Computational Biology Methods and Their Application to the Comparative Genomics of Endocellular Symbiotic Bacteria of Insects

Jennifer Commins, Christina Toft, and Mario A. Fares

Abstract

Comparative genomics has become a real tantalizing challenge in the postgenomic era. This fact has been mostly magnified by the plethora of new genomes becoming available in a daily bases. The overwhelming list of new genomes to compare has pushed the field of bioinformatics and computational biology forward toward the design and development of methods capable of identifying patterns in a sea of swamping data noise. Despite many advances made in such endeavor, the ever-lasting annoying exceptions to the general patterns remain to pose difficulties in generalizing methods for comparative genomics. In this review, we discuss the different tools devised to undertake the challenge of comparative genomics and some of the exceptions that compromise the generality of such methods. We focus on endosymbiotic bacteria of insects because of their genomic dynamics peculiarities when compared to free-living organisms.

Keywords: Comparative Genomics, Orthologs search, BLAST, Functional Categories, Genomics Dynamics.

1. Genomes, Genomes, and More Genomes

The emergence of genome information has overwhelmed our efforts to analyze the unexpected amount of data generated during the last two decades. As an example, today (February, 2009), there are 438 complete microbial genomes and 17 in draft in the J. Craig Venter Institute, Comprehensive Microbial Resource website (URL: http://cmr.jcvi.org/tigr-scripts/CMR/CmrHomePage.cgi) considering that this is only a single resource we estimate that

Shulin Li (ed.), *Biological Procedures Online, Volume 11, Number 1*
© to the author(s) 2009
DOI: 10.1007/s12575-009-9004-1 URL: springerprotocols.com; springerlink.com

Petitioner Illumina, Inc., EX1035, p. 52

the number of completed genomes will be in the order of double that by the end of 2009 with a considerable percentage of these already published in the literature. Already the Entrez Genome project website controlled by National Center for Biotechnology Information (NCBI) reports that on February 3, 2009, 857 genomes are complete, 815 are in draft assembly, and 989 are in progress (http://www.ncbi.nlm.nih.gov/genomes/static/gpstat.html). The number of institutes worldwide with increasing sequencing capacities has been rising at an exponential rate and the first results of analyzing such data have solved old and long debated hypotheses and also have generated breakthrough ideas that have opened new avenues in all fields of genetics and evolutionary biology. However, our ability to cope technically with the amount of generated raw data has become seriously compromised, fueling many initiatives aimed at developing computational tools to analyze genomic and proteomic data. Many of these tools have been developed to perform comparative genomic analyses; each tool has had to face many of the complexities that biologically driven genome remodeling phenomena cause, such as genome duplication, rearrangement, and shrinkage. In this review, we first discuss the different technologies developed to perform genomic and proteomic analyses. We then focus on the importance of the developed tools to study biologically important phenomena such as genome duplication, the dynamics of genome rearrangement, and genome shrinkage that is associated with the intracellular life of bacteria.

2. Common Methods in Comparative Genomics

Comparative genomic methods are vast in number as well as function. A decision about the best way to do something is often a long and arduous task in this field, a task that has resulted in the design and reengineering of many of the tools that are available. To describe every method in this area of research would be next to impossible, and so, this text will provide a snapshot of what is available for many of the common tasks in comparative genomics. The logical place to start is of course the beginning—genome sequencing, assembly, and closing, then continuing to discuss the intricacies of comparative genomics.

While in the past comparative genomics has concentrated on sequencing single genomes and parts of genomes, current excitement lies with the sequencing of environmental communities. This field of research, entitled metagenomics is fast growing and the current hot topic. Its application is most utilized to character-

ize unculturable organisms (an estimated 99% of microbes cannot be cultivated in a laboratory environment (*1*)), but it has also made it possible to sequence genomes without the problems that are associated with cultures maintained in laboratories (*2*). Metagenomics has transformed the uses of such organisms by allowing the focus to move from those that can be cloned in culture (*3*). Depending on the source of the environmental sample to be subjected to environmental shotgun sequencing, a colossal variation in the number of identified species may result. Just looking at prokaryotes alone, as few as five species were identified in a community sequencing carried out on acid mine biofilm (Tyson et al. (*4*)), in contrast, as many as 3,000 species were sequenced from a soil sample taken in Minnesota, USA analyzed by Tringe et al. (*5*). For a comprehensive review of this subject, *see* (*6*).

3. Sequencing

In the context of γ-proteobacteria, sequencing is commonly carried out using a shotgun approach. This technique is popular and is widely used in the generation of long sequences, such as those found in whole genomes. Briefly, this approach involves the sequencing of random small cloned fragments, known as reads, in both directions from the genome. This fragmented reading of the genome is carried out multiple times to provide good coverage and overlap within the sequencing. Having good quality overlap/coverage allows the reads to be assembled into their original order, thus reconstructing the genome (**Fig. 1**a). Not surprisingly, reconstructing the genome from short overlapping reads is a nontrivial task and requires complex computational techniques to produce a quality result. This technique was first described by Sanger et al. (*7*) and has been refined and used as the basis of genome sequencing and assembly ever since. The method has been developed in two main directions: (1) a whole genome shotgun approach (*7*, *8*) and (2) a hierarchical shotgun approach (*9*).

As described above, the whole genome approach where the genome is fragmented into defined length reads is followed by assembly, using purely bioinformatic-based techniques. The second approach, which is more appropriate for larger genomes, utilizes an added step to reduce the computational requirement in assembling the final sequence (**Fig. 1**b). Firstly, the genome is broken into larger fragments, which are in a known order; these fragments are then subsequently subjected to sequencing using the normal shotgun approach. This method requires less computational intervention in assembling the reads into the correct or-

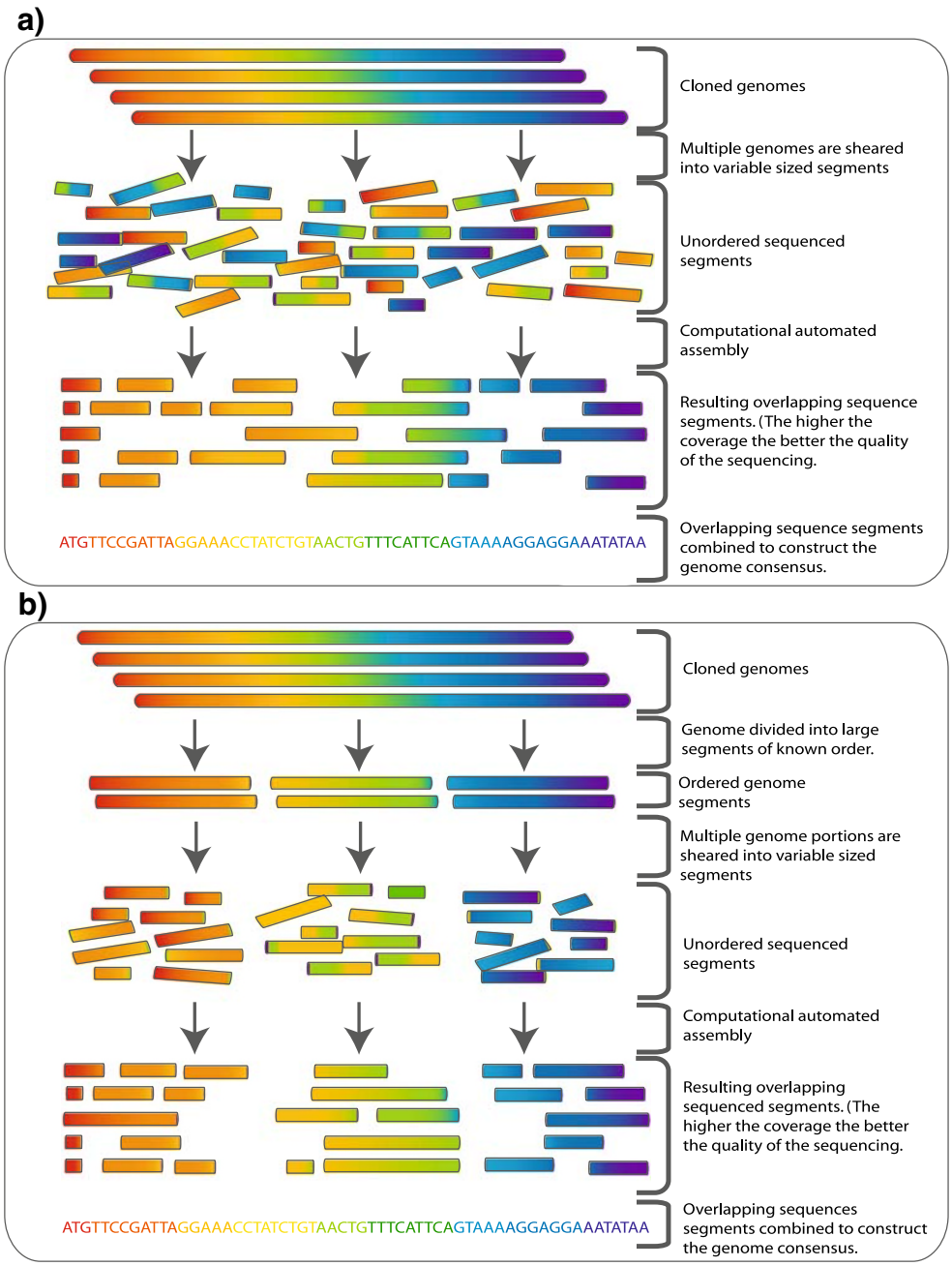

Fig 1. **a** Whole genome shotgun sequencing: Genome is sheared into small approximately equal sized fragments which are subsequently small enough to be sequenced in both directions followed by cloning. The cloned sequences (reads) are then fed to an assembler (illustrated in Fig. 2). **b** To overcome some of the complexity of normal shotgun sequencing of large sequences such as genomes a hierarchical approach can be taken. The genome is broken into a series of large equal segments of known order which are then subject to shotgun sequencing. The assembly process here is simpler and less computationally expensive.

der. Information is already known about the order of each subset of reads and thus less error is incurred in the final assembly. Of course, there are disadvantages with each of these approaches. For instance, with the whole-genome approach, there is the uncertainty as to whether the assembly is correct due to the total reliance on bioinformatics tools to join and order the reads; in addition, coverage may be insufficient (i.e., overlap between the fragments). The second approach is time consuming and labor intensive due to the addition of the extra step at the beginning of the protocol (*10*); this approach is also susceptible to incomplete coverage (*11*). Further advances have been made since the advent of shotgun sequencing but the central concepts remain the same.

Technologies currently used in genome sequencing include high-throughput methods such as 454 (*12*), SOLid (Applied Biosciences), and Solexa (*13*). These methods differ from older technologies in their throughput. Hundreds of thousands of DNA molecules at the same time are sequenced instead of a single DNA clones being processed (*14*). The reads returned from each of these technologies are very short; thus, assembly is rather difficult. This disadvantage is offset by the fact that some much DNA is sequenced. The sequencing methodology of these approaches, in particular 454, is called pyrosequencing. This essentially is the sequencing of DNA utilizing the detection of enzymatic activity to identify the bases. This process is termed "sequencing-by-synthesis" (*15*). Future developments will of course increase the length of reads produced by the technologies, as well as the accuracy of the programs with which the fragments are assembled.

Discussion in the past has provided some insight into the pitfalls of each method and perhaps aided in the decision making process (*14*, *16*, *17*). One thing is certain, the higher the coverage the method is able to achieve, the higher the likelihood that the assembly tool will get the correct result and so that in itself should be one of the highest considerations in the decision making process.

4. Base Calling and Genome Assembly

After genome sequencing is complete, it then becomes necessary to reconstruct the sequence fragments into a meaningful order that will accurately reflect the original orientation and order of the gene and junk (noncoding regions and pseudogenes) content. The most common and popular manner in which this is achieved is through the Phred (*18*, *19*)–PHRAP (*20*)–CONSED (*21*) pipeline of tools

(all of which originate from the University of Washington).

When assembling sequences from the myriad of reads that encompass a genome, several factors must be accounted for. Firstly, base-calling (the operation of determining the nucleotide base sequence from the chromatograph) must be completed with a minimum of erroneous interpretations of the chromatograph. The nucleotide sequence is determined for each read by the base-caller; the assembler then is utilized to piece the reads together into their original order, but must account for insertions, deletions, rearrangements, inversions, and sequence divergence in doing so. In particular, these events are important when assembling using a comparative method (i.e., using the scaffold of an existing genome to predict the locations of the fragments in the newly sequenced genome). No assembler (to date) proposes to handle all of these complications successfully but some do claim to be more capable than others under certain circumstances. For example, Pop et al. (*22*) reported that PHRAP (*20*) is more adept at creating long contigs (collection of contiguous pieces of DNA (reads)) than other available methods such as TIGR Assembler (*23*) or Celera Assembler (WGS-Assembler) (*24*). This can be valuable and has been used in the past as an indication of the success of an assembler. More recently, it has been reported that a reduction in the length of contigs across the assembly is an acceptable outcome if the error rate is reduced (*25*). Probably the most widely used base-calling algorithm is implemented in Phred (*18*, *19*). Others include GeneObject (*26*) and Life-Trace (*27*).

PHRAP has been widely adopted as an integral component of assembly pipelines such as implemented by Havlak et al. (*28*) in the Atlas Genome Assembly System and Mullikin and Ning (*29*) in the Phusion Assembler. It is considered the standard way in which to assemble smaller genomes with larger genomes relying on more complex algorithms provided by programs such as the WGS-Assembler.

Traditionally, assembly algorithms employ a method known as "overlap–layout–consensus" (*30*) (**Fig. 2**). Initially, the reads are compared to one another to identify overlapping regions using a strategy known as hashing to minimize the time required to complete the computation (*31*). When the potentially overlapping reads are positioned, a computationally intensive multiple sequence alignment is carried out to produce a consensus sequence. This consensus sequence is a draft of the genome and requires further computational and manual intervention to reach completion. In some genome assembly pipelines, a further step is introduced, in which information from sequencing in both directions of each fragment is utilized to reconstruct contigs into larger sections. These sections combine to create a scaffold, minimizing the amount of potential misassemble that may be introduced. Newer methods such as described by Pop et al. (*31*)

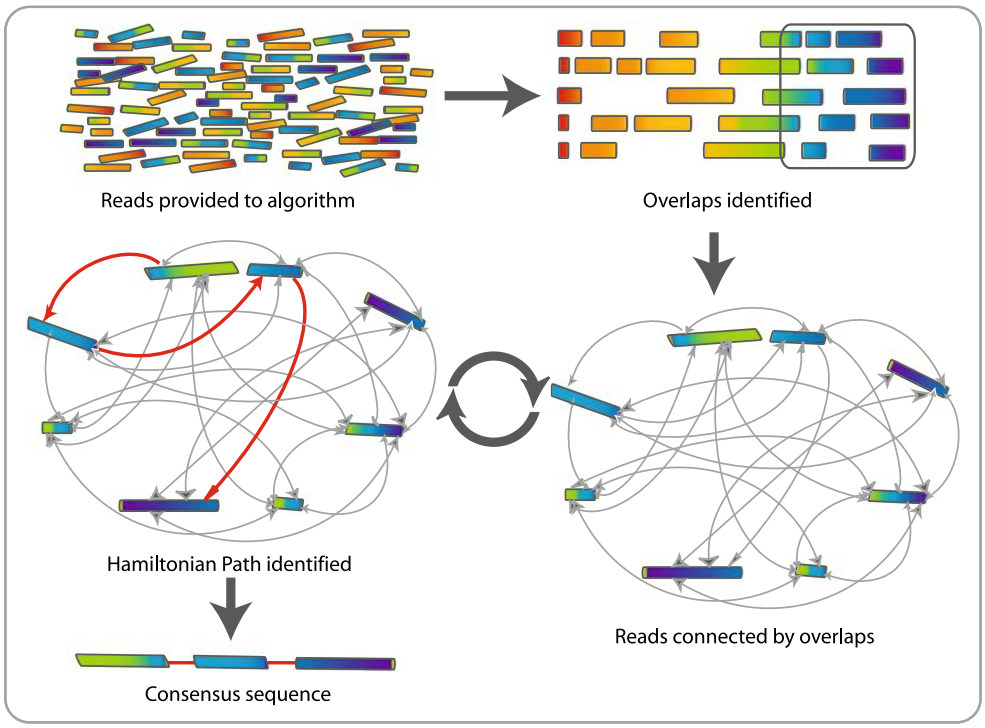

Fig 2. Overlap–layout–consensus genome assembly algorithm: Reads are provided to the algorithm. Overlapping regions are identified. Each read is graphed as a node and the overlaps are represented as edges joining the two nodes involved. The algorithm determines the best path through the graph (Hamiltonian path). Redundant information (i.e., unused nodes and edges) is discarded. This process is carried out multiple times and resulting sequences are combined to give the final consensus sequence that represents the genome.

eliminate the overlap identification step in favor of moving directly to the creation of the multiple sequence alignment, thus reducing the amount of time required to construct a draft assembly considerably. These methods have been entitled "alignment layout consensus" and are implemented in the AMOS Comparative Assembler (AMOS-Cmp). AMOS takes advantage of already available programs in its creation of multiple sequence alignments and scaffolds. Bambus (*32*) is designed to create scaffolds based on the discrete reads resulting from the shearing process of the shotgun technique. It aids in the resolution of the placement and direction of the reads using the mate-pair information produced by sequencing each read in both directions (a process known as double-ended shotgun sequencing). Using this scaffolding approach interleaved with other assembly techniques gives an elevated probability of producing a high quality complete genome.

There is no up-to-date objective comparison of genome assemblers available that takes the consistent development being

carried out on each project into account. Comparisons carried out by groups such as Huang and Madan (*33*) and Chen and Skiena (*34*) are works that seek to validate recently released methods. Chen and Skiena (*34*) come closest to an objective comparison in their rigorous testing of their own creation, STROLL, and latest versions (at the time) of PHRAP by Green (*20*) and the TIGR Assembler by Sutton et al. (*23*). In their evaluation of the programs, they reported that PHRAP was consistently more accurate in producing the correct assembly and had the lowest error rates of the group. STROLL produced similar results to PHRAP while TIGR Assembler produced a considerably more erroneous resultant assembly. The TIGR Assembler produced significantly more and smaller contigs, a higher proportion of gaps remaining unclosed and aside from the result, the process of running the TIGR Assembler on the read data used took approximately five times longer to complete than either of the other two programs evaluated.

In the race to publish the Human genome in the early 2000s, the Celera Whole Genome Assembler was engineered to accommodate large genomes. Its first use was described by (*24*) in the paper reporting the completion of the *Drosophila* genome (Myers et al.). This was enhanced and used later in the initial assembly of the Human genome (*35*) and the publication of the whole human genome assembly (*36*) in addition to the mouse (*37*), dog (*38*), and mosquito (*39*) genomes. While Celera is a private corporation, it has released the Celera Assembler as open source software for free usage.

In early 2007, a new assembly algorithm was described by Sommer et al. (*25*). It is a streamlined approach aimed at providing a simple, faster, and more efficient means of assembling fragmented sequences. Minimus (*25*) performs its best on small assembly jobs such as small genomes, genes, and bacterial artificial chromosome clones (*40*). It has also been assessed with respect to assembling larger sets of fragmented DNA such as those found in bacterial genomes and has been found to produce fewer assembly errors than PHRAP. The cost of this reduction in error rate is that the number of contigs is greater and consequently, the size of the contigs is smaller, resulting in a more fragmented assembly (*25*). In addition, all test assemblies produced by Minimus were completed in approximately half the time that PHRAP used. It remains to be seen whether this new assembler will work its way into common use in assembly systems such as Phusion and Atlas, but it is unlikely to remain at an advantage for long as the development and advancements of new and reworked as assemblers is swift and continuous. It has been suggested that it is beneficial for more than one method to be used, so that the exclusive advantages of each method may be exploited (*33*). This strategy may well of

course be more time consuming but if this time is affordable, it should be implemented.

5. Annotating the Genome

Distilling information from the assembled genome is the next obvious step in the process of building biological understanding of each newly sequenced individual or species. Genome annotation has three main levels—nucleotide-level annotation, protein-level annotation, and process-level annotation. The DNA level annotation process itself has several procedures associated with it. The first procedure is called Mapping, which is the process of identifying known genes, markers, and landmarks within the genome. This is usually carried out using sequence similarity searching programs such as BLAST (*41*). Secondly, Gene Finding as the name indicated involves the prediction of gene locations within the genome. Within the genes, the location of introns and exons are sought out in an effort to characterize the DNA into coding and junk categories. This is not a trivial process and often result in very poor sensitivity and specificity, in particular, results are poor when the signal-to-noise ratio is low, i.e., the amount of noncoding DNA is high (for a more elaborate review and comparison of gene prediction algorithms, *see* (*42*)).

Due to the extraordinary numbers of genes and sequences that have already been characterized in one species or another, a lot of the effort required to identify genes is cut out. Also to be identified are noncoding regions including, for example, tRNAs and rRNAs. These are mostly characterized by means of once again similarity searches and by using programs such as tRNAScan-SE (*43*). Other regions that must be discovered are regulatory regions, such as transcription factor binding sites, the topic of which is covered in detail in a review paper (*44*). In brief, methods have been developed to identify these regions by looking for patterns that occur more often that would be expected by chance; often this strategy is carried out in conjunction with similarity searching techniques.

At the protein-level annotation step, characterization is carried out. Genes are named and assigned functions mostly by means of comparison to already annotated genomes. Often this results in the categorization of many proteins into "unknown function" or "hypothetical protein" categories until experimentation provide light on the purpose of the gene at hand.

The final level of annotation is Process. Here, the biological processes affected by the gene are identified. Process categories usually include cell cycle, cell death, immune response, metabolism, etc. to name but a few. Once again, the processes affected

are usually determined via comparison with the information that is already available. It is useful here to note the existence of a few well-established databases that have devised naming conventions and controlled vocabulary for the description of new genes. Probably, the most commonly utilized of these are Panther (*45*–*48*) and GO (*49*, *50*). Both of these are freely available for use via the World Wide Web and are widely accepted adhered to by the genome analysis community.

Much work has been done in the development of quicker and more reliable ways of dealing with and identifying the protein coding regions of a genome at the same time noncoding regions while not completely neglected have been lesser studied of the two. Neither the detection of coding or noncoding regions is easy nor is the development of reliable and robust methods nearing a plateau. Constant progress is being made in these field; thus, the literature should be watched closely in order to be up to date with the current best practices in annotation.

6. Closing the Genome

Closing and completing a genome-sequencing project has proved to be an important step in ensuring the accuracy and reliability of the output into public databases. While the release of draft sequences is very useful, they are notoriously erroneous—in sequence and assembly (*17*). Error rates for draft sequencing have been reported to be 1 in 1,000–2,000 base pairs (*51*), in contrast to the rates of 1 in 10,000 reported by Selkov et al. (*51*) and 1 in 100,000 reported by Fleischmann (*52*) for whole genome sequencing. The typical errors found in draft sequences are sequencing errors, sequence misassembles, and the inclusion on contaminant sequences from foreign DNA as bona fide reads (*17*). Finding the source of such problems is difficult and time consuming and is often carried out manually. The most important factor taken into account here is the economic tradeoff and whether it is worth the compromise. For example, are there enough financial resources to allow for the whole genome sequencing to be brought to a close? It is important to realize that the quality of the sequencing or lack thereof will propagate forward into whatever analysis is carried out using the DNA sequence. Negative effects will be evident in all downstream analysis; everything from annotation and gene recognition to subsequent identification of homologs, gene families, and phylogenetics relationships will be affected.

While the discussed methods of sequence assembly are thorough and have relatively low error rates, they are not capable of

producing a completely reconstructed genome sequence without manual intervention and some potential resequencing. What the methods do produce is a draft sequence that would normally cover approximately 99% of the genome under reconstruction (*17*). This draft stage of assembly can be reached within a short number of days. In contrast, the process of closing the assembly out may potentially require months to complete and in some instances may take years. For example, the draft human genome was published in 2001, 4 years ahead of the predicted date of availability (2005). The complete whole genome was, however, not finished until 2003 and subsequently published in 2004 (*36*). The time and consequently the monetary cost incurred is a sacrifice that those in the area of comparative genomics are willing to make, as the quality provided by a closed genome is well worth the wait. Moreover, while useful in their own right, draft assemblies are constantly changing and potentially erroneous.

To meet the need for high quality complete genome sequences, several strategies have been developed at facilities such as TIGR, Washington University and Sanger. In some cases, a certain amount of error checking is carried out in conjunction with assembly. Programs such as EULER (*53*) and Arachne (*54*) are examples of assembly systems that include error correction components. Other approaches include the use of correction algorithms a posteriori to the assembly process. Examples of this type of program are Autofinish (of the wider package—CONSED) (*55*), MisEd (*56*), and ReDit (*57*). Autofinish, one of the most popular computer programs, is used in many genome sequencing centers, such as The Genome Center at the University of Washington, the Berkeley Drosophila Genome Project at Lawrence Berkeley National Laboratory, and the Lita Annenberg Hazen Genome Center at Cold Spring Harbor Laboratory among others (*55*). The product of the program must be manually inspected to ensure the quality and accuracy, but the amount of human intervention in this program is significantly reduced. In projects that had sequence coverage as low as four and five times, the human time required to close the project was reduced by more than 51% and 83%, respectively (*55*). As the sequence coverage increased up to 14 times, the difference diminished, but consistently less human effort was required when Autofinish was utilized.

The finishing techniques that are employed in programs such as Autofinish reflect what a human finisher does in identifying problem areas in the assembly that has been produced. They go on to propose possible means of resolving the issues, indicating regions to be resequenced and potential reads to aid in closing any gaps that are present. Due to the nature of the problems that are found in draft genome sequences, the process of finishing is an iterative process that can require many cycles through a workflow to resolve all issues; frequently, it is necessary for a human finisher

to get involved toward the end to complete the process. This intervention must be as efficient as possible and many graphical viewers and editors are available for this purpose. Examples of manual finishing software are components of the aforementioned CONSED: sequence finishing tool (*21*) and ReDit: shotgun assembly finishing aid (*57*), also others include BaCCardI: validate and assist in finishing (*58*) and DNPTrapper: analysis of complex regions and finishing tool (*59*). Each of these software programs aim to make the editing process as user friendly as possible while offering the best possible combinations of editing capabilities.

7. Comparative Genomics: Solving the Puzzle

Comparative genomics is one of the most promising areas that logically follows the success in improving genome sequencing. More and more comparative genomics programs are being demanded to identify protein-coding genome regions, placement of regulatory elements, and the main evolutionary dynamics affecting the complexity of genome organization. Despite its apparent simplicity, such comparative methods have to face many technical as well as theoretical problems. One of the most important problems is aligning whole genomes and visualizing such alignments in a comprehensive and comprehensible way. This problem in sequence alignment leads to other genomic problems such as the finding of orthologs between genomes. The magnitude of this problem becomes increasingly magnified when the comparison is held between genomes with different population dynamics and hence different mutational rates, as we will explain below.

7.1. The First Hurdle—How to Determine the Homologs (Orthologs and/or Paralogs)?

Identification of homologous genes relies on the appropriate definition of a homolog. The most widely accepted definition is that homologous genes share a common ancestry. This definition, however, is not precise as to the nature of this common ancestry and comprises two types of homologs (as described by Fitch (*60*) and Fitch and Margoliach (*61*)): orthologs (common species ancestry caused by speciation event in such away that the homolog genes are in different species) and paralogs (common gene ancestry caused by a gene duplication event and, as a consequence, the homologous genes are present in the same species).

Irrespective of the nature of the ancestry considered, homologs are usually identified on the basis of sequence similarity. So the higher the similarity, the more likely it is that the sequences have derived form a common ancestor. One of the first and the most commonly used software to detect the degree of similarity

between sequences is BLAST (*62*) and the newer version PSI-BLAST (*63*). BLAST uses predefined scoring matrices in comparison to position-specific scoring matrices derived from the scoring hits in the initial search in PSI-BLAST. The two programs yield information about the score for the comparisons and their likelihood, called the *e*-value. Sequences with the highest scores and therefore with the lowest *e*-values are considered to be the closest relatives in the searched database. The assumption underlying this software is that the phylogenetic relationship between any two sequences and their degree of similarity are positively correlated. This, however, leads to another theoretical problem: how to determine if a sequence is more similar to a different particular sequence than it is to another. Unfortunately, setting a statistical cutoff value to determine when two sequences are significantly similar is rather difficult and problematic when determining a set of possible homologs. The lower the cut off, the larger the number of false negatives. On the other hand, the higher the cut off, the larger the number of false positives. As an additional drawback, the sequences with the highest score and lowest *e*-value are not always more closely related to each other than those identified as hits with a lower score (*64*).

In the BLAST searches for homologs, many types of relationships between the homologs can be investigated, including hits of many-to-many, one-to-many, or very strict one-to-one relationships. The first two are a result of duplication events after speciation. A very effective way to identify one-to-one relationship is by performing the generally called reciprocal best BLAST hits (*65*, *66*). This method is based on the assumption that genes that are each other's best hits when performing a BLAST search are more likely to be orthologs compared to ones that are not. The reason for this is that although gene A in genome 1 may be the best match for gene B in genome 2, this match may be worse than gene B in genome 2 with gene C in genome 1. This approach is again limited by the problem of the assumption that best hits ensure orthology, which might not be the case when a particular gene underwent a recent duplication in a particular lineage. The consequence of this is that when a gene finds a paralog as top BLAST hit instead of its ortholog, both the gene and its paralog are excluded from downstream analyses (*67*). These limitations in the BLAST searches have fuelled the development of other ways to identify putative orthologs over the last few years. One of such methods uses the sequence distances instead of similarities to identify orthologs and uses the reciprocal smallest distance algorithm (*67*). It uses global sequence alignment and maximum likelihood to estimate the evolutionary distances between genes to detect orthologous genes. This approach have also been used to determine orthologs in databases like Roundup (*68*). Another simple approach that has contributed significantly to the reduction in the number of false

positive results when conducting BLAST searches is PSI-BLAST (*69*).

Homology may also be ascertained by means of phylogenetic methods such as BranchClust by (*70*). This type of method is capable of determining homology distinguish it from paralogy. BranchClust utilizes similarity searching during the execution of its algorithm but obviously does not rely solely on it. Hits within a certain threshold are used rather than the best hit in order to include paralogs and orthologs. These results are then grouped into what Poptsova and Gogarten has termed superfamilies. These sequences are aligned and phylogenetic trees are constructed. The step of phylogenetic inference is then followed by a complex algorithm that is described fully in the application's article (*70*). The outcome of using this method over more traditional one is that BranchClust is reported to outperform similarity search methods due to its lower false negative rate than the reciprocal best blast hit method.

Irrespective of the method used to identify homologs, visualizing results is a common way to inspect and yield the first insights into trends and patterns when looking at genome evolutionary dynamics. This fact has inspired the creation of software for comparative genomics with graphical solutions to assist in the interpretation of the results. These solutions provide user-friendly environments in which navigation along alignments, etc. is easy and reliable. The question remains, however, whether visualization tools can solve the puzzle of genome rearrangements. An argument against the use of techniques such as this is that the process will not be repeatable or statistically sound. Undoubtedly, insights will be yielded but all sure perceived trends should be investigated in a more analytically robust manner.

7.2. Pairwise Genome Comparisons

Many groups have devoted a substantial amount of their resources to the development of tools aimed at comparing two genomes and have validated such tools by comparing circular prokaryotic genomes. Some visualization software tools have specialized in performing direct comparisons of synteny information through scatter plots of pairwise genome comparisons. For example, software such as DAGchainer (*71*), GeneOrder (*72*), GenePlot from NCBI (*73*), Genome v/s Genome Protein Hits Scatter Plot from The Comprehensive Microbial Resource (CMR) (*74*), and GenomePlot from PLATCOM (*75*) achieve this by presenting a plot where one axis represents the positions of the genes within one of the genomes while the other represents the genes for the other genome (**Fig. 3**). The scatter plot then represents homologous genes for both genomes determined by either total hits or best BLAST hit. Perfectly syntenic genes between the two genomes would therefore represent a linear relationship between the two axes (**Fig. 3**a) whereas alternative arrangements of the scattered dots may indicate that genome

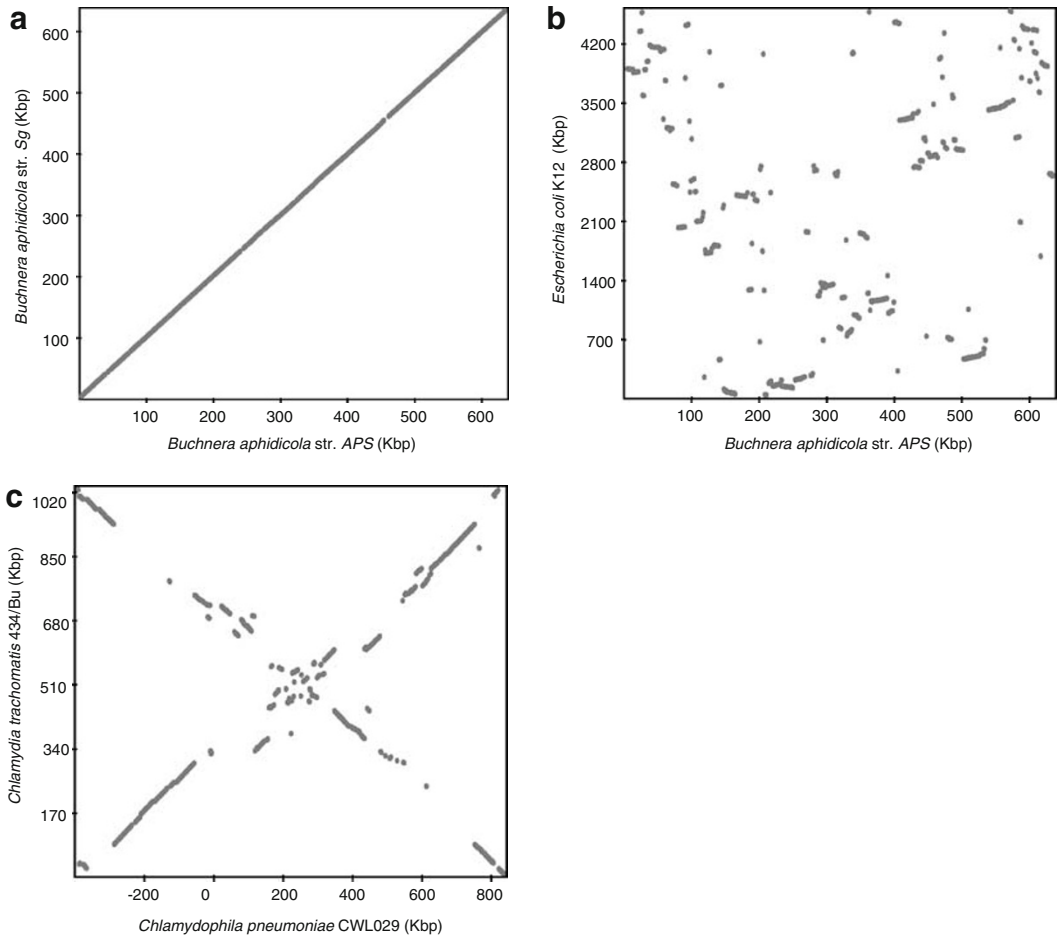

Fig 3. Genome rearrangements plots comparing two genomes. Genome plots can provide information on the kind of rearrangements undergone. These plots represent the location of each gene in one axis for one of the genomes against the location of the found ortholog in the other axis for the second genome. **a** Comparative genomic plot when comparing two genomes showing no lineage-specific genome rearrangements. In this case, the plot was produced for the comparison of two primary symbiotic bacteria of insects (*B. aphidicola* strain *A. pisum* versus *B. aphidicola* strain *Schizaphis graminum*). Since no rearrangements have occurred in any of the two genomes, the comparison yields a straight diagonal line. **b** Comparative genomic plot for two genomes showing lineage-specific genome rearrangements. In this case, the plot was comparing the genome of other patterns that can be observed and are *x*-like patterns **b** (in this case, *B. aphidicola*, *A. pisum*, and *E. coli* k12) where the rearrangements have occurred over the replication axis *E. coli* K12 to the genome of *B. aphidicola* strain *A. pisum*. **c** This is the comparison between *Chlamydophila pneumoniae* CWL029 and *Chlamydia trachomatis* 434/Bu that show an even better example of rearrangements that have occurred over the replication axis (this example have also been shown in (*102*)). As shown, many rearrangements including inversions and translocations have occurred, and consequently, the orthologs are not located in the major diagonal of the plot but rather show an *X*-shape distribution. This is expected if an inversion has taken place near the centromer of the chromosome.

rearrangements have taken place in one of the genomes (**Fig. 3**b). As an alternative to these visual representations, other programs such as GRAST mark the hits between the two genomes and represent them in a circular way (*76*). Finally, other programs such as ACGT (*77*), GOAL from BROP (*78*), BugView (*79*), and GenomeComp (*80*) have contributed to the field of comparative genomics by linearly representing rearrangements or syntenic information by linking homologous regions between the compared genomes using lines. The advantage of programs such as these is that in addition to yielding information about genome rearrangements, they can also spot conserved and nonconserved regions between the two genomes in much greater detail than other programs.

Aside from the syntenic analyses using visualization tools, other programs have been developed to search for other types of information in comparative genomics. For example, GC Comparison Graph from The CMR (*74*) compares the GC content between two genomes by placing orthologs in the axis according to their GC content, highlighting GC compositional shifts at the genome level between two genomes. Although useful in their content, these programs are subject to several drawbacks from the pragmatic point of view among which the most important is the impossibility to perform multiple genome comparisons and hence to establish the ancestry of genome rearrangement dynamics.

7.3. Multiple Genome Comparisons

As the number of genomes increased over the last decade, the demand for an understanding of the dynamics of genome evolution also increased. Dealing with the complexity of multiple genomes comparisons has been halted by the unparalleled development of appropriate software tools. Nowadays, several software tools have been developed. An example of a multiple genome comparison tool is GenColors from Jena Prokaryotic Genome Viewer (JPGV) (*81*). This program allows the user to display a number of features on the genome, like CDS, RNA genes, tRNA genes, rRNA genes, Mics RNA, GC contents, GC skew Keto excess, etc. This database also represents genomes in either a circular diagram or in a linear plot. Although several genomes can be examined at the same time using this tool, these are human observations of the genomes rather than real phylogenetic studies of the genome properties. JPGV allows multiple genome comparisons by determining a core gene set of two or more genomes defined by the set of best-bidirectional hits for all possible pairs of genes. Other methods of the JPGV are implemented to perform pair wise comparisons only.

In addition, there are computational tools that compare multicircular prokaryotic genomes and present their similarities in a circular diagram. Some of these tools perform these comparisons in addition to the BLAST searches and the CGView server is an example of that (*82*). Others also display information about the

percentage of GC for each one of the genomes, such is the case of GenomeViz (*83*).

To gain more information about genome rearrangements and inversions, there has been a great effort in developing tools that perform linear comparisons between genomes. The way these tools compare genomes is by performing genomes alignments where possible and then by conducting multiple genome comparisons. There are many different multiple genome alignments algorithms. The first type is based on defining a reference genome and performing alignments taking into account that reference genome. This type of alignment algorithm is implemented in a program called Vista (*84*). The second approach is that where an iterative pairwise alignment is performed under the control of a guide tree. The tree defines the order in which the genomes should be added to the alignment. The third type of algorithms determines anchors present in all genomes and then proceeds to align them. Once aligned, the last step is to close the gaps between the anchors by aligning the substrings between them. Examples of programs implementing this type of algorithm are MGA (*85*), M-GCAT (*86*), and Mauve (*87*), with each of them having their own algorithm for identifying the anchors and performing the alignment of the interanchor regions afterward.

There are other tools that allow the user to do other things in addition to the alignment of genomes. For example, MANTIS (*88*) is a phylogenetic-group specific (metazoan phylogeny) tool that analyzes the patterns of gene gains and losses at specific branches of the phylogeny. Then, the program infers the gene content of the ancestral genome to the clade and identifies over- or underrepresentation of certain processes among the class of gene gains or losses.

Despite all these effort in developing more robust and accurate methods to perform comparative genomic studies, several biological phenomena pose difficulties in identifying the real genome dynamic processes in organisms. For example, genome duplication, genome shrinkage in intracellular symbiotic bacteria, and lateral gene transfer may well hide the real genome rearrangement processes undergone in particular genomes. To illustrate the importance of the biology of the organismal biology to understand genome dynamics, we will focus the rest of the review on intracellular bacterial genomes.

8. Comparative Genomics of Intracellular Bacteria

Intracellular bacteria are a special group of organisms that have been able to adapt to intracellular life, establishing either a symbi-

otic or pathogenic relationship with the host. Because many of the genes that were important for the free lifestyle are no longer needed by these bacteria, they underwent nonfunctionalization followed by disintegration (*89*). This process has been enhanced by the fact that the host provides these bacteria by some of their needed components and by a chemically stable rich environment. Genome shrinkage is therefore a fact in most if not all the strict intracellular bacteria and this process has been mostly accompanied by genome rearrangements and fast evolutionary rates of proteins. Because of these intracellular associated genomic and evolutionary events, comparative genomics including identification of orthologs, paralogs, synteny analyses, and others pose great challenge in the comparison with free-living bacteria and require including biological information in the comparative genomics analyses to increase the accuracy of the results.

In the case of the symbiotic relationships, the difficulty of comparative genomics acquires another dimension and complexity specifically associated to the mutational dynamics of these organisms. There are two main groups of symbiotic bacteria: the facultative and the obligated. When the association is facultative, it implies that the survival of each partner can be possible without the other under special environmental conditions. This is for example seen between the pea aphid *Acyrthosiphon pisum* and the facultative endosymbionts *Hamiltonella defensa* that acts as a protector of the aphid against parasitism by the solitary endoparasitoids *Aphidium ervi* and *Aphidius eadyi* (*90–92*). The other case, obligated, is when the relationship between the two organisms becomes so close that the host's relative biological fitness would become seriously compromised if deprived of the symbiont. This is the case of the symbiotic relationship between the bacterium *Buchnera aphidicola* sp. and the aphid insect (*93*) and it is an example where the host (the aphid) has evolved specialized cells to house its endosymbionts (so called bacteriocytes) (*94*). This relationship is one of the best characterized in the literature so the last following part of this review will focus on endosymbionts contained in bacteriocyte and the challenges that their mutational dynamics impose in the comparative genomics of bacteria.

8.1. Genome Evolution of Intracellular Bacteria

The clonal vertical transmission of small populations in many intracellular symbiotic bacteria and pathogens to the next host generations imposes a strong bottleneck on the effective population size of these bacteria. This results in relaxed selective constraints in the symbiotic genomes and their channeling into a dynamic of neutral fixation of slightly deleterious mutations and irreversible increase in the endosymbiont genome mutational load (a phenomenon named Muller's ratchet (*95*)). However, these bacteria are also subjected to selection imposed mostly over their insect hosts. Because of their clonal transmission and their confinement to the

interior of bacteriocytes symbiotic bacteria have little or no opportunity for recombination and hence have no alternative means for the removal of these slightly deleterious mutations.

Is there a minimum set of genes necessary for the maintenance of intracellular life? Numerous scientists have addressed this question and many have been attempting to answer it through the study of the smallest endosymbiotic genome (*96*). Comparative genomics studies in a large number of organisms have shown that the minimal gene content will depend on the environmental conditions the organism lives under (*97*, *98*).

The process of gene loss in intracellular organisms has an important effect on rewiring the functional relationships among genes. This would lead to different organisms containing different genes performing the same essential functions in the cell. So when looking at gene content of intracellular bacteria, we should talk about the functional group of genes instead of individual genes (*99*).

8.2. Difficulties with Comparative Genomics of Bacteriocyte-Housed Insect Endosymbionts

Comparison of bacterial genomes may provide clues about the main genome rearrangement dynamics supporting different lifestyles, for example, comparative genomics of intracellular symbiotic bacteria and their closest free-living relatives. Performing comparative genomics on bacteria that are in an intermediate stage between free-living and host specific symbiosis (primary endosymbionts) with each of their groups could shed some light on the establishment of symbiosis itself. These bacteria are the ones we refer to as secondary endosymbionts—they are distinguishable from primary symbionts by their larger genomes and the fact that they are not living under the protection of the bacteriocytes provided by their hosts.

As a consequence of Muller's ratchet in intracellular bacteria in combination with mutational bias, their genomes present a higher AT content than observed in their free-living relatives (*100*). This results in programs like BLAST having increased difficulty in determining homologs—especially between the intracellular bacteria and their free-living relatives.

The difficulty of doing comparative genomics with intracellular bacteria is that few to none of the software programs have been designed to deal with any of the theoretical problems seen in these organisms. Most software and methods have been directed toward the broad stream of the comparison of genomes with similar sizes and belonging to bacteria with minor differences regarding their lifestyle or environmental conditions. The challenge, however, resides on identifying important genomic dynamics that occurred during the transition between two lifestyles and hence between potentially different biological systems.

One of the biggest problems with the comparative genomics of endosymbiontic and pathogenic bacteria to their closest free-living relative bacteria is the different evolutionary force under which they evolve. Because the population sizes of endocellular

symbiotic bacteria undergo strong bottleneck during the intergenerational transmissions, many of the stochastically produced amino acid mutations are fixed by genetic drift despite their slight deleterious effects. This implies that the mean mutational load in the endocellular bacteria will dramatically increase posing serious difficulties to find their orthologs in free-living bacteria. Comparing endosymbionts with each other can yield valuable information about endosymbiosis but it is crucial to compare the endosymbionts to free-living bacteria to be able to investigate the transition from free-living to intracellular lifestyle and predict the shift in evolutionary forces. Novel methods are hence required to account for the biological and population genetics differences of the organisms whose genomes are being compared.

8.3. Databases and Methods for the Analysis of Endosymbionts

BuchneraBASE (*101*) is a database that contains information on *Buchnera* sp. APS. This database is the only of its kinds, to our knowledge, devoted completely to a primary symbiont. It does not offer any direct comparative genome tool for the user like many other databases but it contains some data obtained from comparison between symbiotic gamma-proteobacteria and an in silico model of *Escherichia coli*. This database was built as to integrate new sequenced genomes from symbiotic bacteria as they became available. It performs comparisons between different genomes using the information of gene orthology. The database also has a summary page that shows two user-interactive tables. The first table represents the number of genes in each of the genomes that are in a certain category, i.e., total number of complete genes, total number of pseudogenes, genes with an *E. coli* ortholog shared with the endosymbiont of *Wigglesworthia glossinidia* or not shared with *Wigglesworthia*, etc. The second table can be used to browse through each of the functional classifications for each of the symbionts stored in the database.

To our knowledge, there is only one program, GRAST (*76*), that has been developed with the sole purpose to investigate the evolutionary dynamics of endosymbionts. It performs a pairwise comparison between a free-living (reference genome) genome and an endosymbiotic genome and allows the user to choose between different outputs options, providing valuable insights regarding the change in genome dynamics in comparison to their free-living relatives. The outputs range from generation of genome plots with orthologous and nonorthologous genes' sets are plotted in the two genomes being compared to plots with the analysis of the distribution of genome rearrangements or dynamics in one of the genomes (Fig. 3). Among other types of information, the program yields information about conserved regions between the two genomes, distribution of percentage of differences in the number of genes present in the different functional categories between the two genomes being compared

and deviations from the expected percentage of orthologs between the genomes, and information about intergenic regions according to their position/rearrangement in the two genomes.

A brief look at the genome sizes of bacteria would suffice to realize about the incredible diversity of the genomic dynamic events that have been happening throughout evolution. These events are key to understand the different evolutionary processes shaping organismal organization. Intracellular organisms perform a minority of this diversity but they represent extreme cases where most of the genomic dynamics become dramatically manifested. New methods should therefore be developed to perform in-depth comparative genomic analyses of these bacteria to infer important shifts in the evolution of genomes.

The genomic era has exploded and generated new research avenues that go beyond all expectations. A plethora of novel ways of designing experiments and computational tools has been fuelled by the information generated from the first comparative genomics analyses. The challenge that remains is to design new comprehensive and accurate bioinformatics tools capable of counterbalancing our limitations to analyze the overwhelming amount of genomic data generated.

References

1. Rappe M. S., Giovannoni S. J. (2003) The uncultured microbial majority. *Annu Rev Microbiol* 57, 369–394.
2. Hugenholtz P. (2002) Exploring prokaryotic diversity in the genomic era. *Genome Biol* 3(2), REVIEWS0003.
3. Chen K., Pachter L. (2005) Bioinformatics for whole-genome shotgun sequencing of microbial communities. *PLoS Comput Biol* 1(2), 106–112.
4. Tyson G. W., Chapman J., Hugenholtz P., Allen E. E., Ram R. J., Richardson P. M., Solovyev V. V., Rubin E. M., Rokhsar D. S., Banfield J. F. (2004) Community structure and metabolism through reconstruction of microbial genomes from the environment. *Nature* 428(6978), 37–43.
5. Tringe S. G., von Mering C., Kobayashi A., Salamov A. A., Chen K., Chang H. W., Podar M., Short J. M., Mathur E. J., Detter J. C., Bork P., Hugenholtz P., Rubin E. M. (2005) Comparative metagenomics of microbial communities. *Science* 308(5721), 554–557.
6. Riesenfeld C. S., Schloss P. D., Handelsman J. (2004) Metagenomics: genomic analysis of microbial communities. *Annu Rev Genet* 38, 525–552.
7. Sanger F., Nicklen S., Coulson A. R. (1977) DNA sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci USA* 74 (12), 5463–5467.
8. Edwards A., Voss H., Rice P., Civitello A., Stegemann J., Schwager C., Zimmermann J., Erfle H., Caskey C. T., Ansorge W. (1990) Automated DNA sequencing of the human HPRT locus. *Genomics* 6(4), 593–608.
9. Green P. (2002) Whole-genome disassembly. *Proc Natl Acad Sci USA* 99(7), 4143–4144.
10. Kaiser O., Bartels D., Bekel T., Goesmann A., Kespohl S., Puhler A., Meyer F. (2003) Whole genome shotgun sequencing guided by bioinformatics pipelines—an optimized approach for an established technique. *J Biotechnol* 106(2–3), 121–133.
11. Tauch A., Homann I., Mormann S., Ruberg S., Billault A., Bathe B., Brand S., Brockmann-Gretza O., Ruckert C., Schischka N., Wrenger C., Hoheisel J., Mockel B., Huthmacher K., Pfefferle W., Puhler A., Kalinowski J. (2002) Strategy to sequence the genome of Corynebacterium glutamicum ATCC 13032: use of a cosmid and a bacterial artificial chromosome library. *J Biotechnol* 95(1), 25–38.

12. Goldberg S. M. D., Johnson J., Busam D., Feldblyum T., Ferriera S., Friedman R., Halpern A., Khouri H., Kravitz S. A., Lauro F. M., Li K., Rogers Y. H., Strausberg R., Sutton G., Tallon L., Thomas T., Venter E., Frazier M., Venter J. C. (2006) A Sanger/pyrosequencing hybrid approach for the generation of high-quality draft assemblies of marine microbial genomes (vol 103., pg 11240., 2006). *Proc Natl Acad Sci USA* 103(43), 16057.
13. Potera C. (2006) New gene sequencer targets productivity—Solexa says its novel system offers better cost-effectiveness via use of short-read sequences. *Genet Eng News* 26(17), 10–+.
14. Graveley B. R. (2008) Molecular biology—power sequencing. *Nature* 453(7199), 1197–1198.
15. Wicker T., Schlagenhauf E., Graner A., Close T. J., Keller B., Stein N. (2006) 454 sequencing put to the test using the complex genome of barley. *BMC Genomics* 7, 275.
16. Branscomb E., Predki P. (2002) On the high value of low standards. *J Bacteriol* 184(23), 6406–6409; discussion 9.
17. Fraser C. M., Eisen J. A., Nelson K. E., Paulsen I. T., Salzberg S. L. (2002) The value of complete microbial genome sequencing (you get what you pay for). *J Bacteriol* 184(23), 6403–6405; discusion 5.
18. Ewing B., Hillier L., Wendl M. C., Green P. (1998) Base-calling of automated sequencer traces using phred. I. Accuracy assessment. *Genome Res* 8(3), 175–185.
19. Ewing B., Green P. (1998) Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res* 8(3), 186–194.
20. Green, P. (1994) PHRAP., unpublished. http://www.phrap.org/.
21. Gordon D., Abajian C., Green P. (1998) Consed: a graphical tool for sequence finishing. *Genome Res* 8(3), 195–202.
22. Pop M., Salzberg S. L., Shumway M. (2002) Genome sequence assembly: algorithms and issues. *Computer* 35(7), 47–54.
23. Sutton G., White O., Adams M. D., Kerlavage A. R. (1995) TIGR Assembler: a new tool for assembling large shotgun sequencing projects. *Genome Sci Technol* 1, 9–19.
24. Myers E. W., Sutton G. G., Delcher A. L., Dew I. M., Fasulo D. P., Flanigan M. J., Kravitz S. A., Mobarry C. M., Reinert K. H., Remington K. A., Anson E. L., Bolanos R. A., Chou H. H., Jordan C. M., Halpern A. L., Lonardi S., Beasley E. M., Brandon R. C., Chen L., Dunn P. J., Lai Z., Liang Y., Nusskern D. R., Zhan M., Zhang Q., Zheng X., Rubin G. M., Adams M. D., Venter J. C. (2000) A whole-genome assembly of Drosophila. *Science* 287(5461), 2196–2204.
25. Sommer D. D., Delcher A. L., Salzberg S. L., Pop M. (2007) Minimus: a fast., lightweight genome assembler. *BMC Bioinformatics* 8, 64.
26. Gilchrist, R., Chi, V., inventors (1999) Visible Genetics Inc.., assignee. GeneObject. USA patent 5916747.
27. Walther D., Bartha G., Morris M. (2001) Basecalling with LifeTrace. *Genome Res* 11(5), 875–888.
28. Havlak P., Chen R., Durbin K. J., Egan A., Ren Y., Song X. Z., Weinstock G. M., Gibbs R. A. (2004) The Atlas genome assembly system. *Genome Res* 14(4), 721–732.
29. Mullikin J. C., Ning Z. (2003) The phusion assembler. *Genome Res* 13(1), 81–90.
30. Peltola H., Soderlund H., Ukkonen E. (1984) SEQAID: a DNA sequence assembling program based on a mathematical model. *Nucleic Acids Res* 12(1 Pt 1), 307–321.
31. Pop M., Phillippy A., Delcher A. L., Salzberg S. L. (2004) Comparative genome assembly. *Brief Bioinform* 5(3), 237–248.
32. Pop M., Kosack D. S., Salzberg S. L. (2004) Hierarchical scaffolding with Bambus. *Genome Res* 14(1), 149–159.
33. Huang X., Madan A. (1999) CAP3: A DNA sequence assembly program. *Genome Res* 9(9), 868–877.
34. Chen T., Skiena S. S. (2000) A case study in genome-level fragment assembly. *Bioinformatics* 16(6), 494–500.
35. Venter J. C., Adams M. D., Myers E. W., Li P. W., Mural R. J., Sutton G. G., Smith H. O., Yandell M., Evans C. A., Holt R. A., Gocayne J. D., Amanatides P., Ballew R. M., Huson D. H., Wortman J. R., Zhang Q., Kodira C. D., Zheng X. H., Chen L., Skupski M., Subramanian G., Thomas P. D., Zhang J., Gabor Miklos G. L., Nelson C., Broder S., Clark A. G., Nadeau J., McKusick V. A., Zinder N., Levine A. J., Roberts R. J., Simon M., Slayman C., Hunkapiller M., Bolanos R., Delcher A., Dew I., Fasulo D., Flanigan M., Florea L., Halpern A., Hannenhalli S., Kravitz S., Levy S., Mobarry C., Reinert K., Remington K., Abu-Threideh J., Beasley E., Biddick K.,

Bonazzi V., Brandon R., Cargill M., Chandramouliswaran I., Charlab R., Chaturvedi K., Deng Z., Di Francesco V., Dunn P., Eilbeck K., Evangelista C., Gabrielian A. E., Gan W., Ge W., Gong F., Gu Z., Guan P., Heiman T. J., Higgins M. E., Ji R. R., Ke Z., Ketchum K. A., Lai Z., Lei Y., Li Z., Li J., Liang Y., Lin X., Lu F., Merkulov G. V., Milshina N., Moore H. M., Naik A. K., Narayan V. A., Neelam B., Nusskern D., Rusch D. B., Salzberg S., Shao W., Shue B., Sun J., Wang Z., Wang A., Wang X., Wang J., Wei M., Wides R., Xiao C., Yan C., Yao A., Ye J., Zhan M., Zhang W., Zhang H., Zhao Q., Zheng L., Zhong F., Zhong W., Zhu S., Zhao S., Gilbert D., Baumhueter S., Spier G., Carter C., Cravchik A., Woodage T., Ali F., An H., Awe A., Baldwin D., Baden H., Barnstead M., Barrow I., Beeson K., Busam D., Carver A., Center A., Cheng M. L., Curry L., Danaher S., Davenport L., Desilets R., Dietz S., Dodson K., Doup L., Ferriera S., Garg N., Gluecksmann A., Hart B., Haynes J., Haynes C., Heiner C., Hladun S., Hostin D., Houck J., Howland T., Ibegwam C., Johnson J., Kalush F., Kline L., Koduru S., Love A., Mann F., May D., McCawley S., McIntosh T., McMullen I., Moy M., Moy L., Murphy B., Nelson K., Pfannkoch C., Pratts E., Puri V., Qureshi H., Reardon M., Rodriguez R., Rogers Y. H., Romblad D., Ruhfel B., Scott R., Sitter C., Smallwood M., Stewart E., Strong R., Suh E., Thomas R., Tint N. N., Tse S., Vech C., Wang G., Wetter J., Williams S., Williams M., Windsor S., Winn-Deen E., Wolfe K., Zaveri J., Zaveri K., Abril J. F., Guigo R., Campbell M. J., Sjolander K. V., Karlak B., Kejariwal A., Mi H., Lazareva B., Hatton T., Narechania A., Diemer K., Muruganujan A., Guo N., Sato S., Bafna V., Istrail S., Lippert R., Schwartz R., Walenz B., Yooseph S., Allen D., Basu A., Baxendale J., Blick L., Caminha M., Carnes-Stine J., Caulk P., Chiang Y. H., Coyne M., Dahlke C., Mays A., Dombroski M., Donnelly M., Ely D., Esparham S., Fosler C., Gire H., Glanowski S., Glasser K., Glodek A., Gorokhov M., Graham K., Gropman B., Harris M., Heil J., Henderson S., Hoover J., Jennings D., Jordan C., Jordan J., Kasha J., Kagan L., Kraft C., Levitsky A., Lewis M., Liu X., Lopez J., Ma D., Majoros W., McDaniel J., Murphy S., Newman M., Nguyen T., Nguyen N., Nodell M., Pan S., Peck J., Peterson M., Rowe W., Sanders R., Scott J., Simpson M., Smith T., Sprague A., Stockwell T., Turner R., Venter E., Wang M., Wen M., Wu D., Wu M., Xia A., Zandieh A., Zhu X. (2001) The sequence of the human genome. *Science* 291(5507), 1304–1351.

36. Istrail S., Sutton G. G., Florea L., Halpern A. L., Mobarry C. M., Lippert R., Walenz B., Shatkay H., Dew I., Miller J. R., Flanigan M. J., Edwards N. J., Bolanos R., Fasulo D., Halldorsson B. V., Hannenhalli S., Turner R., Yooseph S., Lu F., Nusskern D. R., Shue B. C., Zheng X. H., Zhong F., Delcher A. L., Huson D. H., Kravitz S. A., Mouchard L., Reinert K., Remington K. A., Clark A. G., Waterman M. S., Eichler E. E., Adams M. D., Hunkapiller M. W., Myers E. W., Venter J. C. (2004) Whole-genome shotgun assembly and comparison of human genome assemblies. *Proc Natl Acad Sci USA* 101(7), 1916–1921.

37. Mural R. J., Adams M. D., Myers E. W., Smith H. O., Miklos G. L., Wides R., Halpern A., Li P. W., Sutton G. G., Nadeau J., Salzberg S. L., Holt R. A., Kodira C. D., Lu F., Chen L., Deng Z., Evangelista C. C., Gan W., Heiman T. J., Li J., Li Z., Merkulov G. V., Milshina N. V., Naik A. K., Qi R., Shue B. C., Wang A., Wang J., Wang X., Yan X., Ye J., Yooseph S., Zhao Q., Zheng L., Zhu S. C., Biddick K., Bolanos R., Delcher A. L., Dew I. M., Fasulo D., Flanigan M. J., Huson D. H., Kravitz S. A., Miller J. R., Mobarry C. M., Reinert K., Remington K. A., Zhang Q., Zheng X. H., Nusskern D. R., Lai Z., Lei Y., Zhong W., Yao A., Guan P., Ji R. R., Gu Z., Wang Z. Y., Zhong F., Xiao C., Chiang C. C., Yandell M., Wortman J. R., Amanatides P. G., Hladun S. L., Pratts E. C., Johnson J. E., Dodson K. L., Woodford K. J., Evans C. A., Gropman B., Rusch D. B., Venter E., Wang M., Smith T. J., Houck J. T., Tompkins D. E., Haynes C., Jacob D., Chin S. H., Allen D. R., Dahlke C. E., Sanders R., Li K., Liu X., Levitsky A. A., Majoros W. H., Chen Q., Xia A. C., Lopez J. R., Donnelly M. T., Newman M. H., Glodek A., Kraft C. L., Nodell M., Ali F., An H. J., Baldwin-Pitts D., Beeson K. Y., Cai S., Carnes M., Carver A., Caulk P. M., Center A., Chen Y. H., Cheng M. L., Coyne M. D., Crowder M., Danaher S., Davenport L. B., Desilets R., Dietz S. M., Doup L., Dullaghan P., Ferriera S., Fosler C. R., Gire H. C., Gluecksmann A., Gocayne J. D., Gray J., Hart B., Haynes J., Hoover J., Howland T., Ibegwam C., Jalali M., Johns D., Kline L., Ma D. S., MacCawley S., Magoon A., Mann F., May D., McIntosh T. C., Mehta S., Moy L., Moy M. C., Murphy B. J., Murphy S. D., Nelson K. A., Nuri Z., Parker K. A., Prudhomme A. C., Puri V. N., Qureshi H., Raley J. C., Reardon M. S.,

Regier M. A., Rogers Y. H., Romblad D. L., Schutz J., Scott J. L., Scott R., Sitter C. D., Smallwood M., Sprague A. C., Stewart E., Strong R. V., Suh E., Sylvester K., Thomas R., Tint N. N., Tsonis C., Wang G., Wang G., Williams M. S., Williams S. M., Windsor S. M., Wolfe K., Wu M. M., Zaveri J., Chaturvedi K., Gabrielian A. E., Ke Z., Sun J., Subramanian G., Venter J. C., Pfannkoch C. M., Barnstead M., Stephenson L. D. (2002) A comparison of whole-genome shotgun-derived mouse chromosome 16 and the human genome. *Science* 296(5573), 1661–1671.

38. Kirkness E. F., Bafna V., Halpern A. L., Levy S., Remington K., Rusch D. B., Delcher A. L., Pop M., Wang W., Fraser C. M., Venter J. C. (2003) The dog genome: survey sequencing and comparative analysis. *Science* 301(5641), 1898–1903.
39. Holt R. A., Subramanian G. M., Halpern A., Sutton G. G., Charlab R., Nusskern D. R., Wincker P., Clark A. G., Ribeiro J. M., Wides R., Salzberg S. L., Loftus B., Yandell M., Majoros W. H., Rusch D. B., Lai Z., Kraft C. L., Abril J. F., Anthouard V., Arensburger P., Atkinson P. W., Baden H., de Berardinis V., Baldwin D., Benes V., Biedler J., Blass C., Bolanos R., Boscus D., Barnstead M., Cai S., Center A., Chaturverdi K., Christophides G. K., Chrystal M. A., Clamp M., Cravchik A., Curwen V., Dana A., Delcher A., Dew I., Evans C. A., Flanigan M., Grundschober-Freimoser A., Friedli L., Gu Z., Guan P., Guigo R., Hillenmeyer M. E., Hladun S. L., Hogan J. R., Hong Y. S., Hoover J., Jaillon O., Ke Z., Kodira C., Kokoza E., Koutsos A., Letunic I., Levitsky A., Liang Y., Lin J. J., Lobo N. F., Lopez J. R., Malek J. A., McIntosh T. C., Meister S., Miller J., Mobarry C., Mongin E., Murphy S. D., O'Brochta D. A., Pfannkoch C., Qi R., Regier M. A., Remington K., Shao H., Sharakhova M. V., Sitter C. D., Shetty J., Smith T. J., Strong R., Sun J., Thomasova D., Ton L. Q., Topalis P., Tu Z., Unger M. F., Walenz B., Wang A., Wang J., Wang M., Wang X., Woodford K. J., Wortman J. R., Wu M., Yao A., Zdobnov E. M., Zhang H., Zhao Q., Zhao S., Zhu S. C., Zhimulev I., Coluzzi M., della Torre A., Roth C. W., Louis C., Kalush F., Mural R. J., Myers E. W., Adams M. D., Smith H. O., Broder S., Gardner M. J., Fraser C. M., Birney E., Bork P., Brey P. T., Venter J. C., Weissenbach J., Kafatos F. C., Collins F. H., Hoffman S. L. (2002) The genome sequence of the malaria mosquito Anopheles gambiae. *Science* 298 (5591), 129–149.
40. Shizuya H., Birren B., Kim U. J., Mancino V., Slepak T., Tachiiri Y., Simon M. (1992) Cloning and stable maintenance of 300-kilobase-pair fragments of human DNA in Escherichia coli using an F-factor-based vector. *Proc Natl Acad Sci USA* 89(18), 8794–8797.
41. Stein L. (2001) Genome annotation: from sequence to biology. *Nat Rev Genet* 2(7), 493–503.
42. Reese M. G., Hartzell G., Harris N. L., Ohler U., Abril J. F., Lewis S. E. (2000) Genome annotation assessment in Drosophila melanogaster. *Genome Res* 10(4), 483–501.
43. Lowe T. M., Eddy S. R. (1997) tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res* 25(5), 955–964.
44. Pennacchio L. A., Rubin E. M. (2001) Genomic strategies to identify mammalian regulatory sequences. *Nat Rev Genet* 2(2), 100–109.
45. Mi H., Guo N., Kejariwal A., Thomas P. D. (2007) PANTHER version 6: protein sequence and function evolution data with expanded representation of biological pathways. *Nucleic Acids Res* 35(Database issue), D247–D252.
46. Mi H., Lazareva-Ulitsky B., Loo R., Kejariwal A., Vandergriff J., Rabkin S., Guo N., Muruganujan A., Doremieux O., Campbell M. J., Kitano H., Thomas P. D. (2005) The PANTHER database of protein families., subfamilies., functions and pathways. *Nucleic Acids Res* 33(Database issue), D284–D288.
47. Thomas P. D., Campbell M. J., Kejariwal A., Mi H., Karlak B., Daverman R., Diemer K., Muruganujan A., Narechania A. (2003) PANTHER: a library of protein families and subfamilies indexed by function. *Genome Res* 13(9), 2129–2141.
48. Thomas P. D., Kejariwal A., Campbell M. J., Mi H., Diemer K., Guo N., Ladunga I., Ulitsky-Lazareva B., Muruganujan A., Rabkin S., Vandergriff J. A., Doremieux O. (2003) PANTHER: a browsable database of gene products organized by biological function., using curated protein family and subfamily classification. *Nucleic Acids Res* 31(1), 334–341.
49. Blake, J. A., Harris, M. A. (2002) The Gene Ontology (GO) project: structured vocabularies for molecular biology and their application to genome and expression analysis. *Curr Protoc Bioinformatics Chapter* 7, Unit 7.2.

50. Camon E., Barrell D., Brooksbank C., Magrane M., Apweiler R. (2003) The Gene Ontology Annotation (GOA) Project—application of GO in SWISS-PROT., TrEMBL and InterPro. *Comp Funct Genomics* 4(1), 71–74.
51. Selkov E., Overbeek R., Kogan Y., Chu L., Vonstein V., Holmes D., Silver S., Haselkorn R., Fonstein M. (2000) Functional analysis of gapped microbial genomes: amino acid metabolism of Thiobacillus ferrooxidans. *Proc Natl Acad Sci USA* 97(7), 3509–3514.
52. Fleischmann R. (2001) Single nucleotide polymorphisms in Mycobacterium tuberculosis structural genes—response to Dr. Musser. *Emerg Infect Dis* 7(3), 487–488.
53. Pevzner P. A., Tang H., Waterman M. S. (2001) An Eulerian path approach to DNA fragment assembly. *Proc Natl Acad Sci USA* 98(17), 9748–9753.
54. Batzoglou S., Jaffe D. B., Stanley K., Butler J., Gnerre S., Mauceli E., Berger B., Mesirov J. P., Lander E. S. (2002) ARACHNE: a whole-genome shotgun assembler. *Genome Res* 12(1), 177–189.
55. Gordon D., Desmarais C., Green P. (2001) Automated finishing with autofinish. *Genome Res* 11(4), 614–625.
56. Tammi M. T., Arner E., Kindlund E., Andersson B. (2003) Correcting errors in shotgun sequences. *Nucleic Acids Res* 31 (15), 4663–4672.
57. Tammi M. T., Arner E., Kindlund E., Andersson B. (2004) ReDiT: Repeat Discrepancy Tagger—a shotgun assembly finishing aid. *Bioinformatics* 20(5), 803–804.
58. Bartels D., Kespohl S., Albaum S., Druke T., Goesmann A., Herold J., Kaiser O., Puhler A., Pfeiffer F., Raddatz G., Stoye J., Meyer F., Schuster S. C. (2005) BACCardI—a tool for the validation of genomic assemblies., assisting genome finishing and intergenome comparison. *Bioinformatics* 21(7), 853–859.
59. Arner E., Tammi M. T., Tran A. N., Kindlund E., Andersson B. (2006) DNPTrapper: an assembly editing tool for finishing and analysis of complex repeat regions. *BMC Bioinformatics* 7, 155.
60. Fitch W. M. (1970) Distinguishing homologous from analogous proteins. *Syst Zool* 19 (2), 99–113.
61. Fitch W. M., Margoliash E. (1970) The usefulness of amino acid and nucleotide sequences in evolutionary studies. *Evol Biol* 4, 67–109.
62. Altschul S. F., Gish W., Miller W., Myers E. W., Lipman D. J. (1990) Basic local alignment search tool. *J Mol Biol* 215(3), 403–410.
63. Altschul S. F., Madden T. L., Schaffer A. A., Zhang J., Zhang Z., Miller W., Lipman D. J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25(17), 3389–3402.
64. Koski L. B., Golding G. B. (2001) The closest BLAST hit is often not the nearest neighbor. *J Mol Evol* 52(6), 540–542.
65. Hirsh A. E., Fraser H. B. (2001) Protein dispensability and rate of evolution. *Nature* 411(6841), 1046–1049.
66. Jordan I. K., Rogozin I. B., Wolf Y. I., Koonin E. V. (2002) Essential genes are more evolutionarily conserved than are nonessential genes in bacteria. *Genome Res* 12(6), 962–968.
67. Wall D. P., Fraser H. B., Hirsh A. E. (2003) Detecting putative orthologs. *Bioinformatics* 19(13), 1710–1711.
68. Deluca T. F., Wu I. H., Pu J., Monaghan T., Peshkin L., Singh S., Wall D. P. (2006) Roundup: a multi-genome repository of orthologs and evolutionary distances. *Bioinformatics* 22(16), 2044–2046.
69. Lee M. M., Chan M. K., Bundschuh R. (2008) Simple is beautiful: a straightforward approach to improve the delineation of true and false positives in PSI-BLAST searches. *Bioinformatics* 24, 1339–1343.
70. Poptsova M. S., Gogarten J. P. (2007) BranchClust: a phylogenetic algorithm for selecting gene families. *BMC Bioinformatics* 8, 120.
71. Haas B. J., Delcher A. L., Wortman J. R., Salzberg S. L. (2004) DAGchainer: a tool for mining segmental genome duplications and synteny. *Bioinformatics* 20(18), 3643–3646.
72. Celamkoti S., Kundeti S., Purkayastha A., Mazumder R., Buck C., Seto D. (2004) GeneOrder3.0: software for comparing the order of genes in pairs of small bacterial genomes. *BMC Bioinformatics* 5, 52.
73. Wheeler D. L., Barrett T., Benson D. A., Bryant S. H., Canese K., Chetvernin V., Church D. M., Dicuccio M., Edgar R., Federhen S., Feolo M., Geer L. Y., Helmberg W., Kapustin Y., Khovayko O., Landsman D., Lipman D. J., Madden T. L., Maglott D. R., Miller V., Ostell J., Pruitt K. D., Schuler G. D., Shumway M., Sequeira E., Sherry S. T., Sirotkin K., Souvorov A., Starchenko G., Tatusov R. L., Tatusova T. A., Wagner L.,

Yaschenko E. (2008) Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res* 36(Database issue), D13–D21.

74. Peterson J. D., Umayam L. A., Dickinson T., Hickey E. K., White O. (2001) The Comprehensive Microbial Resource. *Nucleic Acids Res* 29(1), 123–125.
75. Choi K., Ma Y., Choi J. H., Kim S. (2005) PLATCOM: a Platform for Computational Comparative Genomics. *Bioinformatics* 21 (10), 2514–2516.
76. Toft C., Fares M. A. (2006) GRAST: a new way of genome reduction analysis using comparative genomics. *Bioinformatics* 22 (13), 1551–1561.
77. Xie T., Hood L. (2003) ACGT—a comparative genomics tool. *Bioinformatics* 19(8), 1039–1040.
78. Chen T., Abbey K., Deng W. J., Cheng M. C. (2005) The bioinformatics resource for oral pathogens. *Nucleic Acids Res* 33(Web Server issue), W734–W740.
79. Leader D. P. (2004) BugView: a browser for comparing genomes. *Bioinformatics* 20 (1), 129–130.
80. Yang J., Wang J., Yao Z. J., Jin Q., Shen Y., Chen R. (2003) GenomeComp: a visualization tool for microbial genome comparison. *J Microbiol Methods* 54(3), 423–426.
81. Romualdi A., Felder M., Rose D., Gausmann U., Schilhabel M., Glockner G., Platzer M., Suhnel J. (2007) GenColors: annotation and comparative genomics of prokaryotes made easy. *Methods Mol Biol* 395, 75–96.
82. Grant J. R., Stothard P. (2008) The CGView Server: a comparative genomics tool for circular genomes. *Nucleic Acids Res* 36, W181–W184.
83. Ghai R., Chakraborty T. (2007) Comparative microbial genome visualization using GenomeViz. *Methods Mol Biol* 395, 97–108.
84. Dubchak I., Ryaboy D. V. (2006) VISTA family of computational tools for comparative analysis of DNA sequences and whole genomes. *Methods Mol Biol* 338, 69–89.
85. Hohl M., Kurtz S., Ohlebusch E. (2002) Efficient multiple genome alignment. *Bioinformatics* 18(Suppl 1), S312–S320.
86. Treangen T. J., Messeguer X. (2006) M-GCAT: interactively and efficiently constructing large-scale multiple genome comparison frameworks in closely related species. *BMC Bioinformatics* 7, 433.
87. Darling A. C., Mau B., Blattner F. R., Perna N. T. (2004) Mauve: multiple alignment of conserved genomic sequence with rearrangements. *Genome Res* 14(7), 1394–1403.
88. Tzika A. C., Helaers R., Van de Peer Y., Milinkovitch M. C. (2008) MANTIS: a phylogenetic framework for multi-species genome comparisons. *Bioinformatics* 24 (2), 151–157.
89. Andersson S. G., Kurland C. G. (1998) Reductive evolution of resident genomes. *Trends Microbiol* 6(7), 263–268.
90. Oliver K. M., Russell J. A., Moran N. A., Hunter M. S. (2003) Facultative bacterial symbionts in aphids confer resistance to parasitic wasps. *Proc Natl Acad Sci USA* 100(4), 1803–1807.
91. Bensadia F., Boudreault S., Guay J. F., Michaud D., Cloutier C. (2006) Aphid clonal resistance to a parasitoid fails under heat stress. *J Insect Physiol* 52(2), 146–157.
92. Degnan P. H., Moran N. A. (2008) Evolutionary genetics of a defensive facultative symbiont of insects: exchange of toxin-encoding bacteriophage. *Mol Ecol* 17(3), 916–929.
93. Douglas A. E. (1996) Reproductive failure and the free amino acid pools in pea aphids (Acyrthosiphon pisum) lacking symbiotic bacteria. *J Insect Physiol* 42(3), 247–255.
94. Buchner P. (1965) Endosymbiosis of animals with plant microorganisms. *Interscience*, New York., NY.
95. Muller H. J. (1964) The relation of recombination to mutation advance. *Mutat Res* 1, 2–9.
96. Perez-Brocal V., Gil R., Ramos S., Lamelas A., Postigo M., Michelena J. M., Silva F. J., Moya A., Latorre A. (2006) A small microbial genome: the end of a long symbiotic relationship? *Science* 314(5797), 312–313.
97. Koonin E. V. (2000) How many genes can make a cell: the minimal-gene-set concept. *Annu Rev Genomics Hum Genet* 1, 99–116.
98. Gerdes S. Y., Scholle M. D., Campbell J. W., Balazsi G., Ravasz E., Daugherty M. D., Somera A. L., Kyrpides N. C., Anderson I., Gelfand M. S., Bhattacharya A., Kapatral V., D'Souza M., Baev M. V., Grechkin Y., Mseeh F., Fonstein M. Y., Overbeek R., Barabasi A. L., Oltvai Z. N., Osterman A. L. (2003) Experimental determination and system level analysis of essential genes in Escherichia coli MG1655. *J Bacteriol* 185(19), 5673–5684.

99. Koonin E. V. (2003) Comparative genomics., minimal gene-sets and the last universal common ancestor. *Nat Rev Microbiol* 1 (2), 127–136.

100. Moran N. A. (1996) Accelerated evolution and Muller's rachet in endosymbiotic bacteria. *Proc Natl Acad Sci USA* 93(7), 2873–2878.

101. Prickett M. D., Page M., Douglas A. E., Thomas G. H. (2006) BuchneraBASE: a post-genomic resource for Buchnera sp. APS. *Bioinformatics* 22(5), 641–642.

102. Tillier E. R., Collins R. A. (2000) Genome rearrangement by replication-directed translocation. *Nat Genet* 26(2), 195–197.