Towards Automatic Detecting of Overlapping Genes - Clustered BLAST Analysis of Viral Genomes

Klaus Neuhaus[1], Daniela Oelke[2], David Fürst[3],
Siegfried Scherer[1], and Daniel A. Keim[2]

[1]Chair of Microbial Ecology, Technische Universität München,
Weihenstephaner Berg 3, 85354 Freising, Germany

[2]Chair of Data Analysis and Visualization, Universität Konstanz
Universitätsstr. 10, 78457 Konstanz, Germany

[3]Chair of Data Management and Data Exploration, Rheinisch-Westfälische
Technische Hochschule Aachen, Informatik 9, 52056 Aachen, Germany

Abstract. Overlapping genes (encoded on the same DNA strand but in different frames) are thought to be rare and, therefore, were largely neglected in the past. In a test set of 800 viruses we found more than 350 potential overlapping open reading frames >500 base pairs which generate BLAST hits, indicating a possible biological function. Interestingly, five overlaps with more than 2000 bp were found, the largest may even contain triple overlaps. In order to perform the vast amount of BLAST searches required to test all detected open reading frames, we compared two clustering strategies (BLASTCLUST and k-means) and queried the database with one representative only. Our results show that this approach achieves a significant speed-up while retaining a high quality of the results (>99% precision compared to single BLAST queries) for both clustering methods. Future wet lab experiments are needed to show whether the detected overlapping reading frames are biologically functional.

Key words: overlapping genes, clustering, BLAST analysis

1 Introduction

1.1 Overlapping Reading Frames and Overlapping Genes

DNA consists of two complementary strands, uses a triplet code and, consequently, open reading frames (ORFs), which may code for proteins, are possible in six reading frames overlapping in different phases. A protein coding reading frame on a given DNA sequence is, by convention, phase +1, the next frames are +2 and +3, and, on the other strand -1, -2, and -3 (Fig. 1). This construction results in a considerable theoretical coding density.

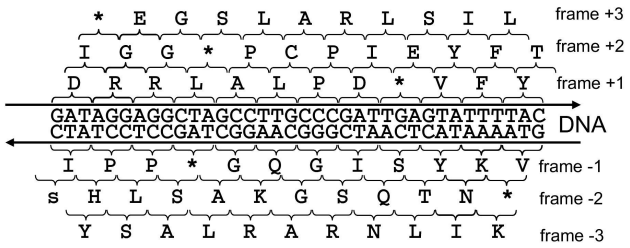

Fig. 1. A double strand of DNA with the different possible reading frames. Encoded amino acids are shown in the single letter code according to the standard genetic code. Stop codons, which do not code for an amino acid, are depicted by a star.

The term "overlapping gene" has been used in the literature for several related biological phenomena. In order to avoid confusion, we introduce a distinction between trivial and embedded overlaps. In case of short, trivial overlaps, the overlapping sequence has no important function at the protein level, but may play a role in the regulation of gene expression, e.g., in transitional coupling [1, 2]. The process of gain and loss of trivial overlapping genes has been modeled by mutational events, which displace the start or stop codons [3–9]. The focus of this project, however, is on embedded genes which encode two completely different functional amino acid chains (proteins) in different phases of the same DNA locus. Here, a protein coding reading frame is largely or entirely superposed on an annotated reading frame (Fig. 2), and is therefore termed "embedded ORF" or, if a function of the encoded protein has been demonstrated, "embedded gene".

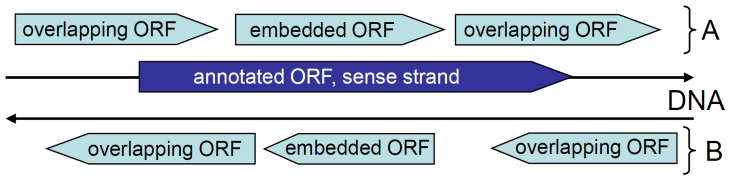

Fig. 2. Types of ORFs to be investigated. A: sense strand ORFs, either partially or completely overlapping. They can be in phase +2 or +3, since the annotated gene is by definition in phase +1. B: antisense strand ORFs, either partially or completely overlapping. They can be in phase -1, -2 or -3.

Usually, genome annotation programs consider embedded ORFs as being non-functional. The rationale may be an intuitive one: It rests on the presumed improbability of embedded genes to originate by chance [10, 11]. Furthermore, overlaps pose severe restrictions on the evolution of both ORFs involved [7] since many mutations in one phase directly affect the amino acid sequence encoded in the other phase [12, 13]. Nevertheless, the first fully sequenced organism, bacte-

riophage ΦX174, contains several embedded genes [14]. Subsequently, a number of embedded genes have been discovered in viruses [15–18] and only recently, the existence of several overlapping genes in other organisms has been acknowledged [19, 20]. Okamura et al. [21] suggested that amino acid chains encoded in alternative reading frames are a hidden coding reserve serving as novelty pool. Once a frame shift mutation occurs, such formerly hidden ORFs become exposed, which means that they are translated. Along those lines we wanted to examine how many of the (overlapping) reading frames currently not annotated have sufficient similarity to annotated genes to generate a BLAST hit in GenBank.

1.2 Detection of embedded genes by computational methods

An ORF, by definition, starts with a start codon and ends with a stop codon. Clearly, not all ORFs are genes. To identify the genes among the many ORFs in a genome during the annotation process is one of the central tasks of bioinformatics. Numerous algorithms have been developed by many bioinformatic groups, such as GeneMark, Glimmer, ZCURVE, BLASTX, FASTA, ORPHEUS or EasyGene (for an overview see [22]). After identification, such ORFs which are likely to be true genes are labeled "annotated" and recorded in genome databases. The aim of this work is to examine genomes by using the implicit state-of-the-art knowledge recorded in databases in terms of annotated genes to see, if "hidden" overlapping reading frames can be discovered. In this feasibility study, we restricted ourselves to viral genomes sequenced until May 2008. However, the main problem is that the number of ORFs encoded in genomes is huge and, therefore, even searches which use a locally installed copy of these databases would take months to complete. An efficient strategy to perform these searches is therefore imperative. One important method to reduce the number of database queries is to use clustering algorithms to meaningfully group the ORFs (see [23, 24] for a review of common clustering algorithms) and then only perform one query per cluster. This allows to query databases such as GenBank using BLAST with only a fraction of the ORFs and to transfer the query results to the rest of the cluster without risking that the introduced error is too large. In later stages, received hits will be analyzed with further bioinformatic methods, e. g. promoter predictions and alike.

2 Computational methods for the detection of overlapping genes

2.1 The data

For our analysis, we downloaded all available viral nucleotide sequences from [25] on May 26, 2008 (nearly 3,000 viruses). All open reading frames (ORFs in six reading frames) were extracted from the viral sequences using *getorf* [26]. ORFs with less than 150 base pairs have not been considered, since smaller ORFs rarely encode for a functional protein [27]. In total, about 229,000 ORFs with at least

150 base pairs were extracted. To find out if the ORFs extracted eventually encode functional proteins a comparison with the NCBI-Protein-Database *nr* [28] is conducted. At the day of the download (May 26, 2008) the database contained 6,544,368 protein sequences, totalling 5.33 GB of data. To query the database the ORFs are translated to the corresponding amino acid sequence. This poses a lesser constraint in finding potentially functional sequences in the ORFs not originally annotated [22].

2.2 Querying the database

For querying the database we use the Basic Local Alignment Search Tool (BLAST) that comprises a set of similarity search programs that were designed to find regions of similarity between biological sequences [29]. BLAST allows searching large databases for optimal local alignments. A list of the sequences with the best local alignments is returned including similarity scores for each sequence. In order to efficiently access the database, the collection of BLAST algorithms was installed locally instead of using the web based version located on the NCBI server.

For an arbitrarily chosen subset of 76,928 ORFs the above mentioned NCBI-Protein-Database *nr* was queried using BLASTP, the algorithm of the BLAST suite for querying protein databases. We used default BLASTP parameters except the cut-off for the e-value has been set to ≤ 0.1. About 43% of the sequences generated a hit which includes the already annotated genes. In average, it took about 47.5 seconds per query. In total, we needed more than 42 days to process the test dataset. Processing all 229,000 ORFs would have taken about 4 month. Thus, the approach is clearly not efficient for future studies comprising more sequence data.

2.3 Clustering for speed-up

To cut runtime, we first cluster the sequences according to their similarity and subsequently query the database with only one representative.

We compared the results of two different clustering algorithms: BLASTCLUST [30] from NCBI [25] and k-means [24]. BLASTCLUST provides hierarchical clustering based on the single linkage approach. Basically, it implements the BLAST-algorithms, which take evolutionary relatedness into account. The advantages are to use end-to-end the same algorithm, and that two sequences can be directly compared without transforming them into an information reduced vector. In contrast to this, applying k-means as a partitioning-based clustering algorithm requires to transform each sequence into a point in Euclidean space. A histogram is formed by counting the occurrence of each amino acid in an ORF. The result is expressed as a 20-dimensional feature vector and similar sequences are assumed to locate at a similar position in this feature space.

Running BLASTCLUST with the default settings resulted in a set of 181,015 clusters. By subsequently relaxing the similarity requirements for sequences that are placed in the same cluster, the number of clusters was reduced to

164,593 (score density threshold S=1.0), 160,915 (S=0.5), 156,009 (S=0.001), and 121,774 (S=5, length covered L=0.1)[1] . The least stringent clustering reduced the dataset approximately half to the starting number. In order to get comparable results the exact same amount of clusters were generated using k-means. Our analysis for both methods showed that there are many clusters that contain only a single sequence and only few containing 20 or more sequences. The reason for this is that the applied thresholds using BLASTCLUST are quite strict to ensure that the whole data set is represented well. Regarding the speed of the clustering process the k-means algorithm turned out to be about five times faster than BLASTCLUST (approximately 7.5h compared to 38h, respectively). However, this difference becomes insignificant if we look at the time required for the total analysis. Figure 3 shows the total runtimes for clustering and data base queries. For the smallest number of clusters, the approach saved up to 45% of the runtime compared to querying each single sequence which in case of about 229,000 candidate sequences accounts for a saving of 58 days. Since BLAST is a computationally demanding algorithm, this achievement is significant. Further runtime reductions using BLAST necessitates special computer hardware [31].

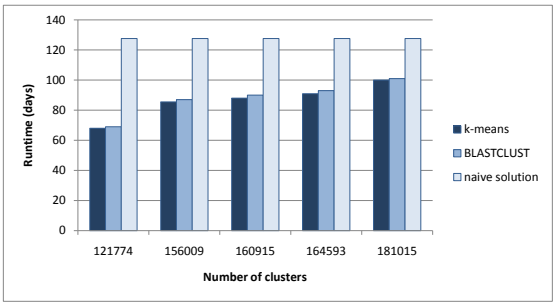

Fig. 3. Comparison of the total runtimes needed as a function of the number of clusters. For comparison the time that would be needed when querying each single sequence is given as well.

2.4 Effectiveness of the approach

Our experiments presented in the former section clearly show that the process can be significantly accelerated by applying clustering techniques before the database is queried. However, this approach can only be considered as useful if the quality of the results remains acceptable.

[1] The S-value denotes the score density which is calculated by dividing the BLAST score by the length of the alignment. The L value specifies the percentage of the length of the sequence that must be covered. If not specified, L is set to 1.0 (=100%) in our experiments.

In order to evaluate this, a measure for effectiveness has to be found. Key for the effectiveness of our approach is to get clusters with a high purity. A cluster is considered as "pure" if it contains either only sequences that generate a BLAST hit or only sequences that do not generate a BLAST hit. If this is the case, then our assumption holds that the result that we get for one sequence can be transferred to all the other sequences in the same cluster. We measure this purity by calculating a precision score for each cluster. The precision is thereby defined as:

$$Precision(C) = max\{Precision_f(C), Precision_{nf}(C)\} \tag{1}$$

where

$$Precision_f(C) = \frac{O_f}{O_a},$$

$$Precision_{nf}(C) = \frac{O_{nf}}{O_a}.$$

with
O_f = number of functional ORFs
O_{nf} = number of non-functional ORFs
O_a = number of all analyzed ORFs

In the experiment that is described in section 2.2, we queried the database due to time constraints using a subset of 76,928 sequences. This dataset serves now as basis for evaluation. Since we could not include all sequences in the experiment in section 2.2 due to time constraints, the calculated numbers can only be considered as an approximation of the precision.

While equation 1 gives us a precision value for a single cluster, we would need a value that measures the quality of the complete clustering. To take the significant differences in cluster sizes into account, we use a weighted average to calculate the cluster precision (equation 2; k = number of clusters, t_i = number of sequences in cluster C_i).

$$Precision = \frac{\sum_{i=1}^{k}(Precision(C_i) \cdot t_i)}{\sum_{i=1}^{k} t_i} \tag{2}$$

Both clustering algorithms, BLASTCLUST and k-means, were evaluated by calculating the precision values for the clustering structure which we retrieved from section 2.3. Figure 4 shows the results. Less cluster lower the precision, which is expected since a smaller number of clusters corresponds to a lesser similarity. Despite the fact that the k-means clustering places the sequences in a 20-dimensional Euclidean space without any consideration of biological significance, the average performance of both clustering algorithms is approximately, and quite surprisingly, the same. It somehow appears that the proteins composition is quite sufficient to circumscribe a cluster. The precision values are convincingly high which confirms our assumption that the speed-up that we gain from using clustering algorithms does not significantly decrease the quality of the results. Using our method, with a loss of at most 0.1% of the precision

(for k-means 0.5 percent) we were able to get a speed-up of approximately 33%. If we are willing to accept a loss in the precision of about 1% (2.5% for BLASTCLUST), the acceleration was even higher with savings of about 45%.
Despite the fact that many "cluster" contain only one sequence, the precision drops faster for BLASTCLUST when relaxing stringency (Fig. 4). This is due to the fact that BLASTCLUST tends to cluster different (in the sense of gaining different BLAST hits) sequences in larger clusters, resulting in a dramatic drop in precision for cluster $\geq$12 sequences of around 90%.

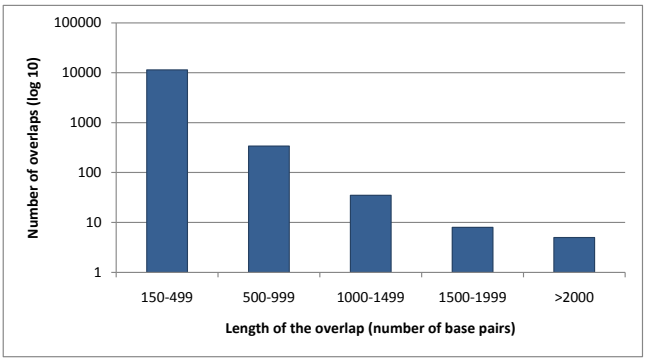

Fig. 4. Comparison of the precision of the clustering BLASTCLUST and k-means for different numbers of clusters.

2.5 Detection of presumed overlapping genes

After recording all ORFs with BLAST hits we determined those which overlap. The analysis of occurring overlaps was conducted on 800 arbitrarily chosen viruses. Their genomes were sequentially examined to find ORFs which overlap in different reading frames. To distinguish between trivial (short) and non-trivial cases, lengths of overlap were recorded and are shown in Figure 5.

3 New Overlapping Genes in Viruses

In total, about 800 viruses were analyzed for overlapping gene sequences which generated a BLAST hit. From those, 62% of the genomes contained at least trivial overlaps, whereas in 44% of the viral genomes overlaps of 100% could be observed. Since non-trivially overlapping genes are considered to be unlikely, one reading frame is usually dismissed in favor of the other one. For instance, Yooseph et al. [10] dismiss overlapping encoded genes if their orthologous cluster is smaller than the cluster of the corresponding gene. However, current genome databases implicitly reflect our state-of-the-art knowledge about which ORF

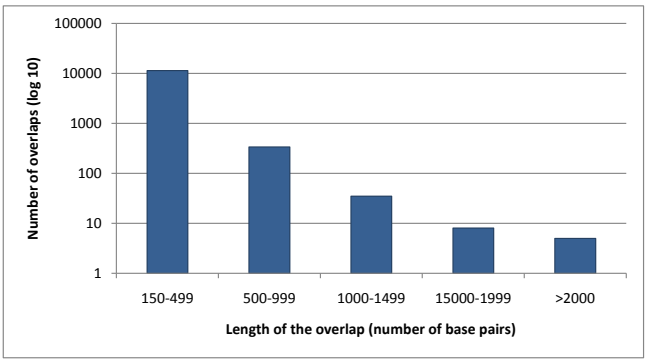

Fig. 5. Histogram of the recorded lengths of the overlaps.

might be (or is) a gene and which one is not. Therefore, a BLAST hit can be used as first approximation for a possible biological function.

Especially, longer overlaps are of biological interest. Figure 5 shows a histogram of the length of the detected overlapping open reading frames which generated a BLAST hit. As expected, there are many short ORFs with less than 500 base pairs. However, there is a considerable number of longer overlapping open reading frames (>350 cases for 500 and more bp) and even five presumed genes with ORFs of more than 2000 base pairs. Since only 800 viruses have been examined, this number must be higher. Indeed, several recent publications about viral genome analysis revealed new overlapping genes. However, those searches included evidence of positive selection (see [32] and references therein).

In the past, viral overlapping genes have been considered to be a "specialty" of these organisms. Most often, viruses have only a limited genome size due to capsid size constraints, with some notable exceptions like *Acanthamoeba polyphaga* mimivirus (genome length ≈1.2Mbp). Indeed, in viruses the number of overlapping genes is inversely correlated with genome length [33]. However, in bacteria, the number of overlapping genes corresponds with genome size and as a rough approximation it can be said that 10-30% of genes overlap [34, 7]. But most of those overlaps are trivial, which means the overlapping encoded amino acid chain is not functional at the protein's level. Biologically more interesting are nested genes in which both protein chains assume a function. A recent textbook like example might be the gene pair *dmdR1* and *adm* from *Streptomyces*. Both genes are antiparallel to each other. DmdR1 regulates iron metabolism and Adm is a regulator for antibiotic production. Quite interestingly, both ORFs were recorded in databases with at that time unknown functions [35].

The longest overlapping ORFs we could find in our analysis is from *Mycobacterium* phage PBI1 and is a very interesting case (Figure 6). The largest ORF, per definition +1, has been annotated as protein g27 (accession no. YP_655223),

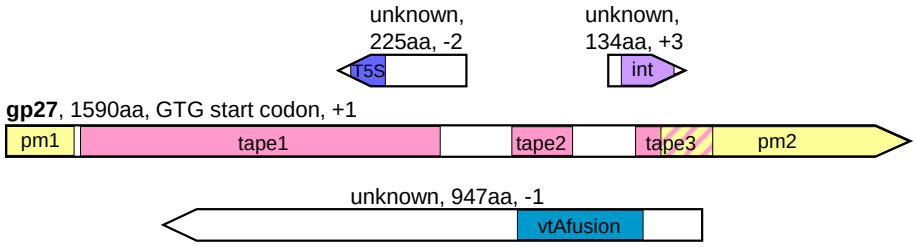

Fig. 6. The genetic locus of the *Mycobacterium* phage PBI1 in which the protein gp27 is encoded (longest arrow; locus tag PBI1p27). This ORF contains three further embedded genes, shown above or below (smaller arrows). For each ORF the length in amino acids (aa), as well as the phase of the reading frame is given. Marked with boxes in different colors are areas generating BLAST hits. int, integron integrase; pm1-2, putative membrane protein of phage origin; T5S, type V secretory pathway protein; tape1-3, tape measure protein; vtAfusion, viral A-type inclusion protein. For further details see text.

but no function has been assigned to it [36]. This reading frame encodes 1590 amino acids and starts according to the GenBank entry with an unusual GTG start codon. However, this start might be questionable, since another CTG start codon can be found upstream and an ATG start codon downstream of the annotated GTG. This reading frame contains two annotated conserved domain fragments, *flhB* from the flagellar biosynthesis protein FlhB and *tra*, which encodes a transglycosylase-like domain. If BLASTed, this ORF will generate three hits with identical ORFs from very similar phages. The e-value of those hits is 0. The next four hits are from different but still related *Mycobacterium* phages, the e-values are in a range of $6 \cdot 10^{-61}$ to $1 \cdot 10^{-51}$ [36]. The next BLAST hit reveals a putative membrane protein of phage origin in *Mycobacterium marinum* strain M; the e-value being $9 \cdot 10^{-51}$ (indicated in Figure 6 with pm1 and pm2). The next hit further down the list, with an e-value of $9 \cdot 10^{-30}$, has similarity to a tape measure protein. Interestingly, several more hits of such tape measure proteins can be found within the first 50 BLAST hits. Areas in which the phage g27 gene generates hits with such tape measure proteins are indicated with tape 1 to 3 in Figure 6. The multiple occurrences of similar genes in a BLAST search indicates that this protein may be indeed a tape measure protein. Interestingly, the sequence of such proteins is under minimal constraints only. It determines the length of a phage tail very much like a ruler. A shorter tape measure protein means a shorter tail and vice versa. Therefore, other protein chains in overlapping reading frames may be easily encoded. Indeed, several additional ORFs can be found embedded in the tape measure protein gene. The largest embedded ORF in frame -1 comprises a protein of 947 amino acids. Amino acid positions 623 to 844 generate a BLAST hit to a viral A-type inclusion protein with an e-value of 0.081 (vtAfusion in Figure 6). Such proteins form inclusion bodies in the host cell during infection [37]. Surprisingly, two more ORFs with BLAST hits are encoded on the same locus of DNA, resulting in triple overlaps. One ORF

with 225 amino acids in frame -2 generates a BLAST hit with a type V secretory pathway protein (ZP_04858685, e-value 0.035, T5S in Figure 6). Those proteins are autotransporters, which transport a protein domain across the membrane of a bacterium [38]. Finally, the last ORF generating a hit encodes 134 amino acids in frame +3. The protein seems to be an integron integrase (e-value 0.031; int in Figure 6). Integrases belong to the large group of mobile genetic elements [38]. It is conceivable that at one point in time such a mobile element jumped into the tape measure protein gene and became incorporated in this DNA locus. Triple overlaps have only rarely been reported [14, 39].

4 Conclusions and Future Prospects

We could demonstrate that in viral genomes several overlapping open reading frames can be found which generate a BLAST hit, which is usually considered as first evidence for a presumed biological function. To speed up BLAST searches for large datasets we implemented clustering strategies. By applying clustering methods previous to querying the database with one representative of each cluster a significant acceleration is possible (in our experiments up to 45%) while retaining a high quality of the results. Our initial results are promising and suggest that further research in this area might be fruitful. For reasons mentioned in the introduction it is comprehensible that embedded ORFs have been almost completely out of focus of experimental and bioinformatic research. Nevertheless, the lack of attention is about to change. Several databases with the aim to aid in the area of overlapping genes have been set up recently [19, 40, 27]. Molecular studies reveal overlapping genes in a diversity of organisms (e. g., see [27]). Therefore, given the availability of many completely sequenced genomes at the beginning of 2010, a number of which will increase steeply in the future [41, 42], we expect the discovery of many yet unknown, but functional overlapping reading frames in natural DNA. Such genes must be tested in wet lab experiments whether they indeed have a biological function.

Acknowledgments. We want to thank Prof. Dr. Thomas Seidl for the fruitful discussions.

References

1. K. Sakharkar, M. Sakharkar, and V. Chow. Gene fusion in *Helicobacter pylori*: making the ends meet. *Antonie van Leeuwenhoek*, 89:169–180, 2006.
2. M. K. Sakharkar, B. S. Perumal, K. R. Sakharkar, and P. Kangueane. An analysis on gene architecture in human and mouse genomes. *In Silico Biol*, 5, 2005.
3. P. Cock and D. Whitworth. Evolution of gene overlaps: Relative reading frame bias in prokaryotic two-component system genes. *J Mol Evol*, 64:457–462, April 2007.
4. Y. Fukuda, T. Washio, and M. Tomita. Comparative study of overlapping genes in the genomes of *Mycoplasma genitalium* and *Mycoplasma pneumoniae*. *Nucl Acids Res*, 27:1847–1853, 1999.

5. D. C. Krakauer. Stability and evolution of overlapping genes. *Evolution*, 54:731–739, 2000.
6. F. Lillo and D. Krakauer. A statistical analysis of the three-fold evolution of genomic compression through frame overlaps in prokaryotes. *Biol Direct*, 2:22, 2007.
7. Y. Luo, C. Fu, D.-Y. Zhang, and K. Lin. Overlapping genes as rare genomic markers: the phylogeny of γ-Proteobacteria as a case study. *Trends Genet*, 22:593–596, 2006.
8. Y. Luo, C. Fu, D.-Y. Zhang, and K. Lin. BPhyOG: An interactive server for genome-wide inference of bacterial phylogenies based on overlapping genes. *BMC Bioinformatics*, 8:266, 2007.
9. N. Sabath, D. Graur, and G. Landan. Same-strand overlapping genes in bacteria: compositional determinants of phase bias. *Biol Direct*, 3:36, 2008.
10. S. Yooseph, G. Sutton, D. B. Rusch, and coworkers. The Sorcerer II Global Ocean Sampling expedition: expanding the universe of protein families. *PLoS Biol*, 5:e16, 2007.
11. H. L. Zaaijer, F. J. van Hemert, M. H. Koppelman, and V. V. Lukashov. Independent evolution of overlapping polymerase and surface protein genes of hepatitis B virus. *J Gen Virol*, 88:2137–2143, 2007.
12. M. Mizokami, E. Orito, K. Ohba, K. Ikeo, J. Y. Lau, and T. Gojobori. Constrained evolution with respect to gene overlap of hepatitis B virus. *J Mol Evol*, 44(Suppl 1):83–90, 1997.
13. A. Nekrutenko, S. Wadhawan, P. Goetting-Minesky, and K. D. Makova. Oscillating evolution of a mammalian locus with overlapping reading frames: an XLαs/ALEX relay. *PLoS Genet*, 1:18, 2005.
14. F. Sanger, G. M. Air, B. G. Barrell, N. L. Brown, A. R. Coulson, C. A. Fiddes, C. A. Hutchison, P. M. Slocombe, and M. Smith. Nucleotide sequence of bacteriophage ϕX174 DNA. *Nature*, 265:687–695, 1977.
15. S. Guyader and D. G. Ducray. Sequence analysis of *Potato leafroll virus* isolates reveals genetic stability, major evolutionary events and differential selection pressure between overlapping reading frame products. *J Gen Virol*, 83:1799–1807, 2002.
16. R. A. Lamb and C. M. Horvath. Diversity of coding strategies in influenza viruses. *Trends Genet*, 7:261–266, 1991.
17. K. M. McGirr and G. C. Buehuring. Tax and rex: overlapping genes of the Deltaretrovirus group. *Virus Genes*, 32:229–239, 2006.
18. A. E. Firth and J. F. Atkins. Analysis of the coding potential of the partially overlapping 3' ORF in segment 5 of the plant fijiviruses. *Virol J*, 6:32, 2009.
19. I. Pedroso, G. Rivera, F. Lazo, M. Chacon, F. Ossandon, F. A. Veloso, and D. S. Holmes. AlterORF: a database of alternate open reading frames. *Nucleic Acids Res*, 36:517–518, 2008.
20. D. S. Kim, C. Y. Cho, J. W. Huh, H. S. Kim, and H. G. Cho. EVOG: a database for evolutionary analysis of overlapping genes. *Nucleic Acids Res*, 37:D698–702, 2009.
21. K. Okamura, L. Feuk, T. Marques-Bonet, A. Navarro, and S. W. Scherer. Frequent appearance of novel protein-coding sequences by frameshift translation. *Genomics*, 88:690–697, 2006.
22. W. H. Majoros. *Methods for Computational Gene Prediction.* Cambridge University Press, 2007.
23. V. Di Gesù. Data Analysis and Bioinformatics, Pattern Recognition and Machine Intelligence. *Lect Notes Comput Sci*, 2007.

24. A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput Surv*, 31:264–323, 1999.
25. National Center for Biotechnology Information (NCBI). NCBI Homepage, 2009. `http://www.ncbi.nlm.nih.gov/`.
26. P. Rice, I. Longden, and A. Bleasby. EMBOSS: The European Molecular Biology Open Software Suite. *Trends Genet*, 16:276–277, 2000.
27. M. Linial. How incorrect annotations evolve–the case of short ORFs. *Trends Biotechnol*, 21:298–300, 2003.
28. National Center for Biotechnology Information (NCBI). The BLAST Databases, 2009. `ftp://ftp.ncbi.nih.gov/blast/db/`.
29. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic Local Alignment Search Tool. *J Mol Biol*, 215(2):403–410, 1990.
30. National Center for Biotechnology Information (NCBI). Documentation of the BLASTCLUST-algorithm. `ftp://ftp.ncbi.nih.gov/blast/documents/blastclust.html`.
31. E. Sotiriades and A. Dollas. A general reconfigurable architecture for the blast algorithm. *J VLSI Signal Process*, 48:189–208, 2007.
32. N. Sabath. *Molecular Evolution of Overlapping Genes*. University of Houston, 2009.
33. R. Belshaw, Oliver G G. Pybus, and Andrew Rambaut. The evolution of genome compression and genomic novelty in RNA viruses. *Genome Res*, 17:1496–1504.
34. Z. I. Johnson and S. W. Chisholm. Properties of overlapping genes are conserved across microbial genomes. *Genome Inform*, 14:2268–2272, 2004.
35. S. Tunca, C. Barreiro, J. J. Coque, and J. F. Martin. Two overlapping antiparallel genes encoding the iron regulator DmdR1 and the Adm proteins control siderophore and antibiotic biosynthesis in *Streptomyces coelicolor* A3(2). *FEBS J*, 276:4814–4827, 2009.
36. G. F. Hatfull, M. L. Pedulla, D. Jacobs-Sera, and coworkers. Exploring the mycobacteriophage metaproteome: phage genomics as an educational platform. *PLoS Genet*, 2:e92, 2006.
37. M. I. Okeke, O. A. Adekoya, U. Moens, M. Tryland, T. Traavik, and O. Nilssen. Comparative sequence analysis of A-type inclusion (ATI) and P4c proteins of orthopoxviruses that produce typical and atypical ATI phenotypes. *Virus Genes*, Epub ahead of print, 2009.
38. N. Dautin and H. D. Bernstein. Protein secretion in gram-negative bacteria via the autotransporter pathway. *Annu Rev Microbiol*, 61:89–112, 2007.
39. X. Zhao, K. M. McGirr, and G. C. Buehring. Potential evolutionary influences on overlapping reading frames in the bovine leukemia virus pxbl region. *Genomics*, 89:502–511, 2007.
40. A. Palleja, T. Reverter, S. Garcia-Vallve, and A. Romeu. PairWise Neighbours database: overlaps and spacers among prokaryote genomes. *BMC Genomics*, 10:281, 2009.
41. I. B. Zhulin. It is computation time for bacteriology. *J Bacteriol*, 191:20–22, 2009.
42. D. Wul, P. Hugenholtz, K. Mavromatis, and coworkers. A phylogeny-driven genomic encyclopaedia of bacteria and archaea. *Nature*, 462:1056–1060, 2009.