Computational Methods for Gene Finding in Prokaryotes

Mihaela Angelova, Slobodan Kalajdziski, Ljupco Kocarev

Faculty of Electrical Engineering and Information Technologies, Skopje, Macedonia
mihaelaangelova@yahoo.com

Abstract. Gene finding is crucial in understanding the genome of a species. The long genomic sequence is not very useful, unless its biologically functional subsequences (genes) are identified. Along with the ongoing revolution in sequencing technology, the number of sequenced genomes has increased drastically. Therefore, the development of reliable automated techniques for predicting genes has become critical.
Automatic gene prediction is one of the essential issues in bioinformatics. Many approaches have been proposed and a lot of tools have been developed. This paper compiles information about some of the currently most widely used gene finders for prokaryotic genomes, explaining the underlying computational methods and highlighting their advantages and limitations. Finally, the gene finders are tested on a strain with high GC-content.
Keywords: *ab initio* prediction, gene prediction, homology-based search, prokaryotes

1 Introduction

The development of automated sequencing technologies with dramatically lower cost and higher throughput has revolutionized biological research, allowing scientists to decode genomes of many organisms [1]. Prokaryotic genomes are sequenced at an increasing rate. After a genomic sequence is reconstructed from the sequencing data, the next and most important step is to understand the content of the genome i.e. identify the gene loci and their functions. These genes then become the basis for much further biological research.

In the earliest days, genes were identified with experimental validation on living cells and organisms, which is the most reliable method, but costly and labor intensive. Since there can be thousands of genes in one bacterial genome, computational methods are essential for automatic analysis of uncharacterized genomic sequences.

At present, there are many prokaryotic gene finders, based on different approaches. Generally, the gene prediction approaches can be divided into two classes: intrinsic (*ab initio*) and extrinsic (homology-based). It is common for gene finders of both types to be used in a gene finding project, owing to their complementary nature. They all have their own advantages and weaknesses. Therefore, high-quality gene annotation of microbial genomes remains an ongoing challenge.

2 Computational Methods for Gene Prediction

DNA sequences that encode proteins are not a random combination of codons (triplets of adjacent nucleotides designating a specific amino acid). On the contrary, the order of codons obeys certain biological rules and is maintained during evolution. Certain patterns in the codon arrangements have been recognized. For example, conserved preference for certain codon pairs within the coding region is confirmed in the three domains of life [2]. Moreover, the genetic code is used differently in different bacterial species. In most bacteria, the synonymous codon usage (usage of codons that represent the same amino acid) varies not only between organisms [3], but also within an organism, since the horizontally transferred genes tend to have different codon usage from the host. These patterns in the genetic code and their conservancy with evolution can be very helpful for the computational gene finding. They enable algorithms for gene prediction to rely on statistics that describe gene patterns or on sequences' resemblance to conserved annotated proteins.

Two general classes of computational methods are adopted: *ab initio* prediction (intrinsic) and homology-based search (extrinsic) [4]. The first method uses gene structure as a guide to gene detection. The latter one, which is based on the observation that coding sequences are more conserved than the non-coding genes and intergenic regions, compares the genome to the available gene sequences and searches for significant homology.

2.1 *Ab initio* Gene Finders

Ab initio approaches do not use extrinsic information for gene prediction. Instead, they inspect the input sequence and search for traces of gene presence. Intrinsic methods extract information on gene locations using statistical patterns inside and outside gene regions as well as patterns typical of the gene boundaries. There are specific DNA motifs called signals that indicate a neighboring gene, *e.g.* promoters, start and stop codons. Apart from signals, *ab initio* methods discover gene signposts based on content search, looking for patterns of codon usage specific for the organism. Mainly, *ab initio* algorithms implement intelligent methods to represent these patterns as a model of the gene structure in the organism. The most widespread algorithms for gene finding in prokaryotes are based on Markov models and dynamic programming.

Prokaryotic gene features useful for *ab initio* prediction. The simplest *ab initio* method is to inspect Open Reading Frames (ORFs). An ORF is a sequence of bases encompassed by the translation initiation and termination site *i.e.* the coding sequence of prokaryotic organisms [5]. Every coding Deoxyribonucleic Acid (DNA) has six possible reading frames: three on the direct (positive) strand and three in the complementary strand read in the opposite direction of the double-stranded DNA. The nucleotides on the positive strand are grouped into triplets starting from the first (+1), second (+2) or third (+3) nucleotide in the start codon. Therefore, there are three possible reading frames in the positive strand. The same follows for the negative strand, by looking at the sequence from the opposite side. An ORF consists of

consecutive triplets and terminates with the first stop codon it encounters. Typically, only one reading frame, the ORF, is used to translate the gene. Therefore, the prokaryotic gene finder should primarily be able to identify which of the six possible reading frames contains the gene i.e. is an ORF. In general, bacterial genes have long ORFs. This is a hint for gene finding. For example, if the frame +1 has the longest sequence without a stop codon, then its amino acid sequence most probably leads to the gene product.

Nevertheless, this is a good, but not assuring indication for selecting the correct ORF. Not every ORF is a coding region. Telling the difference between genes and random ORFs is the most important goal of the gene finding process [5]. Even if we tune a certain length threshold and define that ORFs longer than that threshold are genes, differentiating between short genes and occasional ORFs remains a problem.

There are some other characteristics of the prokaryotic gene that pose difficulties for the gene finding process. Identifying the right ORFs is deteriorated when two ORFs overlap. Although this is considered to happen rarely in prokaryotes, it is difficult to automatically resolve the problem.

Moreover, there are multiple start codons. In most cases, ATG is the start codon that suggests initiation of translation. Occasionally, GTG and TTG act as initiation sites [6]. Multiple start codons can cause ambiguities, because their presence does not ensure translation initiation.

In conclusion, there is no straightforward way to find genes based on their features. Therefore, *ab initio* gene finders rely not only on signal sensors (start and stop codons, promoters, etc.), but they also use content sensors, such as patterns of codon usage or other statistically inferred features.

Markov Model Based Algorithms. Several highly accurate prokaryotic gene-finding methods are based on Markov model algorithms.

The *GeneMark* family [7] includes two major programs, called GeneMark [8] and GeneMark.hmm [9]. Analysis of DNA from any prokaryotic species without a pre-computed species-specific statistical model is enabled by a self-training program, GeneMarkS [10].

GeneMark uses a Bayesian formalism to assess the *a posteriori* probability that a given short fragment is part of a coding or non-coding region. These calculations are performed using Markov chain models. The idea behind this is that there are specific correlations between adjacent nucleotides in chromosomal DNA sequences. Markov chains have shown to be appropriate in inferring the statistical description of the gene structure.

In mathematical terms, a Markov chain is a discrete random process that evolves through the states from the set $S = \{s_1, s_2,\ldots,s_r\}$. The conditional distribution of any future state depends only on the k preceding variables, for some constant k. In the context of gene prediction, the sequence of random variables $X_1, X_2,\ldots, X_k$ take on values from the set of bases (A, C, G, T) and a Markov chain models the probability that a given base b follows the k bases immediately prior to b in the sequence. Using a training set, a Markov chain captures statistical information about a sequence by computing the probability that a certain nucleotide x_i appears after a sequence s_i e.g. $p\ (x_i = A \mid s_i = TTGCA)$, k = 5. The three codon positions have different nucleotide frequency statistics. Therefore, in order to model the codon usage, normally the

variables of the Markov chain are sets of three nucleotides (codons) or multiples of codons. For this reason, the orders of the Markov chains, k, used for prediction are 2, 5, 8, and so on. For the purpose of modeling protein-coding regions, GeneMark utilizes a three-periodic inhomogeneous Markov model (transition probabilities change with time), because the DNA composition and features vary among different species [11]. Ordinary (homogenous) Markov models are found to be appropriate for non-coding DNA.

GeneMark is the oldest method based on Markov models. It does not offer high accuracy, because it lacks precision in determining the translation initiation codon [9]. Markov chain model of the DNA sequences is firstly introduced in GeneMark. The initial success of GeneMark has paved the way for further research in this direction.

GeneMark.hmm is designed to improve GeneMark in finding exact gene starts. Therefore, the properties of GeneMark.hmm are complementary to GeneMark. GeneMark.hmm uses GeneMark models of coding and non-coding regions and incorporates them into hidden Markov model framework. In short terms, Hidden Markov Models (HMM) are used to describe the transitions from non-coding to coding regions and *vice versa*. GeneMark.hmm predicts the most likely structure of the genome using the Viterbi algorithm, a dynamic programming algorithm for finding the most likely sequence of hidden states. To further improve the prediction of translation start position, GeneMark.hmm derives a model of the ribosome binding site (6-7 nucleotides preceding the start codon, which are bound by the ribosome when initiating protein translation). This model is used for refinement of the results.

Both GeneMark and GeneMark.hmm detect prokaryotic genes in terms of identifying open reading frames that contain real genes. Moreover, they both use pre-computed species-specific gene models as training data, in order to determine the parameters of the protein-coding and non-coding regions.

Acceleration of microbial genome sequencing has led to the need for non-supervised gene finding methods. *GeneMarkS* combines GeneMark.hmm and GeneMark with a self-training procedure. The main focus of GeneMarkS is detecting the correct translation initiation sites. It creates a statistical model, runs the GeneMark.hmm program, and corrects the model based on the results. The steps are repeated iteratively until convergence.

Glimmer3.0 [12] The core of Glimmer is Interpolated Markov Model (IMM), which can be described as a generalized Markov chain with variable order. After GeneMark introduces the fixed-order Markov chains, Glimmer attempts to find a better approach for modeling the genome content. The motivational fact is that the bigger the order of the Markov chain, the more non-randomness can be described. However, as we move to higher order models, the number of probabilities that we must estimate from the data increases exponentially. The major limitation of the fixed-order Markov chain is that models from higher order require exponentially more training data, which are limited and usually not available for new sequences. However, there are some oligomers from higher order that occur often enough to be extremely useful predictors. For the purpose of using these higher-order statistics, whenever sufficient data is available, Glimmer IMMs.

Glimmer calculates the probabilities for all Markov chains from 0-th order to 8-th.

If there are longer sequences (*e.g.* 8-mers) occurring frequently, IMM makes use of them even when there is insufficient data to train an 8-th order model. Similarly, when the statistics from the 8-th order model do not provide significant information, Glimmer refers to the lower-order models to predict genes.

Opposed to the supervised GeneMark, Glimmer uses the input sequence for training. The ORFs longer than a certain threshold are detected and used for training, because there is high probability that they are genes in prokaryotes. Another training option is to use the sequences with homology to known genes from other organisms, available in public databases. Moreover, the user can decide whether to use long ORFs for training purposes or choose any set of genes to train and build the IMM.

There are many annotation services that incorporate Glimmer or GeneMark in their pipelines such as RAST [13], Maker [14] and JCVI Annotation Service [15].

AMIGene [16]. The reason for including AMIGene in the list of gene finders revised in this paper is that AMIGene can be very helpful in some cases. The interesting thing about AMIGene is that it serves as substitution for manual curation, because it searches the most likely CoDing Sequences (CDSs) in the output of a GeneMark-like program.

AMIGene predicts the genome structure in the same way as GeneMark. In addition to that, AMIGene investigates codon usage patterns and relative synonymous codon usage in the predicted CDSs, using multivariate statistical technique of factorial correspondence analysis (FCA) and k-means clustering. AMIGene uses these results to evaluate and filter predicted genes. The construction of gene classes based on codon usage can uncover small genes, which are difficult to spot using the typical model.

AMIGene is not yet suitable for identifying true translation initiation sites and does not take into account overlaps between adjacent CDSs. Considering these drawbacks and considering that AMIGene predicts only the most likely CDSs, it follows that it is a good idea to use AMIGene in combination with other gene finders.

FGenesB [17] is another Markov chain-based algorithm, claimed to be more accurate than GeneMarkS and Glimmer. Unlike them, it finds tRNA and rRNA genes, in addition to coding sequences. Initial predictions of ORFs are used as training set for 5^{th} order in-frame Markov chains for coding regions, 2^{nd} order Markov models translation and termination sites. FGenesB uses genome-specific parameters, automatically trained using only genomic DNA as an input.

FGenesB annotates the genes i.e. identifies their functions by homology with protein databases. As the rRNA genes are highly conserved with evolution, FGenesB identifies them easily in the genome, by comparing them against bacterial and archaeal rRNA databases, using the Basic Local Alignment Search Tool (BLAST) [18], which is described in the section for homology based search.

In prokaryotic cells, functionally related genes are usually found grouped together in clusters called operons and transcribed as one unit. FGenesB is able to predict operons based on distances between ORFs and frequencies of different genes neighboring each other in known bacterial genomes.

In conclusion, FGenesB integrates model-based gene prediction with homology-based annotation, accompanied by operon, promoter and terminator prediction in

bacterial sequences.

Dynamic programming. Opposing to the other gene finders described so far, *Prodigal* [19] does not rely on the assumption that long ORFs are potential genes with high probability, because it can be misleading for gene prediction in GC-rich organisms. Because the stop triplets (TAA, TGA, TAG) are AT rich, their frequency is lower in organisms with high GC (guanine-cytosine) content. Hence, the probability that long ORFs occur by chance increases proportionally to the GC content [20].

Prodigal self-trains by detailed analysis of the GC frame plot. It calculates the statistical significance of the bases G and C in different frame positions. The GC frame plot consists of three graphs, depicting the GC content of the 1st, 2nd, and 3rd nucleotide from each codon in each open reading frame. In coding DNA, the GC content of the third base (GC3) is often higher in genes, relative to non-coding regions [21]. Based on this, Prodigal builds its gene model, looking for a bias for G or C in the 1st, 2nd and 3rd position of each codon. After determining the potential genes, Prodigal filters them, by examining the translation initiation site, ribosomal binding site (RBS), and the lengths of ORFs. The refined set of genes is used as training data.

Prodigal utilizes the same dynamic programming algorithm both for its preliminary training phase and for its final gene calling phase. It scores each ORF, start-stop pair, some motifs, etc. and uses a dynamic programming procedure to find the optimal pathway among a series of weighted steps.

The disadvantage of Prodigal is that there are some genes such as laterally transferred genes, genes in phage regions, proteins with signal peptides and other that do not match the typical GC frame bias for the organism in question.

In summary, *ab initio* gene finders find most of the genes, but have a significantly bigger number of false positives. At the present, no *ab initio* gene finder is able to clearly distinguish short non-coding ORFs from real genes. Moreover, most gene finders rely on the assumption that long ORFs in prokaryotes are genes, which usually leads to incorrect results in microbes with high GC. This problem is addressed by Prodigal. On the other hand, Glimmer for example, uses long ORFs as its training set. As a consequence of the low frequency of stop codons in GC-rich organisms i.e. increased likelihood of random long ORFs, Glimmer lengthens the genes. Mainly, the predicted genes are longer than the actual ones. Therefore, it is important for gene finders to be GC content indifferent.

Not only the GC content can influence on the accuracy of gene finders, but horizontally transferred DNA sequences can also affect the statistical model. These sequences are not evolutionary connected to the rest of the genome, which is why they differ significantly in the context of codon usage, GC frame bias, etc. The Markov models of GeneMark and Glimmer differentiate between these regions, whereas Prodigal fails to recognize them.

2.2 Homology-Based Search

The lower evolution rate of the coding regions enables that genes are identified by comparison with existing protein sequences. Given a library of sequences of other

organisms, we search the target sequence in this library and identify library sequences (known genes) that resemble the target sequence [22].

Local alignment and global alignment are two methods based on similarity searches. The most common local alignment tool is the BLAST family of programs. BLAST is a widely-used tool for searching similarity by homology-based gene finders. It identifies regions of similarity by first breaking down the query sequence into a series of DNA or protein sequences, and then it searches a local or NCBI database. Once a match is found, it tries to align the two sequences i.e. identify every matching letter, insertion, deletion and substitution.

This evidence-based approach is the most reliable method for gene prediction [20]. It is able to find biologically relevant genes. Moreover, it is based on a very simple concept. This approach helps not only find the gene loci, but also annotate (infer the function of) that region, because homologous sequences are supposed to have the same or similar functions.

The biggest limitation of this approach is that only an insufficient number of genes have significant homology to genes in external databases. This number is even smaller for organisms whose closest relatives are not sequenced, because there are many species-specific genes that are not present in databases.

In conclusion, the most reliable way to identify a gene in a newly sequenced genome is to find a close homolog from another organism. Homology-based search is the simplest and is characterized with high accuracy. However, it requires huge amounts of extrinsic data and finds only half of the genes. Many of the genes still have no significant homology to known genes.

3 Comparison of the Gene Finding Tools

Gene finders may differ on the type of genes they are able to recognize (non-coding RNA or proteins); some of them accept only one genomic sequence as input, whereas others can process multiple sequences; different gene finders may produce output files in different formats. The following table summarizes the features of all the gene finders that are described or mentioned in this paper.

Table 1. Comparison of Some Features for Gene Finders

Gene finders	CDS	stable RNA[a]	FA[b]	developed for :	Output files format
Prodigal	y	n	n	bacterial & archaeal	GBK, GFF or SCO
GeneMark.hmm	y	n	n	prokaryotes	algorithm-specific
GeneMark	y	n	n	prokaryotes	algorithm-specific
GenMarkS	y	n	y	prokaryotes	algorithm-specific
RAST	y	y	y	bacteria and archarea	GTF,GFF3,GenBank,EMBL

JCVI Annotation Service	y	y	y	prokaryotes	algorithm-specific
AMIGene	y	n	n	prokaryotes	EMBL, GenBank, GFF
Glimmer3	y	n	n	prokaryotes	algorithm-specific
EasyGene	y	n	n	prokaryotes	GFF2
Maker	y	n	y	small eukaryotes and prokaryotes	GFF3
Augustus	y	n	y	eukaryotes	GTF (similar to gff). GFF

[a] stable RNA refers to rRNA, tRNA, tmRNA, RNA Component of RNaseP

[b] FA stands for functional annotations i.e. mRNA, operons, promoters, terminators, protein-binding sties, DNA bends

The gene finders listed in Table 1 were tested on the bacterial strain *Pseudomonas aeruginosa* LESB58 (*P.a.* LESB58), which has a high GC-content (~66.3%). Most of these gene finders are specialized for prokaryotes. Although Augustus is developed only for eukaryotes, it offers the option to be trained on a given set of genomes. Therefore, Augustus was trained on the 10 closest genomes of *Pseudomonas aeruginosa* LESB58.

Table 2. Results from Testing the Gene Finders on *P.a.* LESB58

Gene Finder	# Genes	# Genes on the + Strand	# Genes on the - Strand	#Correct Genes	% Correct Genes (compared to the Original)	% Correct Genes from (from all found genes)
Original	6061	2993	3067	6061	100,00%	100,00%
Prodigal	6055	3014	3041	5286	89,14%	87,30%
FGenesB	6197	3094	3103	5070	85,50%	81,81%
Glimmer3.0	6276	3100	3176	5043	85,04%	80,35%
GeneMarkS	6100	3043	3057	5006	84,42%	82,07%
JCVI	6270	3098	3172	5036	83,10%	80,32%
GeneMarkHMM	6129	3055	3074	4920	82,97%	80,27%
Rast	6297	3116	3181	4940	81,52%	78,45%
MED	7475	3708	3767	4747	80,05%	63,51%
Maker with model	6149	3065	3084	4588	75,71%	74,61%
Maker	5884	2904	2980	4370	72,11%	74,27%
Augustus	5268	2587	2681	3529	59,51%	66,99%
AMIGene	6154	3077	3077	2967	50,03%	48,21%
EasyGene	3150	0	3150	2570	43,34%	81,59%

Expectedly, Prodigal coped robustly with the high GC content of the strain *P.a.* LESB58. From the Markov model-based algorithms for gene prediction, FGenesB generated the biggest number of correct genes, performing slightly better than Glimmer and GeneMarkS. Glimmer lengthened genes, resulting in drastically higher average gene length. AMIGene found many genes that were not recognized by other gene finders; however more than half of the genes it predicted were not correct.

Approximately 11.6% from all the genes in P.a.LESB58 were detected by every gene finder.

The Results suggest that Prodigal is preferable for gene prediction in high GC genomes. However, for the purpose of testing the gene finders, the hypothetical proteins were not excluded from the published annotation.

4 Future Directions in Microbial Gene Prediction

Microbial gene identification is a well-studied problem. Since the early eighties of the twentieth century, there has been great progress in the development of computational gene prediction. There is still much room for improvement, especially in understanding the translation initiation mechanisms.

The accuracy of most of the gene finding methods drops considerably, when high GC content genomes are observed. Moreover, most methods tend to predict too many genes, mainly because of the problem of predicting short genes. Although many short genes without a BLAST hit might be real, the likelihood is that the most are false positives.

The evaluation system of gene prediction programs is still in need of improvement. The authors of all the gene finders mentioned in this review estimated the accuracy of the tools by predicting genes in complete genomes and then comparing the output to the “known” genes. However, it is estimated that 10-30% of the annotated genes are not protein-coding genes, but rather ORFs that occur by chance [20]. The gene finders exclude hypothetical proteins for testing purposes, because published annotations are not 100% accurate; therefore, the question remains open as to how accurate these predictions really are. The need for more reasonable criterions for evaluation of gene prediction programs is apparent.

It is important to improve current methodologies to obtain higher quality gene predictions, translation initiation site prediction and reduction in the number of false positives, in order to minimize the need for manual curation. Future gene finders should enable automatic gene prediction without human intervention.

5 Conclusion

At the present, there is no tool for gene prediction that automatically finds all the genes in a given genomic DNA sequence with 100% accuracy. The most reliable method for identifying genes is by similarity to a protein in other organism. Genes with no match to known proteins can be predicted using statistical measures.

Every algorithm for gene prediction has its advantages and limitations. Currently, the best approach seems to be a combination of gene finders, followed by evidence-based manual curation.

Acknowledgments. Ljupco Kocarev thanks ONR Global (Grant number N62909-10-1-7074) and Macedonian Ministry of Education and Science (grant 'Annotated graphs in system biology') for partial support.

References

1. Kahvejian,A., Quackenbush,J., and Thompson,J.F.: What would you do if you could sequence everything? Nat Biotechnol, 26,1125-1133 (2008).
2. Tats,A., Tenson,T., and Remm,M.: Preferred and avoided codon pairs in three domains of life. Bmc Genomics, 9 (2008).
3. Ermolaeva,M.D.: Synonymous codon usage in bacteria. Curr.Issues Mol Biol, 3,91-97 (2001).
4. Borodovksy, M., Hayes, W.S., and A.V.Lukashin: In R.L.Charlebois (ed), Organization of Prokaryotic Genomes. ASM Press,pp 11-33 (1999).
   Charlebois,R.L.: Statistical Predictions of Coding Regions in Prokaryotic Genomes by Using Inhomogeneous Markov Models. American Society Microbiology, pp 11-33 (1999).
5. Jin Xiong: Essential Bioinformatics. Cambridge University Press,pp 97-112 (2006).
6. Besemer,J., Borodovsky,M.: GeneMark: web software for gene finding in prokaryotes, eukaryotes and viruses. Nucleic Acids Res, 33,W451-W454 (2005).
7. Borodovksy, M., McIninch J.D. GeneMark: Parallel Gene Recognition for Both Strands. Comupt Chem, 17,123-133 (1993).
8. Lukashin,A.V., Borodovsky,M.: GeneMark.hmm: new solutions for gene finding. Nucleic Acids Res, 26,1107-1115 (1998).
9. Besemer,J., Lomsadze,A., and Borodovsky,M.: GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. Implications for finding sequence motifs in regulatory regions. Nucleic Acids Res, 29,2607-2618 (2001).
10. Borodovksy, M., McIninch J.D.: Recognition of genes in DNA sequence with ambiguities. BioSystems, 30,161-171 (1993).
11. Delcher,A.L., Bratke,K.A., Powers,E.C., and Salzberg,S.L.: Identifying bacterial genes and endosymbiont DNA with Glimmer. Bioinformatics, 23,673-679 (2007).
12. NMPDR, http://rast.nmpdr.org/
13. Yandell Lab, , http://www.yandell-lab.org/software/maker.html
14. J. Craig Venter Institute, http://www.jcvi.org/cms/research/projects/prokaryotic-annotation-pipeline/overview/
15. Bocs,S., Cruveiller,S., Vallenet,D., Nuel,G., and Medigue,C.: AMIGene: Annotation of MIcrobial genes. Nucleic Acids Research, 31,3723-3726 (2003).
16. Softberry Inc. FGENESB Suite of Bacterial Operon and Gene Finding Programs, http://linux1.softberry.com/berry.phtml?topic=fgenesb&group=help&subgroup=gfindb 2010. 6-1-2010.
17. Altschul,S.F., Gish,W., Miller,W., Myers,E.W., and Lipman,D.J. (1990) Basic Local Alignment Search Tool. Journal of Molecular Biology, 215,403-410.
18. Doug Hyatt, Gwo-Liang Chen, Philip F.LoCascio, Miriam L.Land, Frank W.Larimer, and Loren Hauser (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. Bmc Bioinformatics, 11-119.
19. Skovgaard,M., Jensen,L.J., Brunak,S., Ussery,D., and Krogh,A. (2001) On the total number of genes and their length distribution in complete microbial genomes. Trends in Genetics, 17,425-428.
20. Bibb,M.J., Findlay,P.R., and Johnson,M.W.: The relationship between base composition and codon usage in bacterial genes and its use for the simple and reliable identification of protein-coding sequences. Gene, 30,157-166 (1984).
21. Baxevanis, A.D., and Ouellette, B.F.F.: ”Sequence alignment and Database Searching” in *Bioinformatics: A Practical Guide to the Analysis of Genes and Proteins*. John Wiley & Sons, New York, pp. 187-212 (2001)