova, K.S., Kolker, N., Anderson, G.A., Qiu, X., Auberry, K.J., Babnigg, G., Beliaev, A.S., Edlefsen, P., Elias, D.A., Gorby, Y.A., Holzman, T., Klappenbach, J.A., Konstantinidis, K.T., Land, M.L., Lipton, M.S., McCue, L.A., Monroe, M., Pasa-Tolic, L., Pinchuk, G., Purvine, S., Serres, M.H., Tsapin, S., Zakrajsek, B.A., Zhu, W., Zhou, J., Larimer, F.W., Lawrence, C.E., Riley, M., Collart, F.R., Yates, J.R., 3rd, Smith, R.D., Giometti, C.S., Nealson, K.H., Fredrickson, J.K. and Tiedje, J.M. (2005) Global profiling of Shewanella oneidensis MR-1: expression of hypothetical genes and improved functional annotations, *Proc Natl Acad Sci U S A.*, **102**, 2099-2104. Epub 2005 Jan 2031.

94. Korbel, J.O., Doerks, T., Jensen, L.J., Perez-Iratxeta, C., Kaczanowski, S., Hooper, S.D., Andrade, M.A. and Bork, P. (2005) Systematic association of genes to phenotypes by genome and literature mining, *PLoS biology*, **3**, e134.

95. Korf, I., Yandell, M. and Bedell, J. (2003) *BLAST.* O'Reilly.

96. Kosuge, T., Abe, T., Okido, T., Tanaka, N., Hirahata, M., Maruyama, Y., Mashima, J., Tomiki, A., Kurokawa, M., Himeno, R., Fukuchi, S., Miyazaki, S., Gojobori, T., Tateno, Y. and Sugawara, H. (2006) Exploration and Grading of Possible Genes from 183 Bacterial Strains by a Common Protocol to

Identification of New Genes: Gene Trek in Prokaryote Space (GTPS), *DNA Res*, **13**, 13.

97. Krylov, D.M., Wolf, Y.I., Rogozin, I.B. and Koonin, E.V. (2003) Gene loss, protein sequence divergence, gene dispensability, expression level, and interactivity are correlated in eukaryotic evolution, *Genome Res*, **13**, 2229-2235.

98. Kunin, V., Cases, I., Enright, A.J., de Lorenzo, V. and Ouzounis, C.A. (2003) Myriads of protein families, and still counting, *Genome Biology*, **4**, art. no.-401.

99. Kwan, T., Liu, J., Dubow, M., Gros, P. and Pelletier, J. (2006) Comparative Genomic Analysis of 18 Pseudomonas aeruginosa Bacteriophages, *J Bacteriol*, **188**, 1184-1187.

100. Lan, R. and Reeves, P.R. (2000) Intraspecies variation in bacterial genomes: the need for a species genome concept, *Trends Microbiol*, **8**, 396-401.

101. Lawrence, J. (2003) When ELFs are ORFs, but don't act like them, *Trends in Genetics*, **19**, 131-132.

102. Lawrence, J.G., Hendrix, R.W. and Casjens, S. (2001) Where are the pseudogenes in bacterial genomes?, *Trends Microbiol*, **9**, 535-540.

103. Lerat, E. and Ochman, H. (2004) Psi-Phi: exploring the outer limits of bacterial pseudogenes, *Genome Res*, **14**, 2273-2278.

104. Lerat, E. and Ochman, H. (2005) Recognizing the pseudogenes in bacterial genomes, *Nucleic acids research*, **33**, 3125-3132.

105. Lesk, A.M. (2005) *Introduction to Bioinformatics.* Oxford University Press.

106. Liolios, K., Tavernarakis, N., Hugenholtz, P. and Kyrpides, N.C. (2006) The Genomes On Line Database (GOLD) v.2: a monitor of genome projects worldwide, *Nucleic Acids Res.*, **34**, D332-334.

107. Liu, Y., Harrison, P.M., Kunin, V. and Gerstein, M. (2004) Comprehensive analysis of pseudogenes in prokaryotes: widespread gene decay and failure of putative horizontally transferred genes, *Genome Biol*, **5**, R64.

108. Lowe, T.M. and Eddy, S.R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence, *Nucleic acids research*, **25**, 955-964.

109. Lukashin, A.V. and Borodovsky, M. (1998) GeneMark.hmm: new solutions for gene finding, *Nucleic Acids Res.*, **26**, 1107-1115.

110. Maglott, D., Ostell, J., Pruitt, K.D. and Tatusova, T. (2007) Entrez Gene: gene-centered information at NCBI, *Nucleic acids research*, **35**, D26-31.

111. Manning, P.A. (1997) The tcp gene cluster of Vibrio cholerae, *Gene*, **192**, 63-70.

112. Markowitz, V.M., Korzeniewski, F., Palaniappan, K., Szeto, E., Werner, G., Padki, A., Zhao, X., Dubchak, I., Hugenholtz, P., Anderson, I., Lykidis, A., Mavromatis, K., Ivanova, N. and Kyrpides, N.C. (2006) The integrated microbial genomes (IMG) system, *Nucleic acids research*, **34**, D344-348.

113. Martiny, J.B.H. and Field, D. (2005) Ecological perspectives on our complete genome collection, *Ecology Letters*, **8**, 1334-1345.

114. Mausolf, J. (2005) Communicating outside the flock, Part 1: Condor-G with Globus.

115. Mausolf, J. (2005) An eagle-eye view of the Condor project.

116. Mazumder, R., Natale, D.A., Murthy, S., Thiagarajan, R. and Wu, C.H. (2005) Computational identification of strain-, species- and genus-specific proteins, *BMC Bioinformatics*, **6**, 279.

117. McGinnis, S. and Madden, T.L. (2004) BLAST: at the core of a powerful and diverse set of sequence analysis tools, *Nucleic acids research*, **32**, W20-25.

118. McInerney, J.O. (2002) Bioinformatics in a post-genomics world--the need for an inclusive approach, *Pharmacogenomics J*, **2**, 207-208.

119. McMeekin, A. and Harvey, M. (2002) The formation of bioinformatics knowledge markets: An 'economies of knowledge' approach. *Centre for Research on Innovation & Competition*. Manchester.

120. Medini, D., Donati, C., Tettelin, H., Masignani, V. and Rappuoli, R. (2005) The microbial pan-genome, *Curr Opin Genet Dev.*, **15**, 589-594. Epub 2005 Sep 2026.

121. Mira, A., Klasson, L. and Andersson, S.G. (2002) Microbial genome evolution: sources of variability, *Curr Opin Microbiol*, **5**, 506-512.

122. Misra, R.V., Horler, R.S., Reindl, W., Goryanin, II and Thomas, G.H. (2005) EchoBASE: an integrated post-genomic database for Escherichia coli, *Nucleic Acids Res.*, **33**, D329-333.

123. Mitchell, J.A., Aronson, A.R., Mork, J.G., Folk, L.C., Humphrey, S.M. and Ward, J.M. (2003) Gene indexing: characterization and analysis of NLM's GeneRIFs, *AMIA Annu Symp Proc*, 460-464.

124. Moran, N.A. (2002) Microbial minimalism: genome reduction in bacterial pathogens, *Cell*, **108**, 583-586.

125. Moxon, E.R. and Higgins, C.F. (1997) E-coli gene sequence - A blueprint for life, *Nature*, **389**, 120-121.

126. Nakayama, K., Takashima, K., Ishihara, H., Shinomiya, T., Kageyama, M., Kanaya, S., Ohnishi, M., Murata, T., Mori, H. and Hayashi, T. (2000) The R-type pyocin of Pseudomonas aeruginosa is related to P2 phage, and the F-type is related to lambda phage, *Mol Microbiol*, **38**, 213-231.

127. Navarre, W.W., Porwollik, S., Wang, Y., McClelland, M., Rosen, H., Libby, S.J. and Fang, F.C. (2006) Selective silencing of foreign DNA with low GC content by the H-NS protein in Salmonella, *Science*, **313**, 236-238.

128. Nei, M. (2005) Selectionism and neutralism in molecular evolution, *Mol Biol Evol.*, **22**, 2318-2342.

129. Nitschke, P., Guerdoux-Jamet, P., Chiapello, H., Faroux, G., Henaut, C., Henaut, A. and Danchin, A. (1998) Indigo: a World-Wide-Web review of genomes and gene functions, *FEMS microbiology reviews*, **22**, 207-227.

130. Ochman, H. (2002) Distinguishing the ORFs from the ELFs: short bacterial genes and the annotation of genomes, *Trends Genet*, **18**, 335-337.

131. Ochman, H. (2003) Neutral mutations and neutral substitutions in bacterial genomes, *Mol Biol Evol.*, **20**, 2091-2096. Epub 2003 Aug 2029.

132. Ochman, H., Lerat, E. and Daubin, V. (2005) Examining bacterial species under the specter of gene transfer and exchange, *Proc Natl Acad Sci U S A*, **102 Suppl 1**, 6595-6599.

133. Ohnishi, M., Kurokawa, K. and Hayashi, T. (2001) Diversification of Escherichia coli genomes: are bacteriophages the major contributors?, *Trends Microbiol*, **9**, 481-485.

134. Orman, L. (1998) Evolutionary Development of Information Systems., *Journal of Management Information Systems*, **5**.

135. Overbeek, R., Begley, T., Butler, R.M., Choudhuri, J.V., Chuang, H.Y., Cohoon, M., de Crecy-Lagard, V., Diaz, N., Disz, T., Edwards, R., Fonstein, M., Frank, E.D., Gerdes, S., Glass, E.M., Goesmann, A., Hanson, A., Iwata-Reuyl, D., Jensen, R., Jamshidi, N., Krause, L., Kubal, M., Larsen, N., Linke, B., McHardy, A.C., Meyer, F., Neuweger, H., Olsen, G., Olson, R., Osterman, A., Portnoy, V., Pusch, G.D., Rodionov, D.A., Ruckert, C., Steiner, J., Stevens, R., Thiele, I., Vassieva, O., Ye, Y., Zagnitko, O. and Vonstein, V. (2005) The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes, *Nucleic Acids Res.*, **33**, 5691-5702. Print 2005.

136. Pal, C., Papp, B. and Hurst, L.D. (2001) Does the recombination rate affect the efficiency of purifying selection? The yeast genome provides a partial answer, *Mol Biol Evol*, **18**, 2323-2326.

137. Pal, C., Papp, B. and Hurst, L.D. (2003) Genomic function: Rate of evolution and gene dispensability, *Nature*, **421**, 496-497; discussion 497-498.

138. Pedulla, M.L., Ford, M.E., Houtz, J.M., Karthikeyan, T., Wadsworth, C., Lewis, J.A., Jacobs-Sera, D., Falbo, J., Gross, J., Pannunzio, N.R., Brucker, W., Kumar, V., Kandasamy, J., Keenan, L., Bardarov, S., Kriakov, J., Lawrence, J.G., Jacobs, W.R., Jr., Hendrix, R.W. and Hatfull, G.F. (2003) Origins of highly mosaic mycobacteriophage genomes, *Cell*, **113**, 171-182.

139. Pellegrini, M. and Yeates, T.O. (1999) Searching for frameshift evolutionary relationships between protein sequence families, *Proteins*, **37**, 278-283.

140. Perna, N.T., Plunkett, G., 3rd, Burland, V., Mau, B., Glasner, J.D., Rose, D.J., Mayhew, G.F., Evans, P.S., Gregor, J., Kirkpatrick, H.A., Posfai, G., Hackett, J., Klink, S., Boutin, A., Shao, Y., Miller, L., Grotbeck, E.J., Davis, N.W., Lim, A., Dimalanta, E.T., Potamousis, K.D., Apodaca, J., Anantharaman, T.S., Lin, J., Yen, G., Schwartz, D.C., Welch, R.A. and Blattner, F.R. (2001) Genome sequence of enterohaemorrhagic Escherichia coli O157:H7, *Nature*, **409**, 529-533.

141. Peterson, J.D., Umayam, L.A., Dickinson, T., Hickey, E.K. and White, O. (2001) The Comprehensive Microbial Resource, *Nucleic acids research*, **29**, 123-125.

142. Petrov, D.A. and Hartl, D.L. (1999) Patterns of nucleotide substitution in Drosophila and mammalian genomes, *Proc Natl Acad Sci U S A.*, **96**, 1475-1479.

143. Pevsner, J. (2003) *Bioinformatics and Functional Genomics.* Wiley-Liss.

144. Posfai, G., Plunkett, G., 3rd, Feher, T., Frisch, D., Keil, G.M., Umenhoffer, K., Kolisnychenko, V., Stahl, B., Sharma, S.S., de Arruda, M., Burland, V., Harcum, S.W. and Blattner, F.R. (2006) Emergent properties of reduced-genome Escherichia coli, *Science.*, **312**, 1044-1046. Epub 2006 Apr 1027.

145. Pruitt, K.D., Tatusova, T. and Maglott, D.R. (2005) NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins, *Nucleic acids research*, **33**, D501-504.

146. Purdy, A., Rohwer, F., Edwards, R., Azam, F. and Bartlett, D.H. (2005) A glimpse into the expanded genome content of Vibrio cholerae through

identification of genes present in environmental strains, *J Bacteriol*, **187**, 2992-3001.

147. Ramarapu, N., Simkin, M. and Raisinghani, M. (1999) The analysis and study of the impact of technology on groups - a conceptual framework., *International Journal of Information Management*, **19**, 157-172.

148. Raoult, D., Audic, S., Robert, C., Abergel, C., Renesto, P., Ogata, H., La Scola, B., Suzan, M. and Claverie, J.M. (2004) The 1.2-megabase genome sequence of Mimivirus, *Science*, **306**, 1344-1350.

149. Rechenmann, F. (1995) Knowledge Bases and Computational Biology. In Mars, N. (ed), *Towards Very Large Knowledge Bases.* IOS Press, 7-12.

150. Ren, S.X., Fu, G., Jiang, X.G., Zeng, R., Miao, Y.G., Xu, H., Zhang, Y.X., Xiong, H., Lu, G., Lu, L.F., Jiang, H.Q., Jia, J., Tu, Y.F., Jiang, J.X., Gu, W.Y., Zhang, Y.Q., Cai, Z., Sheng, H.H., Yin, H.F., Zhang, Y., Zhu, G.F., Wan, M., Huang, H.L., Qian, Z., Wang, S.Y., Ma, W., Yao, Z.J., Shen, Y., Qiang, B.Q., Xia, Q.C., Guo, X.K., Danchin, A., Saint Girons, I., Somerville, R.L., Wen, Y.M., Shi, M.H., Chen, Z., Xu, J.G. and Zhao, G.P. (2003) Unique physiological and pathogenic features of Leptospira interrogans revealed by whole-genome sequencing, *Nature*, **422**, 888-893.

151. Roberts, R.J. (2004) Identifying protein function--a call for community action, *PLoS Biol.*, **2**, E42. Epub 2004 Mar 2016.

152. Roberts, R.J., Karp, P., Kasif, S., Linn, S. and Buckley, M.R. (2005) An Experimental Approach to Genome Annotation, *Critical Issues Colloquia Report, American Society of Microbiology.*

153. Rocap, G., Larimer, F.W., Lamerdin, J., Malfatti, S., Chain, P., Ahlgren, N.A., Arellano, A., Coleman, M., Hauser, L., Hess, W.R., Johnson, Z.I., Land, M., Lindell, D., Post, A.F., Regala, W., Shah, M., Shaw, S.L., Steglich, C., Sullivan, M.B., Ting, C.S., Tolonen, A., Webb, E.A., Zinser, E.R. and Chisholm, S.W. (2003) Genome divergence in two Prochlorococcus ecotypes reflects oceanic niche differentiation, *Nature*, **424**, 1042-1047.

154. Rocha, E.P. and Danchin, A. (2002) Base composition bias might result from competition for metabolic resources, *Trends Genet*, **18**, 291-294.

155. Romine, M.F., Elias, D.A., Monroe, M.E., Auberry, K., Fang, R., Fredrickson, J.K., Anderson, G.A., Smith, R.D. and Lipton, M.S. (2004) Validation of Shewanella oneidensis MR-1 small proteins by AMT tag-based proteome analysis, *Omics.*, **8**, 239-254.

156. Ruby, E.G., Urbanowski, M., Campbell, J., Dunn, A., Faini, M., Gunsalus, R., Lostroh, P., Lupp, C., McCann, J., Millikan, D., Schaefer, A., Stabb, E., Stevens, A., Visick, K., Whistler, C. and Greenberg, E.P. (2005) Complete genome sequence of Vibrio fischeri: a symbiotic bacterium with pathogenic congeners, *Proc Natl Acad Sci U S A*, **102**, 3004-3009.

157. Rusch, D.B., Halpern, A.L., Sutton, G., Heidelberg, K.B., Williamson, S., Yooseph, S., Wu, D., Eisen, J.A., Hoffman, J.M., Remington, K., Beeson, K., Tran, B., Smith, H., Baden-Tillson, H., Stewart, C., Thorpe, J., Freeman, J., Andrews-Pfannkoch, C., Venter, J.E., Li, K., Kravitz, S., Heidelberg, J.F., Utterback, T., Rogers, Y.H., Falcon, L.I., Souza, V., Bonilla-Rosso, G., Eguiarte, L.E., Karl, D.M., Sathyendranath, S., Platt, T., Bermingham, E., Gallardo, V., Tamayo-Castillo, G., Ferrari, M.R., Strausberg, R.L., Nealson, K., Friedman, R., Frazier, M. and Venter, J.C. (2007) The Sorcerer II Global Ocean Sampling Expedition: Northwest Atlantic through Eastern Tropical Pacific, *PLoS biology*, **5**, e77.

158. Rutherford, K., Parkhill, J., Crook, J., Horsnell, T., Rice, P., Rajandream, M.A. and Barrell, B. (2000) Artemis: sequence visualization and annotation, *Bioinformatics (Oxford, England)*, **16**, 944-945.

159. Salama, N., Guillemin, K., McDaniel, T.K., Sherlock, G., Tompkins, L. and Falkow, S. (2000) A whole-genome microarray reveals genetic diversity among Helicobacter pylori strains, *Proc Natl Acad Sci U S A*, **97**, 14668-14673.

160. Salzberg, S.L., Delcher, A.L., Kasif, S. and White, O. (1998) Microbial gene identification using interpolated Markov models, *Nucleic Acids Res.*, **26**, 544-548.

161. Shmuely, H., Dinitz, E., Dahan, I., Eichler, J., Fischer, D. and Shaanan, B. (2004) Poorly conserved ORFs in the genome of the archaea Halobacterium sp NRC-1 correspond to expressed proteins, *Bioinformatics (Oxford, England)*, **20**, 1248-1253.

162. Siew, N., Azaria, Y. and Fischer, D. (2004) The ORFanage: an ORFan database, *Nucleic acids research*, **32 Database issue**, D281-283.

163. Siew, N. and Fischer, D. (2003) Analysis of singleton ORFans in fully sequenced microbial genomes, *Proteins-Structure Function and Genetics*, **53**, 241-251.

164. Siew, N. and Fischer, D. (2003) Twenty thousand ORFan microbial protein families for the biologist?, *Structure*, **11**, 7-9.

165. Siew, N. and Fischer, D. (2004) Structural biology sheds light on the puzzle of genomic ORFans, *J Mol Biol*, **342**, 369-373.

166. Siew, N., Saini, H.K. and Fischer, D. (2005) A putative novel alpha/beta hydrolase ORFan family in Bacillus, *FEBS Lett*, **579**, 3175-3182.

167. Skovgaard, M., Jensen, L.J., Brunak, S., Ussery, D. and Krogh, A. (2001) On the total number of genes and their length distribution in complete microbial genomes, *Trends in Genetics*, **17**, 425-428.

168. Smoot, J.C., Barbian, K.D., Van Gompel, J.J., Smoot, L.M., Chaussee, M.S., Sylva, G.L., Sturdevant, D.E., Ricklefs, S.M., Porcella, S.F., Parkins, L.D., Beres, S.B., Campbell, D.S., Smith, T.M., Zhang, Q., Kapur, V., Daly, J.A., Veasy, L.G. and Musser, J.M. (2002) Genome sequence and comparative microarray analysis of serotype M18 group A Streptococcus strains associated with acute rheumatic fever outbreaks, *Proc Natl Acad Sci U S A*, **99**, 4668-4673.

169. Snel, B., Bork, P. and Huynen, M.A. (2002) Genomes in flux: the evolution of archaeal and proteobacterial gene content, *Genome Res*, **12**, 17-25.

170. Snyder, L.A., Davies, J.K. and Saunders, N.J. (2004) Microarray genomotyping of key experimental strains of Neisseria gonorrhoeae reveals gene complement

diversity and five new neisserial genes associated with Minimal Mobile Elements, *Bmc Genomics*, **5**, 23.

171. Stajich, J.E., Block, D., Boulez, K., Brenner, S.E., Chervitz, S.A., Dagdigian, C., Fuellen, G., Gilbert, J.G., Korf, I., Lapp, H., Lehvaslaiho, H., Matsalla, C., Mungall, C.J., Osborne, B.I., Pocock, M.R., Schattner, P., Senger, M., Stein, L.D., Stupka, E., Wilkinson, M.D. and Birney, E. (2002) The Bioperl toolkit: Perl modules for the life sciences, *Genome Res*, **12**, 1611-1618.

172. Stevens, R.C. (2004) Long live structural biology, *Nat Struct Mol Biol*, **11**, 293-295.

173. Stothard, P. and Wishart, D.S. (2005) Circular genome visualization and exploration using CGView, *Bioinformatics (Oxford, England)*, **21**, 537-539.

174. Taoka, M., Yamauchi, Y., Shinkawa, T., Kaji, H., Motohashi, W., Nakayama, H., Takahashi, N. and Isobe, T. (2004) Only a small subset of the horizontally transferred chromosomal genes in Escherichia coli are translated into proteins, *Mol Cell Proteomics*, **3**, 780-787.

175. Tatusov, R.L., Fedorova, N.D., Jackson, J.D., Jacobs, A.R., Kiryutin, B., Koonin, E.V., Krylov, D.M., Mazumder, R., Mekhedov, S.L., Nikolskaya, A.N., Rao, B.S., Smirnov, S., Sverdlov, A.V., Vasudevan, S., Wolf, Y.I., Yin, J.J. and Natale, D.A. (2003) The COG database: an updated version includes eukaryotes, *BMC Bioinformatics.*, **4**, 41. Epub 2003 Sep 2011.

176. Tett, A., Spiers, A.J., Crossman, L.C., Ager, D., Ciric, L., Dow, J.M., Fry, J., Harris, D., Lilley, A., Parkhill, J., Quail, M.A., Rainey, P.B., Saunders, N.J., Seeger, K., Snyder, L.A.S., Squares, R., Thomas, C., Turner, S.L., Zhang, X., Field, D. and Bailey, M.J. ((submitted)) Sequence-based analysis of pQBR103; a representative of a unique, transferproficient mega plasmid resident in the microbial community of sugar beet, *ISME*.

177. Tettelin, H., Masignani, V., Cieslewicz, M.J., Donati, C., Medini, D., Ward, N.L., Angiuoli, S.V., Crabtree, J., Jones, A.L., Durkin, A.S., Deboy, R.T., Davidsen, T.M., Mora, M., Scarselli, M., Margarit y Ros, I., Peterson, J.D., Hauser, C.R., Sundaram, J.P., Nelson, W.C., Madupu, R., Brinkac, L.M., Dodson, R.J., Rosovitz, M.J., Sullivan, S.A., Daugherty, S.C., Haft, D.H., Selengut, J., Gwinn,

M.L., Zhou, L., Zafar, N., Khouri, H., Radune, D., Dimitrov, G., Watkins, K., O'Connor, K.J., Smith, S., Utterback, T.R., White, O., Rubens, C.E., Grandi, G., Madoff, L.C., Kasper, D.L., Telford, J.L., Wessels, M.R., Rappuoli, R. and Fraser, C.M. (2005) Genome analysis of multiple pathogenic isolates of Streptococcus agalactiae: implications for the microbial "pan-genome", *Proc Natl Acad Sci U S A.*, **102**, 13950-13955. Epub 12005 Sep 13919.

178. Thomas, C.M. and Nielsen, K.M. (2005) Mechanisms of, and barriers to, horizontal gene transfer between bacteria, *Nat Rev Microbiol*, **3**, 711-721.

179. Unger, R., Uliel, S. and Havlin, S. (2003) Scaling law in sizes of protein sequence families: from super-families to orphan genes, *Proteins*, **51**, 569-576.

180. Ussery, D.W. and Hallin, P.F. (2004) Genome Update: annotation quality in sequenced microbial genomes, *Microbiology.*, **150**, 2015-2017.

181. Venter, J.C., Remington, K., Heidelberg, J.F., Halpern, A.L., Rusch, D., Eisen, J.A., Wu, D., Paulsen, I., Nelson, K.E., Nelson, W., Fouts, D.E., Levy, S., Knap, A.H., Lomas, M.W., Nealson, K., White, O., Peterson, J., Hoffman, J., Parsons, R., Baden-Tillson, H., Pfannkoch, C., Rogers, Y.H. and Smith, H.O. (2004) Environmental genome shotgun sequencing of the Sargasso Sea, *Science*, **304**, 66-74.

182. von Mering, C., L, J.J., Kuhn, M., Chaffron, S., Doerks, T., Kruger, B., Snel, B. and Bork, P. (2006) STRING 7--recent developments in the integration and prediction of protein interactions, *Nucleic acids research*, **10**, 10.

183. Waldor, M.K. and Mekalanos, J.J. (1996) Lysogenic conversion by a filamentous phage encoding cholera toxin, *Science*, **272**, 1910-1914.

184. Wall, L., Christiansen, T. and Schwartz, R. (1996) *Programming Perl.* O'Reilly.

185. Wang, K. (2006) Gene-function wiki would let biologists pool worldwide resources, *Nature*, **439**, 534.

186. Wernegreen, J.J. (2002) Genome evolution in bacterial endosymbionts of insects, *Nature Review Genetics*, **3**, 850-861.

187. Wilson, G.A., Bertrand, N., Patel, Y., Hughes, J.B., Feil, E.J. and Field, D. (2005) Orphans as taxonomically restricted and ecologically important genes, *Microbiology*, **151**, 2499-2501.

188. Wilson, G.A., Feil, E.J., Lilley, A.K. and Field, D. (2007) Large-Scale Comparative Genomic Ranking of Taxonomically Restricted Genes (TRGs) in Bacterial and Archaeal Genomes, *PLoS ONE*, **2**, e324.

189. Wood, D.W., Setubal, J.C., Kaul, R., Monks, D.E., Kitajima, J.P., Okura, V.K., Zhou, Y., Chen, L., Wood, G.E., Almeida, N.F., Jr., Woo, L., Chen, Y., Paulsen, I.T., Eisen, J.A., Karp, P.D., Bovee, D., Sr., Chapman, P., Clendenning, J., Deatherage, G., Gillet, W., Grant, C., Kutyavin, T., Levy, R., Li, M.J., McClelland, E., Palmieri, A., Raymond, C., Rouse, G., Saenphimmachak, C., Wu, Z., Romero, P., Gordon, D., Zhang, S., Yoo, H., Tao, Y., Biddle, P., Jung, M., Krespan, W., Perry, M., Gordon-Kamm, B., Liao, L., Kim, S., Hendrick, C., Zhao, Z.Y., Dolan, M., Chumley, F., Tingey, S.V., Tomb, J.F., Gordon, M.P., Olson, M.V. and Nester, E.W. (2001) The genome of the natural genetic engineer Agrobacterium tumefaciens C58, *Science*, **294**, 2317-2323.

190. Wootton, J.C. and Federhen, S. (1993) Statistics of Local Complexity in Amino-Acid-Sequences and Sequence Databases, *Computers & Chemistry*, **17**, 149-163.

191. Yang, J., Gu, Z. and Li, W.H. (2003) Rate of protein evolution versus fitness effect of gene deletion, *Mol Biol Evol*, **20**, 772-774.

192. Yin, Y. and Fischer, D. (2006) On the origin of microbial ORFans: quantifying the strength of the evidence for viral lateral transfer, *BMC Evol Biol*, **6**, 63.

193. Yooseph, S., Sutton, G., Rusch, D.B., Halpern, A.L., Williamson, S.J., Remington, K., Eisen, J.A., Heidelberg, K.B., Manning, G., Li, W., Jaroszewski, L., Cieplak, P., Miller, C.S., Li, H., Mashiyama, S.T., Joachimiak, M.P., van Belle, C., Chandonia, J.M., Soergel, D.A., Zhai, Y., Natarajan, K., Lee, S., Raphael, B.J., Bafna, V., Friedman, R., Brenner, S.E., Godzik, A., Eisenberg, D., Dixon, J.E., Taylor, S.S., Strausberg, R.L., Frazier, M. and Venter, J.C. (2007) The Sorcerer II Global Ocean Sampling Expedition: Expanding the Universe of Protein Families, *PLoS biology*, **5**, e16.

194. Zheng, Y., Anton, B.P., Roberts, R.J. and Kasif, S. (2005) Phylogenetic detection of conserved gene clusters in microbial genomes, *BMC Bioinformatics*, **6**, 243.