Annotation of Bacterial and Archaeal Genomes: Improving Accuracy and Consistency

Ross Overbeek,[†] Daniela Bartels,[‡,§] Veronika Vonstein,[†] and Folker Meyer*[,‡,§]

Fellowship for Interpretation of Genomes, Burr Ridge, Illinois 60527, The Computation Institute, University of Chicago, Chicago, Illinois 60637, and Mathematics and Computer Science, Argonne National Laboratory, Argonne, Illinois 60439

Received June 11, 2003

Contents

* Phone: (+1) (630) 252-3261. E-mail: folker@mcs.anl.gov. Fax: (+1) (630) 252-5986. Web: http://www.mcs.anl.gov/~folker.
[†] Fellowship for Interpretation of Genomes.
[‡] University of Chicago.
[§] Argonne National Laboratory.

1. Introduction

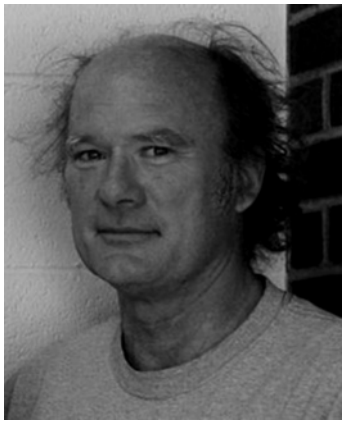
The speed with which sequencing technology has advanced and will continue to advance plays a central role in the topic of this review. Figure 1 summarizes the situation and has been presented in one form or another by many authors.

This rate of increase and its impact on biology are analogous to the impact of chip development technology on computing. In the case of computing, Moore's law is widely thought of as symbolizing the impact of a technology driving a scientific discipline. It is important to consider the cascading scientific opportunities produced by what appeared to be just an engineering triumph. If indeed there is a lesson to be learned from the developments in computing, it is that the outcome of such rapid technological advances can have profound and often unpredictable consequences for science. The rapid advances in basic infrastructure support and produce numerous research opportunities.

If we pursue the analogy between advances in sequencing technology and advances in microchip production, the topic of this review, annotation of genomes, corresponds very roughly to packaging the underlying electronics in forms that support mass consumption. Such packaging was of far less significance than the underlying driving technology, but it did play an essential role. The innovations that arose allowed the advances in microchips to have an almost immediate impact on both commercial and scientific applications. Similarly, the quality of annotations that we make for the thousands of genomes that will soon exist will determine their utility.

In this review, we focus on annotation of prokaryotic genomes. We do so for a number of reasons. The most basic

10.1021/cr068308h CCC: $65.00 © 2007 American Chemical Society
Published on Web 07/21/2007



Ross Overbeek received a doctorate in computer science in 1972 from Penn State University. He taught at Northern Illinois University for 11 years (in mathematics and computer science). His research areas were computational logic and database systems. From about 1983–1998, he worked at Argonne National Laboratory (ANL), focusing on parallel computation and logic programming. While a senior scientist at ANL, he became convinced that the fun had gone out of high-performance computing. At a critical moment, he met Carl Woese, who convinced him that the most important science during the next decades would be done in biology and that it would be driven by comparative analysis based on a rapidly growing body of genomic sequence data. He collaborated with Woese and participated in the founding of the Ribosomal Database Project. He went on to participate in the analysis of *Methanococcus jannaschii* (the first archaeal genome). He was the lead architect of the PUMA and WIT systems at ANL before becoming a founder of Integrated Genomics (where he spent 1998–mid-2003). While at IG he participated in the sequencing and analysis of over 50 genomes and led the bioinformatics effort. The most significant product was ERGO, a system to support comparative analysis. In mid 2003, he left IG to become a founding fellow of FIG (the Fellowship for Interpretation of Genomes). His efforts at FIG have centered on building a new system for comparative analysis, the SEED, which will be open source and free for all. Since 2004, he has been co-PI of the National Microbial Pathogen Data Resource, a framework to support comparative analysis of pathogen genomes.

Daniela Bartels received her diploma (M.Sc. level) in computer science in 2002 and her diploma (M.Sc. level) in 2003 in biology both from Bielefeld University. In 2006, she received her Ph.D. in biology working in the area of bioinformatics. Her thesis project involved data analysis and software development in the field of bacterial genome sequencing. She began working with the bioinformatics group of Dr. Meyer in 1999. She joined Bielefeld University's Center for Biotechnology when it was founded in 2001 and worked in bioinformatics, software engineering, and data evaluation in the context of multiple genome and post-genome projects. She is currently working at Argonne National Laboratory and the University of Chicago in the group of Dr. Meyer.

Veronika Vonstein received her doctorate in molecular biology in (1987) from Humboldt Unviversity, Berlin, Germany. In 1989, she became interested in microbial whole-genome analysis, first building physical maps of different strains of *Corynebacterium glutamicum* via pulse field electrophoresis at the All-Union Research Institute for Genetics and Selection of Industrial Microorganisms, Moscow, Russia, and later (1996) leading the *Thermus flavus* sequencing project at Thermogen, Inc. in Chicago, IL. In 1997, she was co-founder and Vice President Operations of Integrated Genomics, Inc. (1998–2003), where her collaboration with Dr. R. Overbeek began. His WIT, WIT2, and later ERGO environments for comparative analysis were key to the analysis of the more than 50 microbial genomes (including bacteria, archaea, and fungi) that were sequenced by the company during this period. After parting from IG in 2003, she cofounded the non-profit Fellowship for Interpretation of Genomes, where she now focuses on the comparative analysis of public genomes in the new SEED environment, using the subsystem approach to annotation.

Folker Meyer received his M.Sc. (1996) and Ph.D. (2002) in computer science from Bielefeld University, Germany. He worked in the groups of Prof. Robert Giegerich and Prof. Alfred Pühler on problems at the intersection of computer science and biology. He went on to form his own group in bioinformatics and high-performance computing in the Center for Biotechnology at Bielefeld University. His team built a number of software systems to handle the analysis of data sets from high-throughput biology. Best known is the GenDB genome annotation system, which is widely used for the analysis of microbial genomes. Meyer is now a computational biologist at Argonne National Laboratory and a Senior Fellow in the Computation Institute at the University of Chicago.

is that we are now seeing the introduction of newly sequenced prokaryotic genomes at a rate of tens or even hundreds per month. This rapid increase in volume offers real opportunities for improving the quality (as counter-intuitive as this may seem). The presence of more genomes lays the foundation for comparative analysis, which is the key to high quality. This will also be true of eukaryotic genomes, but not for a few more years; in eukaryotes, problems relating to gene identification and the presence of numerous copies of paralogs (closely related genes resulting from duplications) introduce substantial complexities that can be largely avoided in the analysis of prokaryotes. Moreover,

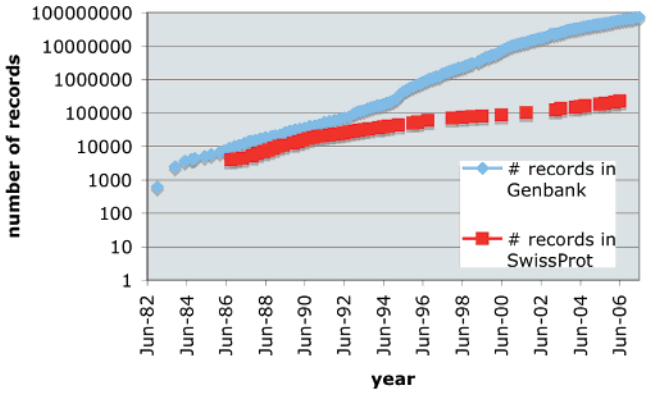


Figure 1. Growth of available genomes and SwissProt annotations. While the primary sequence repository (GenBank[1]) doubles in size every 18 months, high-quality annotations (we take SwissProt[2] as an example) cannot keep up with this growth. The graph compares the growth on a logarithmic scale.

the importance of addressing simpler problems before more complex ones seems obvious, and the scientific infrastructure offered by thousands of well-annotated prokaryotic genomes does arguably lay the foundation for study of the most fundamental issues in biology.

Before proceeding to a detailed discussion of what we mean by annotation (essentially the identification of gene locations for RNA-encoding and protein-encoding genes, along with a short description of the functions of the gene products), let us consider the overall picture of how annotations are now produced.

The process shown in Figure 2 is our view of how annotations are now done and how they will probably be done for the foreseeable future. This process has emerged as the most viable way to improve a situation containing substantial inconsistency and numerous errors. Our very broad-brush overview is as follows:

1. The initial annotations are in most cases provided by the sequencing centers. As we move to sequencing thousands of genomes per year, we believe that the percentage done by the sequencing centers will increase. The quality of these annotations is determined by the availability of well-curated protein families. The initial annotations are deposited in sequencing archives and may or may not be updated as time passes.
2. The production of well-curated protein families is now largely dominated by a few groups, most notably UniProt[3] (SwissProt and the Protein Identification Resource (PIR)), Kyoto Encyclopedia of Genes and Genomes[4] (KEGG), The Institute for Genome Research[5] (TIGR), and the Project to Annotate a 1000 Genomes[6] (P1K Project). These collections of protein families represent ongoing maintenance of archived sequences. It is through these families that most errors are corrected and new wet-lab results propagated.
3. Ultimately, the improvement of annotations results from wet-lab results and careful analyses provided by domain experts. These advances are reflected in the initial annotations and through ongoing curation of protein families. The archived annotations tend to go out of date, since regular updates are often not performed.

It is worth reflecting on the central curation role played by the teams maintaining protein families and their failure to impact the archives. There is no single, accepted collection of protein families. Rather, we see a rapid improvement based on open exchange that will gradually improve consistency of vocabulary and increase accuracy. Each of the main annotation/integration groups seeks accurate, comprehensive annotations in a controlled vocabulary. The fact that each of the main annotation efforts directly benefits from inspection of the others is producing a spontaneous convergence.

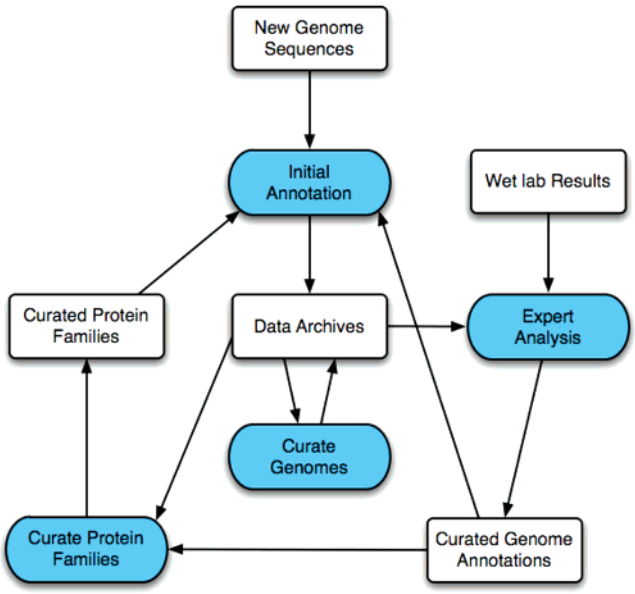


Figure 2. How annotations are done. This diagram is intended to convey the interactions between the different types of activities that make up the annotation process (blue). A key point is that maintenance and improvement of annotations originate in expert analysis and are reflected through protein family curation, since the "curate genomes" activity is seldom done.

1.1. What Is Meant by "Annotating a Genome"?

Abstractly, annotating a genome amounts to attaching information to support use of the genome. This includes an almost endless variety of types of analysis and attachment of interpretations. In our experience, it has proven useful to prioritize the analyses, and the following items provide at least a reasonable working notion of what is meant:

1. Genes are identified. This effort includes at least protein-encoding genes and some of the RNA-encoding genes (often just tRNAs and rRNAs).
2. The functions of genes are predicted.
3. Metabolic reconstructions are developed and tied to the specific genes.
4. Prophages, insertion sequences, and transposons are labeled.
5. Frameshifts and pseudogenes are predicted.
6. Regulatory sites and operons are identified as a step toward developing an inventory of regulons.

In practice, usually only the locations of genes and their predicted functions are provided by the initial annotation effort. Accordingly the first part of this review deals with the status of gene identification in prokaryotes, and the second part deals with the task of predicting the function to be associated with protein-coding genes.

2. Gene Prediction in Bacteria and Archaea

Once the sequence of a prokaryotic genome has been determined, the next step is the definition of the functional

elements coded by the sequence. This process is usually referred to as gene prediction or gene calling.

For high-quality genomes of prokaryotes, the quality of gene calls is usually very high; several good software solutions exist. Today more than 95−97% of the protein-coding genes can be correctly identified by state-of-the-art software.

For a number of years, the only type of genes that was considered was protein-coding genes (so-called coding sequences or CDSs). In recent years however, this notion has changed, and nowadays genes not coding for proteins (non-protein-coding genes) are included as well. Work on CDS prediction has a long history, starting with a number of publications in the early 1980s by Staden, Borodovsky, and others,[7−9] and is still an active field of research.[10−14] Work on the prediction of noncoding sequences has been picking up speed lately.

2.1. Prediction of Protein-Coding Genes

Unlike the prediction of eukaryotic genes, prediction of genes in prokaryotes is a rather well-understood process with rates of recognition well above 90% (see, e.g., McHardy et al.[15]).

2.1.1. Calling ORFs vs Gene Prediction

Sometimes gene prediction in prokaryotes is mislabeled as ORF finding. Prediction of ORFs and gene prediction are two distinctly different tasks, the first one being of trivial nature. Open reading frames (ORFs) are defined as stretches on the chromosome between a start codon initiating protein translation and a stop codon terminating it.

By simply extracting all subsequences that end at a stop codon whose length is divisible by three that have a valid start codon at their first position and include no other stop codon, the set of all ORFs can be extracted for a given genomic sequence. Extracting all ORFs of minimum length 90 nucleotides for *Escherichia coli* K12 (NCBI Taxonomy ID 83333.1) generates a list of 86 919 ORFs or protein-coding gene candidates. A good rule of thumb predicts approximately 1000 genes per million base pairs in a prokaryotic genome, leading us to assume about 4600 genes in this 4.63 million base pair genome; indeed most annotations show roughly 4600 genes.

While it is relatively simple to compute all ORFs for any given genome, the selection of the ORFs that code for proteins from that large set is much harder. Figure 3 shows the ORFs and protein-coding genes for a short sequence, highlighting the difference between ORF calling and CDS prediction.

From a computer science viewpoint, the problem of gene prediction in prokaryotes can be expressed as a binary classification problem. From the large set of ORFs, the smaller set of CDSs needs to be extracted.

2.1.2. Strategies for Gene Calling in Prokaryotes

Today numerous software solutions are available to automate prediction of genes for prokaryotes. The task that they fulfill is the determination of the subset of ORFs[1] that code for proteins *in vivo*.

While a manual procedure can easily be devised, it is error-prone and extremely tedious; after all, the ratio of ORFs to *real* proteins is >18:1.

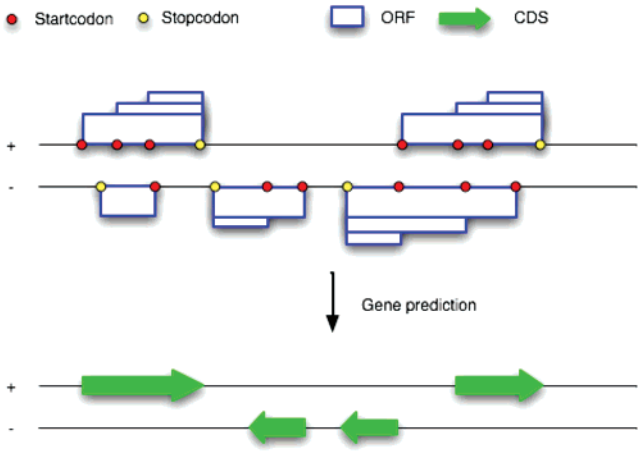


Figure 3. Gene prediction: from ORFs to coding sequences. From a large set of candidate open reading frames (ORFs), a smaller subset of coding sequences is selected.

Two distinct classes of automated gene prediction algorithms exist: intrinsic approaches that rely primarily on statistical properties of the coding sequences (see section 2.1.4) and extrinsic approaches (see section 2.1.5).

We note that a comprehensive comparative study of the prediction performance of most of the tools cannot be easily implemented. Only some tools are available in a format that allows large-scale testing and comparison of results. Therefore, a systematic study of the performance characteristics is next to impossible; we are drawing heavily on previous studies done by McHardy et al.[15] and our experience with reannotating hundreds of genomes.

2.1.3. Assessing Performance

Before we can study the gene calling systems and their underlying technologies in greater detail, a quick detour into statistics is required. We are treating the prediction of protein-encoding genes as a binary classification problem. The output of a classification algorithm (here the attempt to sort ORFs into coding and noncoding) can be split into four distinct classes: true positives (tp), true negatives (tn), false positives (fp), and false negatives (fn). By comparing the predicted genes with the annotated genes, one can determine the number of correct gene predictions (tp), the number of false gene predictions (fp), the number of genes that were not found (fn), and the number of correctly classified noncoding ORFs (tn).

Of course, classifying results into these four types requires a standard of truth. Since only very few genes have been actually confirmed in wet-lab experiments for a limited number of organisms, current best practice is to evaluate gene prediction algorithms by comparison to published annotations for relatively well-studied organisms. While this method has obvious shortcomings, we know of no alternative except for comparative analysis of genomes or large-scale wet-lab experiments.

We can analyze the performance of a given gene prediction method by measuring sensitivity (Sn = tp/(tp + fn), fraction of correctly identified genes) and specificity (Sp = tp/(tp + fp), fraction of correct predictions).

If these terms seem to be complicating the issue, imagine a classification strategy that achieves a perfect score for either specificity or sensitivity with no regard to the other dimension. Just calling all ORFs will yield a perfect sensitivity; however, the specificity will be terrible. On the other hand,

perfect specificity can be achieved by just calling one ORF with very solid sequence homology to support it. Clearly any attempt to evaluate gene callers must include both parameters.

2.1.4. (Mostly) Intrinsic Approaches

In the 1980s, researchers discovered[7−9] that coding sequences exhibit a number of properties that distinguish them from noncoding sequence and thus allow their automatic detection.

As early as 1984, R. Staden described how base composition, codon composition, and amino acid composition of a coding ORF can be used to distinguish it from noncoding ORFs.[8]

To achieve automatic detection of sequences showing the distinguishing properties of coding ORFs, techniques from machine learning are applied. As shown in Figure 4, a positive training set describing ORFs assumed to be coding is extracted from the genome. The training set consists of genes that are very likely to be coding, for example, long nonoverlapping ORFs or genes that, from BLAST comparison or their domain composition, have evidence to be real. While sometimes the genomic mean is used as the background or negative training set, some approaches use regions deemed to be not coding for the negative training set.

Typically the gene prediction programs will use Markov chains to derive models for coding and noncoding ORFs. A Markov chain is a network of *states* (the letters in the DNA alphabet) connected by *transitions*. Let us assume that in a very simple world the probability for the occurrence of a single nucleotide in coding and in noncoding sequences depends only on one previous letter. The Markov chain would contain *transitions* from every letter in the alphabet to every other letter.

From a (small) training set of known true ORFs, potentially genes that have been experimentally verified, we can derive the specific probabilities for the transitions for coding and for noncoding stretches of DNA. If coding and noncoding ORFs exhibit different properties, a simple approach to gene prediction is to check whether the probabilities for a given ORF under the coding model is higher than those under the noncoding model.

Computing the probability of GATC under a Markov chain can be translated into

$$P_{\text{chain}}(\text{G}|<\text{startsymbol}>) \times P_{\text{chain}}(\text{A}|\text{G}) \times P_{\text{chain}}(\text{T}|\text{A}) \times P_{\text{chain}}(\text{C}|\text{T})$$

If we performed this computation for both models and compared the results, we would have a basis for determining whether the ORF is more likely to code for a protein or not. Even in this simple view, however, picking the ORF with the correct start location is problematic.

Unfortunately, in the real world, models used in gene prediction need to be more complex. More factors than simply the previous nucleotide must be included in the statistical models. Examples of such properties are as follows: different GC content in coding genes; the GC content of the third position in a codon; nucleotide frequency; trinucleotide frequency; dinucleotide frequency and multicharacter frequencies; presence of ribosomal binding sites (RBS).[16] Still the model described above uses the basic approach taken by most of today's tools.

Figure 4. Positive and negative training set are derived from the genomic sequence. From the genome sequence, either a specific background model of noncoding regions is generated, or the complete genome is used as a background model.

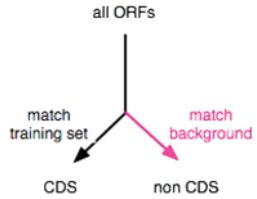


Figure 5. Using the derived model to classify ORFs into coding and noncoding. If a given ORF fits the training set better than the background set, it is assumed to be coding.

Initially a set of ORFs deemed to be coding, as well as the properties of that set, is extracted. A model is formed, and then every ORF in the genome is compared against the model (Figure 5). A background model is used for an additional comparison.

More details on the mechanics of hidden Markov models (HMMs) and their applications in bioinformatics are nicely described in the book by Durbin et al.[17]

Since the matching procedure can be performed irrespective of the data contained in the Markov model, some groups make their precomputed models available but offer no access to the training procedure. This in effect limits the user to the set of organisms that the sets were trained for. Using the models for other organisms is next to impossible and can generate an unpredictable outcome.

2.1.4.1. GLIMMER by Salzberg et al. One of the most often applied programs for gene prediction is GLIMMER, originally developed at TIGR and now supported and developed by S. Salzberg's group.

Originally published in 1998,[18] it has seen various improvements. The second major release, GLIMMER2,[19] utilizes variable word length when creating the models. More recently, a change in the early phase of a GLIMMER2 run has been introduced; by changing the code for the creation of the initial training set, a significant performance increase was achieved. Initially GLIMMER2 relied only on long, nonoverlapping ORFs.

A newer version, GLIMMER3,[20] has been made available recently, but there is as yet no third-party analysis of its performance characteristics. The authors of the software cite reduced numbers of false positive predictions and more accurate start predictions as the main improvements over the previous versions.

GLIMMER2 uses a variant of hidden Markov models, called interpolated context models, that allow for contexts of varying length to be used for each genome individually. However, the properties discussed for the HMMs still apply. In addition, RBSFinder[21] can be used to predict translation start sites, which can help to refine the start positions of the genes GLIMMER2 predicts.

One of the problems most often incurred with GLIMMER2 is its tendency to overpredict, resulting in a low specificity. Depending on the GC content of the respective genome, overprediction rates of 60% have been observed. While this can be tolerated and may in fact be viewed as beneficial if false positive gene calls have no adverse effect on a certain

project, it can nevertheless be quite frustrating in several situations. If false positive gene calls are viewed as problematic for a given application, postprocessing is required for many genomes. The sensitivity of GLIMMER is very high.

Fortunately, several solutions for postprocessing GLIMMER2 output have been published that address the issue of overprediction. The Reganor approach (see below) modifies both the training stage and the postprocessing, thus substantially enhancing the performance of GLIMMER2.

GLIMMER and all companion programs are open source and can be installed locally. While currently no webserver provides GLIMMER for on-line use, installation is simple, and the prediction software can be installed and run on any off-the-shelf LINUX PC in a few minutes.

GLIMMER is available from http://www.tigr.org/software/genefinding.shtml.

2.1.4.2. GeneMark by Borodovsky et al. One of the most often used gene-finding tools for prokaryotes is GeneMark from M. Borodovsky's group. This set of tool uses the hidden Markov model-based approaches and in fact offers several other tools to identify genes.

However, none of the tools is available for download or local use. While GeneMark/S[22] is a self-trained approach and can be used without a pretrained model, GeneMark.hmm for Prokaryotes[23] can be only be used with a predefined training set. The group provides 265 training sets, but many of these are for closely related organisms.

A comparison of GeneMark/S with other tools[13] revealed a high level of accuracy for predictions made by GeneMark/S. Since only a few pretrained models are available however, a more comprehensive comparison is impossible.

GeneMark is generally perceived as having a high specificity and high sensitivity. Because of the lack of availability for extensive testing, however, these statements should be viewed with caution.

The software can be used at http://exon.gatech.edu/GeneMark/.

2.1.4.3. EasyGene by Krogh et al. Another more recent development is the EasyGene[14,24] tool from A. Krogh's group in Denmark. As with the other tools, EasyGene is based on a hidden Markov model (HMM) trained for every new genome. In the training stage, external evidence in the form of SwissProt similarities is used; the resulting ORFs are scored, and their statistical significance is calculated. In order to determine the significance threshold, an artificial sequence is generated that has the same statistical properties as the input sequence, and the expected number of ORFs is compared to the actual sequence.

Unfortunately, EasyGene suffers from the same shortcoming as GeneMark.hmm. Only a very small number, here only 25, of pretrained models are available. While more predictions are available for already sequenced genomes, EasyGene as presented on the web is not usable for newly sequenced genomes.

A recent study showed EasyGene showed to have both high sensitivity and high specificity.[13]

In addition the group has recently published a study evaluating existing gene calls in published genomes as they are deposited in GenBank, revealing substantial shortcomings.[14]

The software can be used at http://www.cbs.dtu.dk/services/EasyGene/.

2.1.4.4. ZCurve by Guo et al. Another approach called ZCurve[25] uses linear discrimination functions to classify ORFs. Here the DNA sequence is transformed to a so-called zcurve, and predictions are made on the basis of this curve. The program has not been widely used, and we know of no independent evaluation of Zcurve.

The software is available in binary format for the Microsoft Windows platform.

2.1.4.5. GISMO by Krause et al. GISMO[13] uses a two-stage approach for gene identification. In the first stage, ORFs are screened against the PFAM database.[26] The set of ORFs with good protein domain matches is used as a training set for a support vector machine (SVM).[27] A set of ORFs overlapping the ORFs in the training set is used as a negative training set. The SVM can be used to solve nonlinear classification problems by separating a multidimensional feature space through a hyperplane. Here the feature space is defined by the codon usage vectors; each candidate ORF can be subsequently classified by the SVM using the models.

Krause et al. show that with this approach multiple sets of ORFs with different properties (genes with different codon usage or horizontally transferred genes) can be reliably detected. Since GISMO works well with reasonably small training sets, the software is well suited for small or fragmented genomes.

While no independent evaluation has been published, Krause et al. claim that GISMO achieves both high specificity and high sensitivity.

The software can be downloaded from http://www.cebitec.uni-bielefeld.de/groups/brf/software/gismo/.

2.1.5. Extrinsic Gene Callers

The second major class of tools for gene prediction in prokaryotes are the so-called extrinsic tools. These tools rely on existing knowledge stored in sequence databases. While a similarity search against a database of all known proteins appears to be a good way to identify protein-coding ORFs, this approach has at least two disadvantages: new proteins cannot be identified, and noise in databases will contaminate the genome annotation.

Beyond this straightforward approach, a number of more sophisticated approaches have been published over the past few years. Three programs can be considered the main representatives of this group: ORPHEUS, CRITICA, and Reganor.

Each of these tools uses intrinsic gene prediction technology in some form but also relies on sequence similarity searches. All three tools employ different concepts and show different performance properties.

2.1.5.1. ORPHEUS by Frishman et al. The oldest of the three tools, ORPHEUS[28] performs a search of the complete genome against a protein database and subsequently extends all resulting significant alignments thus creating a set of seed ORFs. From these seed ORFs, a model is trained for the codon frequencies. Seed ORFs and ORFs that according to the model can be classified as "true" ORFs constitute an output by ORPHEUS.

This tool, using just the codon frequencies as a simple model in conjunction with the algorithm for creation of the training set, provided a good solution in the late 1990s, when it was created. However, ORPHEUS is outperformed by GLIMMER for specificity and by CRITICA in terms of sensitivity.

The software is available upon request from Frishman et al.[28]

2.1.5.2. CRITICA by Badger and Olsen. While CRITICA[29] might seem like a variation of the approach taken by ORPHEUS, it provides a new approach with a fresh new insight. Relying on the fact that pressure for conservation is exerted on the protein level, Badger and Olsen based their tool on the detection of conserved stretches of DNA.

Initially a BLAST alignment is computed for subsequences of length 3000 for the genomic DNA. For each resulting alignment, the alignment is recomputed after translating both sequences into amino acid sequences. If the alignment on the protein level shows more sequence similarity than that on the DNA level, Badger and Olsen believe they have found indicators for "true" protein-coding ORFs. The next step of the program extends the candidate alignments into full-length ORFs.

An evaluation of CRITICA[12] shows that it produces very few false positive gene calls unlike many other gene prediction systems. However, the rate of false negatives is quite high. CRITICA is routinely used in genome projects. In comparison to *intrinsic* gene prediction programs, however, it requires significantly more CPU time; when considering the overall effort required to acquire and annotate a genome, this is usually not viewed as a serious problem.

Since CRITICA relies on BLAST database searches, it can only predict genes with already sequenced homologs. However the approach uses a somewhat simple model. Both factors result in very high specificity and sensitivity that on average is around 90%. For some purposes, where a high specificity is paramount, CRITICA is the tool of choice.

The software is freely available at http://www.ttaxus.com/files/critica105.tar.gz.

2.1.5.3. Reganor by McHardy et al. To unite the best of both worlds, A. McHardy et al. tried to combine the strengths of CRITICA (specificity) and GLIMMER (sensitivity). The main idea for the resulting Reganor[12] tool was to use CRITICA to create a training set for GLIMMER.

The approach is best described saying that the learning step for model creation is altered, using CRITICA prediction as a much larger training set. Thus, the false positive rate of GLIMMER was reduced at some loss of sensitivity. However, many users feel that the greater specificity outweighs the lost sensitivity.

Reganor has been successfully used for several dozen genome projects. Often, the resulting genes were manually annotated by teams of annotators. The software is available as part of the GenDB[30] genome annotation system and can be used online at https://www.cebitec.uni-bielefeld.de/groups/brf/software/reganor/. This approach is also implemented by Tech et al.[31]

2.2. Discussion of Tools for Predicting Protein-Coding Genes

Faced with the choice between possibly missed genes (false negatives or low sensitivity) and overcalling genes (false positives or low specificity), many groups in the past chose to lean toward overcalling. While still a suitable approach in many situations, certain downstream analysis steps are made substantially harder by overcalling genes.

Extrinsic (i.e., sequence similarity-based) tools have a tendency to miss genes that have not been deposited in the databases. However, using protein-domain-based analysis (e.g., GISMO) is likely to overcome this shortcoming.

In addition, since many groups would like an early start to the annotation process, there is pressure to compute gene predictions as soon as possible. The result is often a situation where multiple contigs still exist and genes are called for these, leading to fragmented genes.

Here the tools based solely on intrinsic methods expose another weakness: their training phase requires a large data set, and generally their performance is weaker with shorter contig lengths.

Another length-related phenomenon is the problem of correctly calling short genes. A number of systems simply exclude genes shorter than a given threshold. Most systems, however, still have difficulties in correctly identifying ORFs shorter than 300 bp or 100 amino acids. A study from 2001 clearly shows the shortcomings of existing gene calls with respect to short genes.[32]

2.2.1. Difficulties with the Correct Start Prediction

When one looks at multiple genomes using a tool for comparative genomics, it becomes clear that for many closely related genomes gene starts have been called differently by the various annotation teams or by their gene calling software. While a number of approaches for the correct identification of gene starts have been attempted,[21,33] most of the genomes in the public databases[34] still show incorrect start calls in addition to false and missing gene calls.

When viewing clusters[35] of related genes across multiple genomes, the shortcomings become apparent. The authors firmly believe that only thorough comparative analysis will lead to correct calls for the gene starts.

2.2.2. Problems Caused by Low Sequence Quality Genomes

All software systems as of today are not aware of the sequence quality of the genomes for which they are calling genes. Existing algorithms have significant shortcomings when presented with fragmented or low-quality genomic sequences. Because it is routine practice to chain contigs together using either runs of the N character, fragmented genomes cannot easily be detected by the gene calling software. A more sophisticated approach to contig chaining is adding a sequence fragment that contains stop codons in all six frames. As discussed above, genomes in multiple contigs posed difficulties to intrinsic gene callers since the size of the training set is often too small.

As it is becoming routine practice to sequence genomes using the new pyrosequencing-based technologies, genomes in several dozens or even hundreds of contigs are becoming more and more common. Using the procedures that have been devised for *finished* microbial genomes that are in one contig is more than likely to produce a large proportion of false positive gene calls. In fact the authors have witnessed this behavior for several low-quality data sets already. Here, a method based on comparative genomics could provide a much needed solution.

Unfortunately today, a large number of low-quality gene calls[14] for low-quality genomes are in the databases. Since in general the sequence quality of genomes deposited in GenBank is not known, there is no method for computing exact numbers.

The latest generation of gene calling procedures has provided the community with systems that reliably work for high- to medium-quality genomes; with use of support vector

machines, significant improvements were made in the areas of short genomes and alien sequences.

Foreseeable future improvements are in the area of gene starts and short genes both using improvements in machine learning technology and exploiting the power of comparative genomics.

2.2.3. Gene Calling for Metagenomics

To the best of our knowledge, current gene prediction tools cannot be applied to data sets coming out of random community genomics where DNA from an environment is sequenced without first cloning the DNA. While numerous data sets like the one described by Edwards et al.[36] exist, our ability to call genes in these sets is currently very limited. With existing gene calling technology rendered useless because of extremely short fragments of DNA, most groups rely on a simple extrinsic approach: they BLAST all fragments against the database of known proteins and derive possible open reading frames from the hits. This, however, cripples the ability to discover new proteins in these data sets. The approach described by Krause et al.[37] is a first attempt to overcome this problem by combining BLAST search and the analysis of synonymous and non-synonymous substitutions rates, a technique similar to the one applied by CRITICA.

2.3. Prediction of Non-Protein-Coding Genes and Features

The prediction of non-protein-coding features on the chromosome has been gaining importance over the past few of years. A number of standard tools have been used by many groups.

2.3.1. Prediction of Ribosomal RNA (rRNA) Genes

N. Larsen has developed a BLAST-based system that predicts the set of ribosomal RNA features on the chromosome. The software has not been published but is available from the author upon request (niels@genomics.dk).

2.3.2. Prediction of Transfer RNA (tRNA) Genes

A hidden Markov model-based detection program published[38] by S. Eddy's group still defines the state of the art for the prediction of transfer RNAs in prokaryotes. With high sensitivity and a very low ratio of false positives, tRNAs predicted with tRNAscan-SE are included in many genome annotations. Even though the software was initially described in 1997, it is still widely used. In addition to the source code for local installation, the authors provide a Web server to run the software.

The software is available and can be used online at http://lowelab.ucsc.edu/tRNAscan-SE/.

2.3.3. Prediction of Other Non-Coding RNA (ncRNA) Genes

RFAM[39] is a collection of noncoding RNA families and the corresponding sequence alignments and covariance models that enables searching for RNA genes. While the RFAM library also contains models that enable searching for tRNAs, the computational resources needed to run the associated tool *Infernal* render such searching almost impossible. While the data and software are available as open source, running it for complete genomes requires substantial resources. Accordingly no Web server offers screening of complete genomes against RFAM.

A more realistic scenario for running RFAM would be to run it as the last step of a feature prediction pipeline after all other features have been predicted. By limiting the searches to the areas not covered by other features, the computational costs can be limited.

2.4. Discussion of Gene Calling

The existing tools allow high-quality gene calls to be computed in a matter of minutes for most genomes. However, accurate prediction of starts and short genes still presents a challenge for most systems.

From the perspective of a consumer of gene predictions, the important points to keep in mind are the requirements of the downstream analysis and the sequence quality of the genomes. Methods that work well for high-quality sequences with one contig and very limited numbers of frameshifts generally do not work well for fragmented, low-coverage genomes.

Since many methods in bioinformatics rely on evaluation of sequences as they are deposited in the sequence databases, the existence of numerous false positives[32] and false negatives[12] in these databases poses significant problems to all users of bioinformatics tools.

We expect that a number of groups start projects to clean up existing gene calls. Initial work in this direction has already been done by a number of groups (for example, in Germany[12] and Denmark[14]).

3. Characterizing Function

The focus of this part of the review is how to characterize the function of identified genes accurately. That is, given a set of identified genes, we focus on the task of assigning an estimate of function to each gene. We think of the *function of a gene* as represented by a relatively short text string, although it is certainly equally reasonable to think of it as a complex set of assertions. What we are calling the function of a gene, in the case of a protein-encoding gene, is frequently called the *name of the gene product*, or just the *product name.*

In most cases, individuals annotating the genes within a newly sequenced genome think of themselves as attempting to leave accurate estimates of function or useful clues when precise estimates are not achievable. The emphasis is on accuracy. On the other side, those researchers annotating protein families tend to think of functions as Platonic abstractions; in this case, the annotation process amounts to connecting individual genes to one or more (in the case of multifunctional gene products) of these abstractions. The functional abstractions are grouped to represent abstractions of cellular machinery (e.g., pathways). Both perspectives address real needs, and as we progress toward establishing conventions for structuring product names, both aspects will need to be reconciled.

One basic view of how characterization of function is achieved is that we have a continuously progressing wet-lab effort producing reliable characterizations and that the role of genome annotation efforts is to accurately project these solid assignments to as broad a class of genomes as possible. This leads to a simple visual image: If one thinks of accurate characterizations as pebbles being dropped into a calm pond, then the ripples represent the impact moving

outward in the space of similar sequences. Points that are not extremely close to a dropped pebble end up absorbing waves from a number of sources, and the outcome is not clear. Integration of information (i.e., of the ripples) is complicated badly by the fact that controlled vocabularies have not yet been adopted (a topic discussed below).

Two aspects of the common view need to be noted. First, the impact of experimentally characterized genes is often limited to new genomes because groups annotating genomes often fail to go back and update the function assignments. In this regard, the groups annotating protein families play a critical role. It is not unusual to find an annotation on a primary sequence source failing to reflect wet-lab results published years before but accurately reflected in the protein family collections. Second, the bioinformatics aspect of annotation efforts is now beginning to play a more active role than simply projecting experimental results; predictions of merit are now being made regularly based solely on comparative analysis.

3.1. The Use of a Controlled Vocabulary and the Need for Consistency

The bulk of this review focuses on how one attempts to determine the function of a gene, not how to express that function. However, the issue of whether the functions of genes should be expressed in a controlled vocabulary and more specifically which vocabulary is a topic of growing interest and deserves some discussion. The need has been succinctly stated in the UniProt protein naming guidelines:[40]

> *Consistent nomenclature is indispensable for communication, literature searching and entry retrieval. Many species-specific communities have established gene nomenclature committees that try to assign consistent and, if possible, meaningful gene symbols. Other scientific communities have established protein nomenclatures for a set of proteins based on sequence similarity and/or function. But there is no established organization involved in the standardization of protein names, nor are there any efforts to establish naming rules that are valid across the largest spectrum of species possible.*

At least two aspects of establishing a "consistent nomenclature" should probably be thought of as distinct: using a controlled vocabulary for the gene functions (product names) and embedding the gene function within one or more hierarchies (or directed acyclic graphs, in cases in which strict hierarchies are viewed as too constraining).

Let us first consider just the issue of a controlled vocabulary for the actual gene function. In practice, there are two key factors limiting the rate of standardization: (1) Many of the important genomes have established organism-specific databases, and within these, the incentives for seeking and adopting a vocabulary common to all prokaryotes are limited. (2) Much of the existing vocabulary for gene functions is due to large annotation efforts focusing on the annotation of protein families.[3,4,6,41,42] These efforts have produced large numbers of annotations in a few, somewhat inconsistent vocabularies. Figure 6 illustrates the level of incompatibility for even the most commonly accepted gene functions.

It is important to note that this level of incompatibility poses minimal inconvenience for a trained biologist but

COG0187: Type IIA topoisomerase (DNA gyrase/topo II topoisomerase IV) B subunit
DNA gyrase subunit B (EC 5.99.1.3)
DNA gyrase subunit B (type II topoisomerase)
DNA gyrase subunit B GyrB
DNA gyrase subunit B type II topoisomerase
DNA gyrase subunit B, type II topoisomerase [EC:5.99.1.3] [KO:K02470]
DNA gyrase, B subunit
DNA gyrase, subunit B
DNA gyrase, subunit B (gyrB)
DNA gyrase, subunit B (type II topoisomerase)
gyrB
putative DNA gyrase, subunit B [EC:5.99.1.3] [KO:K02470]
50S ribosomal protein L33
COG0267: Ribosomal protein L33
50S ribosomal subunit protein L33
LSU ribosomal protein L33p
50S ribosomal protein L33 [KO:K02913]
50S ribosomal subunit protein L33 [KO:K02913]

Figure 6. The need for a controlled vocabulary. These product names all occur within the archives and protein families. In these two cases, a biologist would quickly grasp that the different names describe the same abstract function, but programs often fail.

makes it difficult (if not impossible) for programs to accurately predict when two gene functions represent the same abstract notion. This in turn substantially reduces the capability of automated annotation efforts. We feel that it is essential that consistent, controlled vocabularies be established for expression of gene function. These will be critical to achieving and maintaining accuracy as the number of annotated genomes grows exponentially. We believe that the perspective of maintenance of protein families is critical to this discussion for reasons that will become apparent below. For now, we wish to emphasize the following specific needs:

1. The single most critical need is to make it possible to easily determine whether two gene functions (product names) describe precisely the same abstract function. The exact wording of a gene function is not tremendously significant, since one can associate whatever clarifications or amplifications to the function one desires.

2. The next most important need is to give distinct functions distinct names. This is a far from obvious need, since it is common practice to specify unknown functions as *hypothetical protein.* However, it is extremely useful to be able to recognize that a set of proteins all share the same function, even if that function is unknown or imprecisely characterized.

3. It is important that sets of proteins sharing the same function name can all be easily reassigned new descriptions of function in response to newly published research. That is, it is essential that renaming the function associated with a protein family can be done easily.

It is, perhaps, worth noting that the development of PIR families, the construction of a vocabulary at SwissProt, and

the use of clusters of orthologous groups (COGs) were all significant steps in addressing these needs.

There has also been a rapidly growing interest in placing gene function within hierarchically structured ontologies. These grew out of early efforts to impose order on the collection of gene functions within single genomes.[43,44] At this point, the GeneOntology (GO) effort[45] has emerged within eukaryotic genome analysis as the framework in which a controlled vocabulary for the hierarchical ontologies is being developed. The situation within prokaryotic annotation is less clear. As mentioned above, large numbers of high-quality annotations have now been produced by teams annotating protein families. Each of these groups uses a vocabulary in which the precise product names often fail to identically match the existing GO vocabulary, and the GO vocabulary is far from complete for prokaryotic gene functions. It seems likely that at least the nomenclature of gene functions will be standardized in the next few years. Whether or not the result is based on GO is not yet clear. The intermediate position of developing mappings between distinct controlled vocabularies has begun. This effort involves reconciliation of a number of carefully maintained collections of protein families. This act of reconciliation is central to the effort to standardize nomenclature, and we expect the UniProt[3] effort to ultimately play a major role.

It is important that some comments be made about the annotation of genes encoding multifunctional proteins. At least three distinct notions need to be expressed: (1) The product has multiple distinct functions implemented by distinct domains. (2) The product has multiple distinct functions implemented by a single domain (e.g., an enzyme with broad specificity). (3) The product has one of a set of functions, but we do not know which.

Any effort to generate a controlled vocabulary for expressing the function of a gene must address each of these three distinct classes of assertions. To gain some insight into the complexities, consider the case of the 2-methylisocitrate dehydratase in *Salmonella typhimurium* LT2.[46] The following facts are relevant: (1) A four-step pathway exists that is capable of converting propionyl-CoA to succinate. (2) Three of the four steps are catalyzed by genes that occur in a tight cluster on the chromosome (which is conserved throughout numerous of the existing sequenced genomes). (3) The remaining enzymatic role, 2-methylisocitrate dehydratase (EC 4.2.1.99), converts 2-methyl-*cis*-aconitate to methylisocitrate. Aconitases, which are normally associated with the conversion of citrate to isocitrate in the tricarboxylic acid cycle (TCA), implement this role.

Hence, the function assigned to the genes encoding aconitases in organisms that have the four-step pathway must express the fact that they catalyze two distinct reactions (citrate to isocitrate and 2-methyl-*cis*-aconitate to methylisocitrate). However, in organisms without the propionyl-CoA to succinate pathway, essentially the same aconitases catalyze only the reaction from the TCA. That is, a normal aconitase will often have the potential of catalyzing EC 4.2.1.99, but in organisms lacking the other enzymes, it only actually catalyzes the conversion of citrate to isocitrate. A decision must be made about whether the gene function should express just the actual or also the potential enzymatic roles (see Figure 7).

In our judgment, the gene function should express the functional roles that the encoded protein is believed to actually implement, but good arguments can be made for either side of the issue. One strong argument for expressing the potential is that it simplifies automated projection of function assignments to new genomes. If only actual roles are to be specified, then a rule-based projection strategy will be essential to support automation.

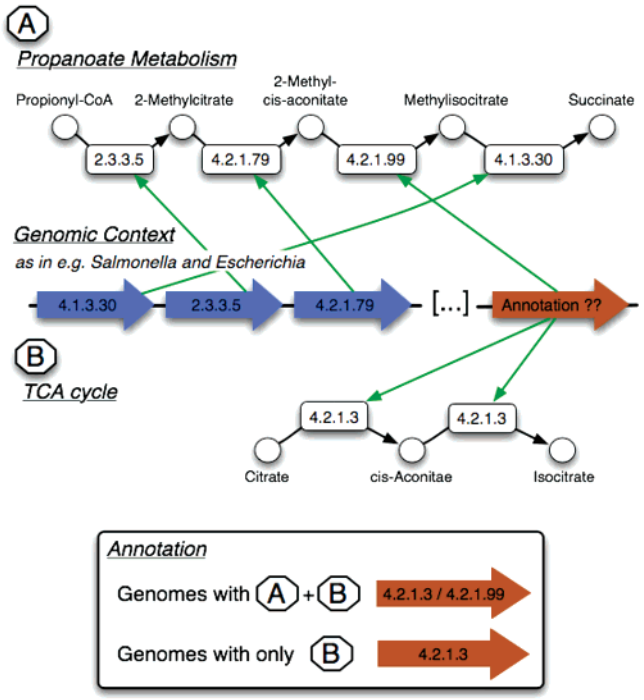


Figure 7. Do we annotate the actual or potential functions? Here, if A (the ability to convert propionyl-CoA to succinate) is present, then the function of the product of the red gene is multifunctional (implementing 4.2.1.3 and 4.2.1.99); if not, then the gene actually only implements 4.2.1.3 (which in itself catalyzes two distinct reactions).

3.2. Initial Annotations

The technology for producing and curating function assignments has progressed rapidly since 1995, when the first complete bacterial genome became available.[47] However, the single most significant component of annotation technology was and remains inferences based on sequence similarity. Two tools stand out as having played major roles in the use of similarity to infer homology (leading to conjectures of function): FASTA[48] and BLAST.[49] FASTA was developed first and was widely used. Both the WIT[50] and ERGO[41] systems used it almost exclusively. It provided what were felt to be somewhat more sensitive searches for global similarity (similarity across the entire length of the similar proteins). However, BLAST offered performance advantages and has been incrementally improved. It is now considered to be the standard tool for detection of sequence similarity. Given a protein sequence, the first step in seeking an appropriate functional assignment is to use BLAST to locate similar sequences in a large collection of already annotated sequences. There are a number of blast servers, but by far the one most commonly used is the one supported by NCBI.[51] It would be fair to say that the creation of BLAST and the provision of this blast service have been the foundation upon which most of the analysis of the first few hundred genomes has been based.

The basic annotation strategy using BLAST involved BLASTing each new sequence against previously annotated

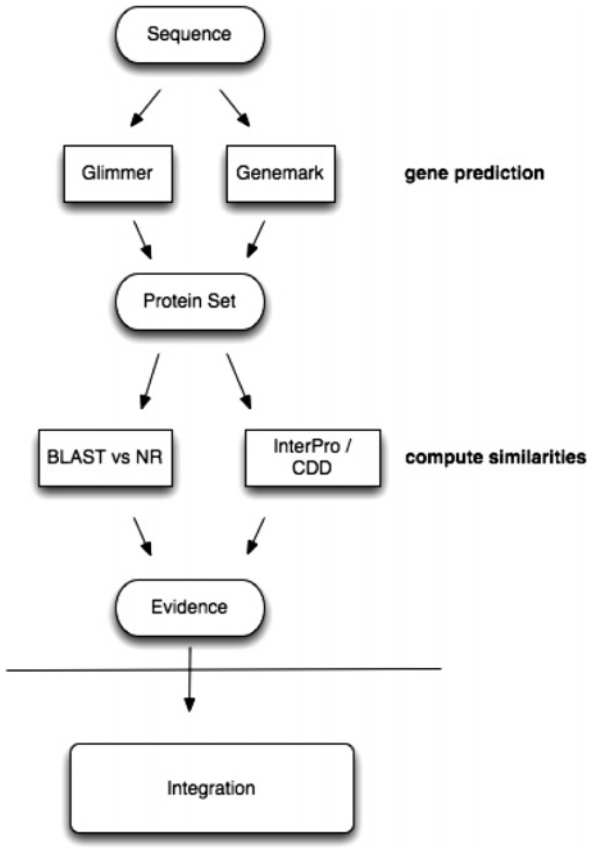


Figure 8. The majority of the genomes available in the public archives have been annotated using an approach very similar to this. Either GLIMMER[19,52] or GeneMark[53] is used to predict genes, a BLAST[49] similarity search is computed against NCBI's nonredundant protein database, and in addition a number of protein family or protein domain databases such as Pfam,[26] InterPro,[54] and the conserved domain database (CDD)[55] are searched. The resulting evidence is then integrated in the next step.

sequences, gathering the reported similarities, and then integrating the evidence to make a judgment. When early genomes were studied, the integration process involved a skilled biologist looking at the clues and pursuing them using a broader suite of tools. Some of these annotators were far more effective than others. In many cases, integration amounted to picking the most similar sequence and copying the annotation. As volumes of data grew exponentially, it became clear that there was a great deal to be gained by thinking about efficient ways to both automate the processes used by successful annotators and to address issues of scalability. The overall annotation task can reasonably be divided into two components: (1) a data acquisition component produced raw evidence by running a set of relevant, proven tools, and (2) an integration component attempted to replicate as much as possible of the processes used by the better human annotators.

The first task is relatively simple, but requires access to computational resources. As outlined in Figure 8, for most genomes, a number of tools are run against the set of proteins predicted for that genome. We believe that the most valuable source of evidence was and remains similarities against proteins in carefully annotated protein families.

The second task, the integration of the output of a set of tools, is often viewed from one of two perspectives. The first position notes that the rapid increase in volume will continue to be exponential and focuses on the development of a fully automated technology. The second approach notes the complexity of the judgments made by human experts and follows the goal of increasing the productivity of the human annotators. These are, in fact, not incompatible approaches. Clearly, the vast majority of annotations produced during the next 5 years (for thousands of genomes) will be done almost entirely automatically. Anything less than capturing and applying the types of judgments of skilled human annotators will simply produce a growing body of mistakes. In a work that we highly recommend, Koonin and Galperin offer many insights into what is wrong with the existing annotations and what is needed to address these errors[56,57] and offers interesting insights into problems resulting from incorrect annotations.

3.3. Specialized Tools To Support Assignment of Function

Table 1 lists a number of useful tools for the assignment of function. These compute specific properties that are often relevant to accurately characterizing gene function.

Most of these tools are quite accurate, at the expense of significant computational costs. While Web servers exist for some tools, most groups aiming at annotating a complete genome will need to locally provide the required computational resources.

Rey et al.[67] presented a detailed comparison of prediction performance.

3.4. Protein Families

The perspective of the annotator trying to annotate the genes in a newly sequenced genome is to consider "one gene/protein at a time". A major improvement in accuracy can be attained when annotators focus on sets of similar (and presumably homologous) sequences. The basic steps in investigating protein families may be summarized as follows:

- Form a set of sequences that are similar, and construct a multiple sequence alignment.
- Using the alignment, construct a phylogenetic tree that represents an estimate of the evolutionary relationships of the sequences.
- Label the leaves of the tree with the functions corresponding to each sequence.
- Carefully analyze and consider the points where function appears to change.
- If possible, characterize the motifs or structural features that characterize functional groupings.

Figure 9 illustrates what we mean by these steps. The underlying hypothesis is that function changes rarely so we expect to see a tree with a single function or that when a function does shift we expect to see *coherent subtrees* in the sense that the function does not simply flip back and forth. The original picture is often a tree in which function assignments are badly mixed. The process of slowly using the tree to bring the assignments into agreement with a hypothesized evolutionary history composed of a limited number of shifts in function is at the heart of curating these families.

Once we have formed a working model of what functions should be assigned to the sequences in the tree, the tree is used to support assigning function to new sequences. The process of annotation reduces to adding each new sequence to the multiple sequence alignment, using the alignment to

Table 1. Useful Tools for Detailed Functional Annotation[a]

tool	description	citation	URL	availability
PSORTb (v2.0)	localization prediction tool	58	http://www.psort.org/psortb/	GPL
TMHMM	predict transmembrane helices	59	http://www.cbs.dtu.dk/services/TMHMM/	binary available upon request
SignalP-(v3.0)	predict signal peptides	60	http://www.cbs.dtu.dk/services/SignalP/	binary available upon request
CELLO	SVM-based subcellular localization sites for Gram-negative bacteria	61	http://cello.life.nctu.edu.tw/	
TMPRED	homology-based (weight-matrix-based) prediction of membrane-spanning regions	62	http://www.ch.embnet.org/software/TMPRED_form.html	open source
tRNAscan	predict tRNAs	38,63	http://lowelab.ucsc.edu/tRNAscan-SE/	GPL
Phobius	combined prediction of signal peptides and transmembrane helices	64	http://phobius.cgb.ki.se	
PSLpred	PSIblast- and SVM-based method for subcellular localization of Gram-negative bacterial proteins	65	http://www.imtech.res.in/raghava/pslpred/	Web-Server, no local install
RFAM and infernal	find noncoding short RNAs	39,66	http://www.sanger.ac.uk/Software/Rfam/, http://infernal.janelia.org/	GPL

[a] In addition to sequence similarity based tools, a number of special-purpose tools have become widely used to assist with assignment of specific functions or protein localization in the cell.

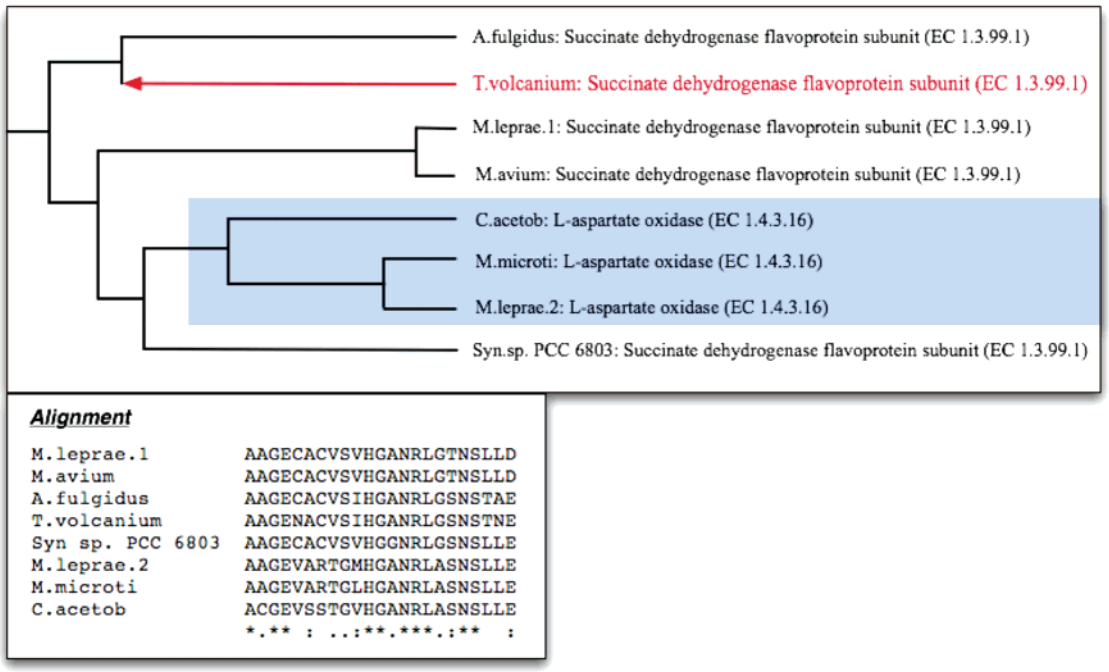


Figure 9. The relationship of alignments, trees, and gene function. This figure illustrates the notion that function determination can be viewed as a process based on analysis of where a new sequence belongs in a phylogenetic tree derived from an alignment.

insert the new sequence into the phylogenetic tree, and inferring the appropriate function based on the position of insertion.

We find it useful to think of a protein family as *a set of proteins that perform the same function and are globally homologous*. This amounts to partitioning the tree into coherent subtrees and labeling each of these subgroups as a distinct family. Associated with each family is a decision procedure that would take a new sequence in as input and decide whether it belonged to the family. Perhaps the most accurate such procedure would be the one mentioned in the preceding paragraph (determination of where the sequence belongs in the tree), but simpler procedures work just as well for many families. Indeed, the construction of hidden Markov models (HMMs) to implement the decision procedure is an extremely effective approach for many (perhaps a majority) of the protein families. A number of groups have made significant progress using HMMs,[68,69] most notably the way of Bateman et al.[26,70] or Haft et al.[71] Other important approaches that contain HMM technology or use alternative but quite similar technology are InterPro[54] and CDD.[55] Attaching decision procedures based on distinct technologies to each family will almost certainly be necessary as we progress toward more automation, while improving accuracy. We believe the common view that the rapid introduction of new genomes will inevitably lead to cascading errors is, we

believe, essentially wrong; as the details for each family are carefully developed, the additional data provided by the growing wealth of genomes support more accurate comparative analysis, which will rapidly reduce the overall error rate.

3.4.1. PIR: In the beginning...

The Protein Identification Resource (PIR)[71] grew out of the Atlas of Protein Sequence and Structure, a project initiated by M. Dayhoff in 1965. This effort pioneered many of the techniques for creating and maintaining protein families and continues today at the Georgetown University Medical Center. The terminology chosen by this group is as follows:

> *The primary level is the homeomorphic family, whose members are both homologous (evolved from a common ancestor) and homeomorphic (sharing full-length sequence similarity and a common domain architecture). At a lower level are the subfamilies which are clusters representing functional specialization and/or domain architecture variation within the family. Above the homeomorphic level there may be parent superfamilies that connect distantly related families and orphan proteins based on common domains.*[72]

This terminology does not agree with our notion of a family containing proteins with a common function. In the PIR perspective, the overall grouping is a superfamily, which may contain multiple families, each displaying a common domain structure. Subfamilies may be developed when detailed divisions based on function or minor variations in domain can be established. Families may or may not be subdivided. Hence, our notion of a set of proteins with a common function and global homology would amount to a lowest division (either family or subfamily) in which the PIR grouping included only proteins with a common function.

This group is one of the major participants in the UniProt effort[3] and, with SwissProt,[2] will undoubtedly play a leading role in moving toward a common controlled vocabulary and carefully curated set of protein families.

3.4.2. SwissProt

SwissProt[73] has become widely recognized as the central repository of high-quality annotations. At its center is a collection of extremely well annotated protein sequences, which are then linked to a wide variety of categories of data. The ProSite[74] collection of patterns that characterize function has historically played a serious role in forming and maintaining protein families. SwissProt has adhered to extremely high standards of manual curation. The actual number of sequences in the collection represents a small fraction of the overall number of available protein sequences, due to the effort required to maintain high quality (high-quality annotations, low redundancy, and integration have been the dominant themes of the collection[2]). The collection has had a major impact because of efforts to propagate this relatively high quality of annotations to related proteins. TrEMBL[2,3] constitutes a project to provide automatically derived annotations based on propagation of the SwissProt core to newly available protein sequences.

Perhaps the most notable developments relating to SwissProt are its participation in UniProt, a consortium that hopefully will lead to establishment of clean, consistent protein families and a controlled vocabulary,[3] and the development of HAMAP.[75]

The goals of the HAMAP are central to the topic of this review:

> *The HAMAP project aims to automatically annotate in UniProtKB/Swiss-Prot a significant percentage of proteins originating from bacterial and archaeal genome sequencing projects, with no decrease in quality. It is also used to annotate proteins encoded by complete plant and algal plastid genomes (e.g., chloroplasts, cyanelles, apicoplasts, non-photosynthetic plastids), and will be extended to mitochondrial genomes.*
>
> *Our automatic annotation methods, using a rule-based system, are only applied in the cases where they are able to produce the same quality as manual annotation would. This concerns two distinct subsets of proteins:*
>
> *1. proteins that have no significant similarity to any other microbial or nonmicrobial proteins (ORFans);*
>
> *2. proteins that are part of well-defined families or subfamilies.*

3.4.3. UniProt

UniProt was formed as a consortium merging efforts at SwissProt, PIR, and European Bioinformatics Institute.[3] The formation of this cooperative effort is rapidly leading to the creation of a controlled vocabulary, a distributed set of protein families reflecting that vocabulary, and an integration of the families with a broad variety of protein-related data. It is quite likely that this first step at integration will continue. At the very least, it seems likely to us that mappings between the protein families emerging from UniProt and those from other sources (e.g., those from KEGG, TIGR, and P1K) will be maintained. This would constitute a major step toward a common nomenclature.

3.4.4. COGs

In 1997, Tatusov, Koonin, and Lipman published a paper in *Science* releasing "720 clusters of orthologous groups (COGs)".[76] This was a seminal piece of work that attempted to group orthologs and to use these clusters of orthologs to develop and maintain a consistent set of function assignments. It is fair to say that this effort had a substantial impact on both cleaning up the huge inconsistencies in the public annotations and moving toward a framework that could support semi-automated annotations while maintaining accuracy.

3.4.5. TIGRFAMs

TIGRFAMs were introduced in 2001 as a set of protein families with associated HMMs "designed to support the automated functional identification of proteins by sequence homology".[70] The original collection included over 800 protein families divided into two classes. The first class, *equivalogs*, are those families in which all of the members have been assigned the same function (essentially the notion of protein familiy we suggested above). The second class represents sets of proteins that share related functions (that is, a common class functions in which the precise functions cannot yet be differentiated). There are over 2946 (release 6.0) TIGRFAMS in the distributed collection.

3.5. Annotation of Related Protein Families

Just as moving from annotating single genes/proteins at a time to annotating families of genes/proteins improves

accuracy, the annotation of sets of families simultaneously can be used to introduce more accuracy. The essential idea is that one forms a set of related functional roles and then annotates the entire protein families that implement these roles simultaneously. The term *subsystem* has come to refer to such a set of simultaneously annotated protein families (in refs 6 and 42, the term *subsystem* is defined as "a set of functional roles", but this amounts to the same notion). People often speak of the set as imposing a *context* or *neighborhood* that allows numerous consistency checks.

The advantages of annotating a set of protein families simultaneously, using these different forms of context to support the development of a consistent interpretation, have become widely recognized.[5,6,56,77,78]

The concept of annotation via subsystems is the basis of the Project to Annotate 1000 genomes.[6,42] In some sense, this paper did not represent a radical departure; indeed, annotation of protein families and proposals to construct expert systems for specific classes of proteins[75] certainly amounted to a widespread recognition that annotation of specific genes out of context was error prone. On the other hand, the view that one must annotate sets of families, that this should be done by experts who have studied specific cellular processes (rather than increasingly skilled annotators unfamiliar with the details of a research area), and that tools to support annotation of subsystems were critical did amount to a significant development. With the rapid sequencing of thousands of genomes, we believe that annotation of a growing body of subsystems by specialists who use tools to support near-automatic extensions of existing analyses will be the dominant strategy.

At this point, 35−40% of the genes in a relatively few genomes have been placed into a collection of subsystems. This number will grow to at least 50% for most genomes within a year or two.

3.6. Functional Coupling

While similarity-based reasoning has dominated most annotation efforts, there are a growing class of technologies can be used to reveal what is often called *functional coupling.* The term itself is vague (and, hence, somewhat irritating), but it does represent a notion of central significance. Operationally, we might think of two genes as "functionally coupled" if a skilled annotator would place the functional roles implemented by the two genes within the same subsystem. This, of course, sidesteps the issue of the criteria an annotator would use to place two functional roles within a single subsystem. In any event, the use of these non-similarity-based technologies plays a rapidly increasing role in contributing to the annotation of "hard cases", most commonly those characterized by numerous paralogs or in cases in which relevant genes show no similarity to known cases (that is, cases in which there are too many candidates because of similarity or in which there is no detectable similarity).

Figure 10 depicts some of the more common and useful techniques for detecting functional coupling; we will discuss the examples from that figure in detail.

We alluded to the basic idea that annotation is the science of propagating wet-lab characterizations, and we suggested that such a view needs to be modified in the presence of current advances. We suggest that the process of annotation be viewed as establishing sets of relatively reliable assertions of function supported by wet-lab results and consistency

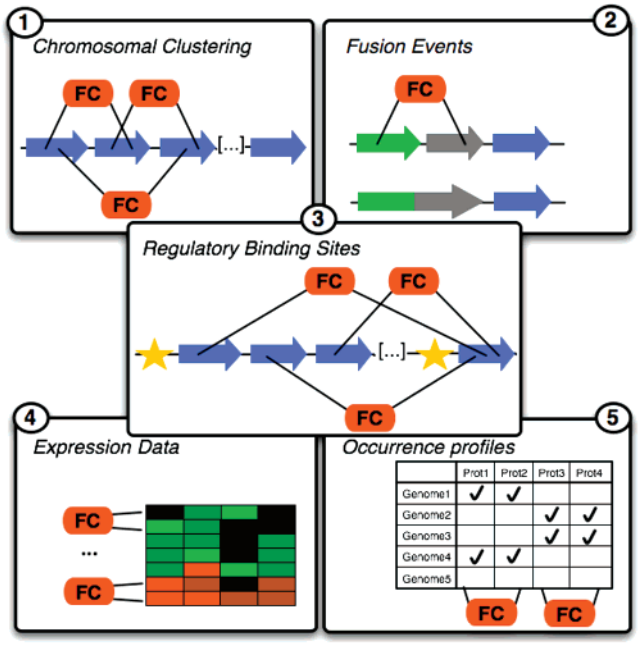


Figure 10. Clues to functional coupling. Functionally related genes can be revealed by using a variety of methods.

arguments and then propagating these results based on similarity and consistency arguments. The consistency arguments are based on these different approaches to estimating functional coupling. As the number and diversity of available genomes increase, the clues arising from a number of these techniques increase rapidly. In particular, the functional coupling estimates due to clustering of related genes in prokaryotic genomes and to clarification of regulatory sites are now leading to a growing number of wet-lab confirmations. There is a growing awareness that integration of disparate sources of functional coupling clues will allow more sophisticated and precise consistency arguments and interest in constructing such integrations is increasing.[79−82]

3.6.1. Functional Coupling Based on Chromosomal Clusters

In 1998, two independent efforts reported on how one might exploit the fact that functionally related genes tend to cluster on the chromosome.[83,84]One of these efforts reported an estimate that approximately 50% of the genes in a typical prokaryotic genome occur in close proximity to functionally related genes. This is easily verified: pick any cellular subsystem for which the genes are known and look at the distribution on the chromosome of those genes. The degree of clustering does depend on specific genomes (cyanobacteria tend to have approximately 20−25% of the genes clustered, while most other prokaryotes appear to have 50−60% clustered). The clustering seems to occur for both genes in central pathways and those in pathways of secondary metabolism (this can easily be checked by looking at instances of known pathways, complexes, or non-metabolic subsystems).

The context-based analysis introduced by the presence of clusters of functionally related genes has proven extremely significant.[85−88] It has supported efforts to construct families with functions that could be predicted with confidence, which are then used to address ambiguous cases. The power of this technique has increased with the number of available genomes. As we move into an era of thousands of diverse

genomes, we expect the use of chromosomal clusters to play a central role in clarification of function and the construction of protein families in which each member plays the same function.

3.6.2. Functional Coupling Based on Detection of Fusion Events

The prediction of functional coupling based on detection of fusion events is often referred to as the Rosetta Stone method. The basic idea behind the technique[79,89] is to detect instances in which two genes sometimes appear as distinct genes and sometimes as a single fused gene. Detecting such *fusions* provides extremely strong evidence that the functional roles implemented by the two distinct genes (or by the fused gene) are functionally related. Just as with chromosomal clustering, the value of this technique increases with the number of genomes. As the number of completely sequenced genomes continues to grow rapidly, the number of detected fusions increases.

3.6.3. Functional Coupling Based on Regulatory Sites

Over the past 4−5 years, it has become possible to use the analysis of upstream regulatory sites to support accurate characterization of regulons (and, hence, to support characterization of the genes that make up each regulon). Much of this work was pioneered by the team of Gelfand and Mironov,[90] but a number of other groups have also been quite successful.[91] These efforts are based on careful, case-by-case analysis, and we are not yet in a position to comprehensively characterize regulatory sites in genomes. However, it is now routine to use analysis based on detection of regulatory sites to develop an unambiguous function for a family of proteins.

3.6.4. Functional Coupling Based on Analysis of Expression Data

The growing body of expression data (largely in the form of microarray experiments) offers an obvious mechanism for characterization of regulons. A growing body of work centers on the extraction of interaction networks from a collection of such experiments. In our view, the characterization of regulons represents the next step in advancing prokaryotic annotations. The expression data, as well as the exposure of regulatory sites, will be integrated with the subsystem annotations to develop estimates of regulons as sets of subsystems.[92]

3.6.5. Functional Coupling Based on Occurrence Profiles

Suppose that we have m sequenced genomes and n protein families. Then, we can construct an $m \times n$ matrix in which each entry contains 0 or 1 depending on whether the corresponding genome includes at least one gene that encodes a protein from the given family. In this case, each of the m rows (one per genome) would encode a vector indicating which protein families were present, and each column (one per protein family) would constitute an occurrence profile. Two columns with similar profiles correspond to proteins that might play related roles. It is certainly not obvious that they must (since, for example, all universal proteins will have completely identical occurrence profiles), but when the profile correlates closely with a recognizable phenotype (e.g., the entries corresponding to photosynthetic organisms all contain 1, and the rest contain 0), the technique can be remarkably predictive. As the number of complete genomes grows, the utility of this technique continues to improve.

3.6.6. Functional Coupling Based on Protein–Protein Interaction Data

Protein−protein interaction data represents another technology that can be used to expose functional coupling. At this stage, it has (at least for prokaryotes) produced a fairly limited amount of data.[79,93,94] Moreover, the data sets that are available tend to be relatively small, and the data is often noisy.

3.7. Expert Curation

Until recently, expert curation was basically a wet-lab effort led by individuals with decades of experience in understanding specific domains. Improvements in bioinformatics tools and the advances described as "context-based" analysis have led to a number of efforts in which acknowledged domain experts made sustained efforts to clean up annotations relating to their area of expertise. The results are, in our opinion, stunning.

3.8. Why Annotations Will Rapidly Improve

The overview of the annotation process provided in Figure 2 depicts three interacting components: initial annotations, integration/construction of gene families, and expert curation. The annotation of previously published genomes (as clarification of the genes advances) is largely handled by two types of efforts: the literature summarizes the impact of wet-lab advances, and groups developing protein families attempt to project advances and develop an integrated view of function. We expect that the availability of large amounts of genomic data, as well as the growing accuracy of conjectures feeding the wet-lab efforts, will dramatically increase the productivity of wet labs. The existing annotations of complete prokaryotic genomes constitute a model in which a relatively small percentage of the annotations are supported by direct wet-lab data and the vast majority (necessarily) are supported by different forms of consistency arguments. As comparative analysis clarifies these consistency issues, our ability to focus wet-lab efforts to resolve the weakest portions of the model will improve. Similarly, as the model improves and the consistency dependencies come into focus, our ability to accurately project conjectures onto thousands of new genomes will improve dramatically.

4. Summary

Thousands of bacterial and archaeal genomes of widely varying quality will be made available in public archives during the next 5 years. The value of this data will directly depend on the quality of the annotations. There is every reason to believe that the overall quality of annotations will improve rapidly because of a number of factors:

1. The body of reliable annotations is expanding rapidly. The core of wet-lab confirmations is being supplemented by annotations supported by context-based analysis, and the consistency arguments supporting these annotations are gradually removing numerous ambiguities. Tools to support domain experts trying to rationalize specific subsystems or fragments of metabolism across the entire set of genomes are improving.

2. The core of reliable annotations is being used to develop collections of protein families of rapidly improving quality and coverage.

3. Controlled vocabularies are emerging from the major collections of protein families. It is reasonable to expect that the distinct vocabularies will rapidly converge; in cases in which differences persist, mappings between the vocabularies will be maintained.

4. Our ability to accurately and automatically annotate new genomes will steadily improve largely as a result of improvements in the protein families.

Numerous issues relating to special classes of proteins (e.g., transposable elements, transporters, regulatory proteins, and restriction enzymes) will be addressed by customized rules. These will be important, but the overall quality will be determined by the protein families. Substantial efforts will be devoted to pseudogenes and frameshifts, especially in the presence of thousands of genomes with low-quality sequence, but these should be viewed simply as part of the opportunity to actively compare and analyze thousands of distinct genomes.

We expect that many of the architectural details and diversity present in prokaryotic genomes will be cast in dramatic detail over the next 5 years. These dramatic advances in the basic infrastructure will lay the foundation for advancing our understanding of unicellular life.

5. Acknowledgment

We thank Gail Pieper for her critical reading of the manuscript.

6. References

(1) (1) Benson, D. A.; Karsch-Mizrachi, I.; Lipman, D. J.; Ostell, J.; Wheeler, D. L. *Nucleic Acids Res.* **2007**, *35*, D21.
(2) 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.; Schneider, M. *Nucleic Acids Res.* **2003**, *31*, 365.
(3) Wu, C. H.; Apweiler, R.; Bairoch, A.; Natale, D. A.; Barker, W. C.; Boeckmann, B.; Ferro, S.; Gasteiger, E.; Huang, H.; Lopez, R.; Magrane, M.; Martin, M. J.; Mazumder, R.; O'Donovan, C.; Redaschi, N.; Suzek, B. *Nucleic Acids Res.* **2006**, *34*, D187.
(4) Kanehisa, M. *Novartis Found. Symp* **2002**, *247*, 91−101; discussion 101−103, 119−128, 244.
(5) Selengut, J. D.; Haft, D. H.; Davidsen, T.; Ganapathy, A.; Gwinn-Giglio, M.; Nelson, W. C.; Richter, A. R.; White, O. *Nucleic Acids Res.* **2007**, *35*, D260.
(6) Overbeek, R.; Begley, T.; Butler, R. M.; Choudhuri, J. V.; Diaz, N.; Chuang, H.-Y.; Cohoon, M.; de Crécy-Lagard, V.; Disz, T.; Edwards, R.; Fonstein, M.; Frank, E. D.; Gerdes, S.; Glass, E. M.; Goesmann, A.; Krause, L.; Linke, B.; McHardy, A. C.; Meyer, F.; Hanson, A.; Iwata-Reuyl, D.; Jensen, R.; Jamshidi, N.; Kubal, M.; Larsen, N.; Neuweger, H.; Rückert, C.; Olsen, G. J.; Olson, R.; Osterman, A.; Portnoy, V.; Pusch, G. D.; Rodionov, D. A.; Steiner, J.; Stevens, R.; Thiele, I.; Vassieva, O.; Ye, Y.; Zagnitko, O.; Vonstein, V. *Nucleic Acids Res.* **2005**, *33* (17), 5691.
(7) Fickett, J. W. *Nucleic Acids Res.* **1981**, *10*, 5305.
(8) Staden, R. *Nucleic Acids Res.* **1984**, *12*, 551.
(9) Gribskov, M.; Devereux, J.; Burgess, R. R. *Nucleic Acids Res.* **1984**, *12*, 539.
(10) Guo, F. B.; Zhang, C. T. *BMC Bioinf.* **2006**, *7*, 9.
(11) Ou, H. Y.; Guo, F. B.; Zhang, C. T. *Int. J. Biochem. Cell Biol.* **2004**, *36*, 535.
(12) Linke, B.; McHardy, A. C.; Neuweger, H.; Krause, L.; Meyer, F. *Appl. Bioinf.* **2006**, *5*, 193.
(13) Krause, L.; McHardy, A. C.; Nattkemper, T. W.; Pühler, A.; Stoye, J.; Meyer, F. *Nucleic Acids Res.* **2007**, *35*, 540.
(14) Nielsen, P.; Krogh, A. *Bioinformatics* **2005**, *21*, 4322.
(15) McHardy, A. C.; Goesmann, A.; Puhler, A.; Meyer, F. *Bioinformatics* **2004**, *20*, 1622.
(16) Shine, J.; Dalgarno, L. *Proc. Natl. Acad. Sci. U.S.A.* **1974**, *71*, 1342−1346.
(17) Durbin, R.; Eddy, S. R.; Krogh, A.; Mitchison, G. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*; Cambridge University Press: Cambridge, 1999.
(18) Salzberg, S. L.; Delcher, A. L.; Kasif, S.; White, O. *Nucleic Acids Res.* **1998**, *26*, 544.
(19) Delcher, A. L.; Harmon, D.; Kasif, S.; White, O.; Salzberg, S. L. *Nucleic Acids Res.* **1999**, *27*, 4636.
(20) Delcher, A. L.; Bratke, K. A.; Powers, E. C.; Salzberg, S. L. *Bioinformatics* **2007**, *23*, 673.
(21) Suzek, B. E.; Ermolaeva, M. D.; Schreiber, M.; Salzberg, S. L. *Bioinformatics* **2001**, *17*, 1123.
(22) Besemer, J.; Lomsadze, A.; Borodovsky, M. *Nucleic Acids Res.* **2001**, *29*, 2607.
(23) Lukashin, A. V.; Borodovsky, M. *Nucleic Acids Res.* **1998**, *26*, 1107.
(24) Larsen, T. S.; Krogh, A. *BMC Bioinf.* **2003**, *4*, 21.
(25) Guo, F. B.; Ou, H. Y.; Zhang, C. T. *Nucleic Acids Res.* **2003**, *31*, 1780.
(26) 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.; Eddy, S. R. *Nucleic Acids Res.* **2004**, *32*, D138.
(27) Vapnik, V. N. *The Nature of Statistical Learning Theory*; Springer: Berlin, 1995.
(28) Frishman, D.; Mironov, A.; Mewes, H. W.; Gelfand, M. *Nucleic Acids Res.* **1998**, *26*, 2941.
(29) Badger, J. H.; Olsen, G. J. *Mol. Biol. Evol.* **1999**, *16*, 512.
(30) Meyer, F.; Goesmann, A.; McHardy, A. C.; Bartels, D.; Bekel, T.; Clausen, J.; Kalinowski, J.; Linke, B.; Rupp, O.; Giegerich, R.; Puhler, A. *Nucleic Acids Res.* **2003**, *31*, 2187.
(31) Tech, M.; Merkl, R. *In Silico Biol.* **2003**, *3*, 441.
(32) Skovgaard; al, e. *Trends Genet.* **2001**, *17*.
(33) Tech, M.; Pfeifer, N.; Morgenstern, B.; Meinicke, P. *Bioinformatics* **2005**, *21*, 3568.
(34) Pruitt, K. D.; Tatusova, T.; Maglott, D. R. *Nucleic Acids Res.* **2007**, *35*, D61.
(35) Overbeek, R.; Fonstein, M.; D'Souza, M.; Pusch, G. D.; Maltsev, N. *In Silico Biol.* **1998**, *1*, 93.
(36) Edwards, R. A.; Rodriguez-Brito, B.; Wegley, L.; Haynes, M.; Breitbart, M.; Peterson, D. M.; Saar, M. O.; Alexander, S.; Alexander, E. C., Jr.; Rohwer, F. *BMC Genomics* **2006**, *7*, 57.
(37) Krause, L.; Diaz, N. N.; Bartels, D.; Edwards, R. A.; Puhler, A.; Rohwer, F.; Meyer, F.; Stoye, J. *Bioinformatics* **2006**, *22*, e281.
(38) Lowe, T. M.; Eddy, S. R. *Nucleic Acids Res.* **1997**, *25*, 955.
(39) Griffiths-Jones, S.; Moxon, S.; Marshall, M.; Khanna, A.; Eddy, S. R.; Bateman, A. *Nucleic Acids Res.* **2005**, *33*, D121.
(40) SwissProt, http://www.expasy.org/sprot/.
(41) Overbeek, R.; Larsen, N.; Walunas, T.; D'Souza, M.; Pusch, G.; Selkov, E., Jr.; Liolios, K.; Joukov, V.; Kaznadzey, D.; Anderson, I.; Bhattacharyya, A.; Burd, H.; Gardner, W.; Hanke, P.; Kapatral, V.; Mikhailova, N.; Vasieva, O.; Osterman, A.; Vonstein, V.; Fonstein, M.; Ivanova, N.; Kyrpides, N. *Nucleic Acids Res.* **2003**, *31*, 164.
(42) The Institute for Genomic Research (TIGR) Comprehensive Microbial Resource, http://cmr.tigr.org.
(43) Riley, M. *Microbiol. Rev.* **1993**, *57*, 862.
(44) Overbeek, R.; Larsen, N.; Smith, W.; Maltsev, N.; Selkov, E. *Gene* **1997**, *191*, GC1.
(45) Harris, M. A.; Clark, J.; Ireland, A.; Lomax, J.; Ashburner, M.; Foulger, R.; Eilbeck, K.; Lewis, S.; Marshall, B.; Mungall, C.; Richter, J.; Rubin, G. M.; Blake, J. A.; Bult, C.; Dolan, M.; Drabkin, H.; Eppig, J. T.; Hill, D. P.; Ni, L.; Ringwald, M.; Balakrishnan, R.; Cherry, J. M.; Christie, K. R.; Costanzo, M. C.; Dwight, S. S.; Engel, S.; Fisk, D. G.; Hirschman, J. E.; Hong, E. L.; Nash, R. S.; Sethuraman, A.; Theesfeld, C. L.; Botstein, D.; Dolinski, K.; Feierbach, B.; Berardini, T.; Mundodi, S.; Rhee, S. Y.; Apweiler, R.; Barrell, D.; Camon, E.; Dimmer, E.; Lee, V.; Chisholm, R.; Gaudet, P.; Kibbe, W.; Kishore, R.; Schwarz, E. M.; Sternberg, P.; Gwinn, M.; Hannick, L.; Wortman, J.; Berriman, M.; Wood, V.; de la Cruz, N.; Tonellato, P.; Jaiswal, P.; Seigfried, T.; White, R. *Nucleic Acids Res.* **2004**, *32*, D258.
(46) Horswill, A. R.; Escalante-Semerena, J. C. *Biochemistry* **2001**, *40*, 4703.
(47) Fleischmann, R. D.; Adams, M. D.; White, O.; Clayton, R. A.; Kirkness, E. F.; Kerlavage, A. R.; Bult, C. J.; Tomb, J. F.; Dougherty, B. A.; Merrick, J. M.; et al. *Science* **1995**, *269*, 496.
(48) Pearson, W. R. *Methods Mol. Biol.* **1994**, *25*, 365.
(49) Altschul, S. F.; Madden, T. L.; Schaffer, A. A.; Zhang, J.; Zhang, Z.; Miller, W.; Lipman, D. J. *Nucleic Acids Res.* **1997**, *25*, 3389.
(50) Overbeek, R.; Larsen, N.; Pusch, G. D.; D'Souza, M.; Selkov, E., Jr.; Kyrpides, N.; Fonstein, M.; Maltsev, N.; Selkov, E. *Nucleic Acids Res.* **2000**, *28*, 123.
(51) National Center for Biotechnology Information (NCBI) BLAST, http://www.ncbi.nlm.nih.gov/BLAST/.
(52) Delcher, A. L.; Bratke, K. A.; Powers, E. C.; Salzberg, S. L. *Bioinformatics* **2007**.
(53) Besemer, J.; Borodovsky, M. *Nucleic Acids Res.* **2005**, *33*, W451.

(54) Mulder, N. J.; Apweiler, R.; Attwood, T. K.; Bairoch, A.; Bateman, A.; Binns, D.; Bork, P.; Buillard, V.; Cerutti, L.; Copley, R.; Courcelle, E.; Das, U.; Daugherty, L.; Dibley, M.; Finn, R.; Fleischmann, W.; Gough, J.; Haft, D.; Hulo, N.; Hunter, S.; Kahn, D.; Kanapin, A.; Kejariwal, A.; Labarga, A.; Langendijk-Genevaux, P. S.; Lonsdale, D.; Lopez, R.; Letunic, I.; Madera, M.; Maslen, J.; McAnulla, C.; McDowall, J.; Mistry, J.; Mitchell, A.; Nikolskaya, A. N.; Orchard, S.; Orengo, C.; Petryszak, R.; Selengut, J. D.; Sigrist, C. J.; Thomas, P. D.; Valentin, F.; Wilson, D.; Wu, C. H.; Yeats, C. *Nucleic Acids Res.* **2007**, *35*, D224.
(55) Marchler-Bauer, A.; Anderson, J. B.; Derbyshire, M. K.; DeWeese-Scott, C.; Gonzales, N. R.; Gwadz, M.; Hao, L.; He, S.; Hurwitz, D. I.; Jackson, J. D.; Ke, Z.; Krylov, D.; Lanczycki, C. J.; Liebert, C. A.; Liu, C.; Lu, F.; Lu, S.; Marchler, G. H.; Mullokandov, M.; Song, J. S.; Thanki, N.; Yamashita, R. A.; Yin, J. J.; Zhang, D.; Bryant, S. H. *Nucleic Acids Res.* **2007**, *35*, D237.
(56) Galperin, M. Y.; Koonin, E. V. *Sequence − Evolution − Function*; Kluwer Academic Publishers: Norwell, 2003.
(57) Iyer, L. M.; Aravind, L.; Bork, P.; Hofmann, K.; Mushegian, A. R.; Zhulin, I. B.; Koonin, E. V. *Genome Biol.* **2001**, *2*, RESEARCH0051.
(58) Gardy, J. L.; Laird, M. R.; Chen, F.; Rey, S.; Walsh, C. J.; Ester, M.; Brinkman, F. S. *Bioinformatics* **2005**, *21*, 617.
(59) Krogh, A.; Larsson, B.; von Heijne, G.; Sonnhammer, E. L. *J. Mol. Biol.* **2001**, *305*, 567.
(60) Bendtsen, J. D.; Nielsen, H.; von Heijne, G.; Brunak, S. *J. Mol. Biol.* **2004**, *340*, 783.
(61) Yu, C. S.; Lin, C. J.; Hwang, J. K. *Protein Sci.* **2004**, *13*, 1402.
(62) Hofmann, K.; Stoffel, W. *Biol. Chem. Hoppe-Seyler* **1993**, *374,166*.
(63) Schattner, P.; Brooks, A. N.; Lowe, T. M. *Nucleic Acids Res.* **2005**, *33*, W686.
(64) Kall, L.; Krogh, A.; Sonnhammer, E. L. *J. Mol. Biol.* **2004**, *338*, 1027.
(65) Bhasin, M.; Garg, A.; Raghava, G. P. *Bioinformatics* **2005**, *21*, 2522.
(66) Griffiths-Jones, S.; Bateman, A.; Marshall, M.; Khanna, A.; Eddy, S. R. *Nucleic Acids Res.* **2003**, *31*, 439.
(67) Rey, S.; Gardy, J. L.; Brinkman, F. S. *BMC Genomics* **2005**, *6*, 162.
(68) Krogh, A.; Mian, I. S.; Haussler, D. *Nucleic Acids Res.* **1994**, *22*, 4768.
(69) Eddy, S. R. *Bioinformatics* **1998**, *14*, 755.
(70) Haft, D. H.; Loftus, B. J.; Richardson, D. L.; Yang, F.; Eisen, J. A.; Paulsen, I. T.; White, O. *Nucleic Acids Res.* **2001**, *29*, 41.
(71) McGarvey, P. B.; Huang, H.; Barker, W. C.; Orcutt, B. C.; Garavelli, J. S.; Srinivasarao, G. Y.; Yeh, L. S.; Xiao, C.; Wu, C. H. *Bioinformatics* **2000**, *16*, 290.
(72) Wu, C. H.; Nikolskaya, A.; Huang, H.; Yeh, L. S.; Natale, D. A.; Vinayaka, C. R.; Hu, Z. Z.; Mazumder, R.; Kumar, S.; Kourtesis, P.; Ledley, R. S.; Suzek, B. E.; Arminski, L.; Chen, Y.; Zhang, J.; Cardenas, J. L.; Chung, S.; Castro-Alvear, J.; Dinkov, G.; Barker, W. C. *Nucleic Acids Res.* **2004**, *32*, D112.
(73) Schneider, M.; Tognolli, M.; Bairoch, A. *Plant Physiol. Biochem.* **2004**, *42*, 1013.
(74) Hulo, N.; Bairoch, A.; Bulliard, V.; Cerutti, L.; De Castro, E.; Langendijk-Genevaux, P. S.; Pagni, M.; Sigrist, C. J. *Nucleic Acids Res.* **2006**, *34*, D227.
(75) Gattiker, A.; Michoud, K.; Rivoire, C.; Auchincloss, A. H.; Coudert, E.; Lima, T.; Kersey, P.; Pagni, M.; Sigrist, C. J.; Lachaize, C.; Veuthey, A. L.; Gasteiger, E.; Bairoch, A. *Comput. Biol. Chem.* **2003**, *27*, 49.
(76) Tatusov, R. L.; Koonin, E. V.; Lipman, D. J. *Science* **1997**, *278*, 631.
(77) Krieger, C. J.; Zhang, P.; Mueller, L. A.; Wang, A.; Paley, S.; Arnaud, M.; Pick, J.; Rhee, S. Y.; Karp, P. D. *Nucleic Acids Res.* **2004**, *32*, D438.
(78) Osterman, A.; Overbeek, R. *Curr. Opin. Chem. Biol.* **2003**, *7*, 238.
(79) Marcotte, E. M.; Pellegrini, M.; Ng, H. L.; Rice, D. W.; Yeates, T. O.; Eisenberg, D. *Science* **1999**, *285*, 751.
(80) Eisenberg, D.; Marcotte, E. M.; Xenarios, I.; Yeates, T. O. *Nature* **2000**, *405*, 823.
(81) Wright, M. A.; Kharchenko, P.; Church, G. M.; Segre, D. *Proc. Natl. Acad. Sci. U.S.A.* **2007**, *104*, 10559.
(82) Segre, D.; Zucker, J.; Katz, J.; Lin, X.; D'Haeseleer, P.; Rindone, W. P.; Kharchenko, P.; Nguyen, D. H.; Wright, M. A.; Church, G. M. *Omics* **2003**, *7*, 301.
(83) Dandekar, T.; Snel, B.; Huynen, M.; Bork, P. *Trends Biochem. Sci.* **1998**, *23*, 324.
(84) Overbeek, R.; Fonstein, M.; D'Souza, M.; Pusch, G. D.; Maltsev, N. *In Silico Biol.* **1999**, *1*, 93.
(85) Zheng, Y.; Anton, B. P.; Roberts, R. J.; Kasif, S. *BMC Bioinf.* **2005**, *6*, 243.
(86) Fujibuchi, W.; Ogata, H.; Matsuda, H.; Kanehisa, M. *Nucleic Acids Res.* **2000**, *28*, 4029.
(87) Zheng, Y.; Roberts, R. J.; Kasif, S. *Genome Biol.* **2002**, *3*, RESEARCH0060.
(88) Rosenfeld, J. A.; Sarkar, I. N.; Planet, P. J.; Figurski, D. H.; DeSalle, R. *Bioinformatics* **2004**, *20*, 3462.
(89) Enright, A. J.; Iliopoulos, I.; Kyrpides, N. C.; Ouzounis, C. A. *Nature* **1999**, *402*, 86.
(90) Gelfand, M. S.; Novichkov, P. S.; Novichkova, E. S.; Mironov, A. A. *Briefings Bioinf.* **2000**, *1*, 357.
(91) Yang, C.; Rodionov, D. A.; Li, X.; Laikova, O. N.; Gelfand, M. S.; Zagnitko, O. P.; Romine, M. F.; Obraztsova, A. Y.; Nealson, K. H.; Osterman, A. L. *J. Biol. Chem.* **2006**, *281*, 29872.
(92) Faith, J. J.; Hayete, B.; Thaden, J. T.; Mogno, I.; Wierzbowski, J.; Cottarel, G.; Kasif, S.; Collins, J. J.; Gardner, T. S. *PLoS Biol.* **2007**, *5*, e8.
(93) Mika, S.; Rost, B. *PLoS Comput. Biol.* **2006**, *2*, e79.
(94) Aytuna, A. S.; Gursoy, A.; Keskin, O. *Bioinformatics* **2005**, *21*, 2850.

CR068308H