Current Challenges in Bioinformatics

João Meidanis

[1] Scylla Bioinformatics, Estrada Unicamp-Telebrás, km 0,97, P.O.Box 6123,
13084-971 Campinas, Brazil
meidanis@scylla.com.br
[2] University of Campinas, Institute of Computing, P.O.Box 6176,
13084-970 Campinas, Brazil
meidanis@ic.unicamp.br,
http://www.ic.unicamp.br/~meidanis

Abstract. My purpose in this text is to highlight some of the most important challenges in the area of Bioinformatics, drawing from several sources. The field is already pretty large and becoming more so. Therefore the selection of challenges presented here will tend to focus on topics I am more familiar with, with only brief mentions of topics I do not know in depth. The challenges vary in scope and motivation: some are broad, abstract while others are specialized to a given topic. Some are biologically motivated, others are nice as computer science problems. Also, I tried to show the dependencies among the challenges in order to get a global picture of the area. A basic knowledge on the principles of computational biology is assumed.

1 Introduction

In Physics, we know how to predict the behavior of solids (hence we are able to construct tall buildings and long bridges), the behavior of waves (hence we are able to communicate through fiber optics), and of the atoms (hence we are able to use nuclear energy).

In Chemistry we know less. We are able to predict the outcome of some mixtures of substances, we know how to make dyes, plastic, and how to process petrol. We know a lot about molecules, but still cannot predict the macroscopic behavior of a substance from its molecules, even for something as simple as water.

In Biology we know even less. There are few, if any, general, quantitative theories. Most of the knowledge is in the form of "lists of facts." For every rule there are exceptions. We know the basics about cells, but we do not know how cells differentiate in a multicellular organism. We can control some diseases, but many others still defy our knowledge. The attempts at biological pest control have seen success stories, but also formidable failures. "Biology is a mess", a phrase I once heard from a distinguished bioscientist, conveys well the amount of work still needed.

It seems like Biology is in the stage that Chemistry was a while ago, and that Chemistry is in the stage that Physics was a while ago. On the other hand,

M.A. Nascimento, E.S. de Moura, A.L. Oliveira (Eds.): SPIRE 2003, LNCS 2857, pp. 16–27, 2003.

Computer Science is, like Mathematics, a helper science. It finds applications in Physics and Chemistry, but mainly related to "number crunching." Although "number crunching" applications do exist in Biology, genomics brought combinatorial problems to the scene. DNA and protein sequences, the conveyors of information in living cells, have a strong combinatorial appeal.

But Computer Science and Biology have one important thing in common. Software is easier to generate than concrete objects like cars and buildings, so we are seeing the appearance of very large, very complex software systems. I dare say that the complexity of software systems today exceeds, or is about to exceed, the complexity of any other artifact produced by humans, and as such is approaching the complexity of living cells, being nonetheless still far from it.

So, apart from the combinatorial problems, a new kind of challenges related to managing complexity, more closely related to Software Engineering, appears when Computer Science touches Biology. I will try and comment on challenges of both kinds, and the relationship among them.

It is hard to understand complex systems. The human mind seems to have a limited capacity in the number of details it can handle. One approach to deal with complex systems is to try and extract the essential aspects, the most important for a given goal, and to design models to predict just those aspects. It is more or less like projecting a multidimensional vector into a given subspace. To describe the entire system in a meaningful way, many such projected, integrated views are necessary. We will see how this is being done, for instance, in protein classification.

The rest of the paper is organized as follows. Section 2 presents the major challenges as seen by biologists. Section 3 presentes the major challenges as seen by computer scientists. Section 4 presents more specialized areas, which can be seen as providing building blocks for the major challenges. We conclude with perspectives in Section 5.

2 Bioinformatic Challenges as Seen by Biologists

I will begin trying to capture what people primarily trained as bioscientists see as challenges. I am not a bioscientist myself, so I will draw from other sources. Collins and colleagues from the US National Human Genome Research Institute wrote about the future of genome research, and cite Computational Biology as an important resource, but do not enter in specifics [4]. At a bioinformatics conference in 2002, EBI's Ewan Birney, MIT's Chris Burge, and GlaxoSmithKline's Jim Fickett gave an impromptu roundup of the future challenges for the field. They came up with the "Ten Most Wanted," listed below. Highlighting in boldface is mine.

- Precise, predictive model of **transcription** initiation and termination: ability to predict where and when transcription will occur in a genome
- Precise, predictive model of RNA **splicing**/alternative splicing: ability to predict the splicing pattern of any primary transcript in any tissue

- Precise, quantitative models of signal **transduction** pathways: ability to predict cellular responses to external stimuli
- Determining effective protein:DNA, protein:RNA and protein:protein **recognition codes**
- Accurate ab initio **protein structure prediction**
- Rational **design** of small molecule inhibitors of proteins
- Mechanistic understanding of **protein evolution**: understanding exactly how new protein functions evolve
- Mechanistic understanding of **speciation**: molecular details of how speciation occurs
- Continued development of effective gene **ontologies** - systematic ways to describe the functions of any gene or protein
- **Education**: development of appropriate bioinformatics curricula for secondary, undergraduate and graduate education

These challenges are in general of a high difficulty, i.e., they are likely to require many years, decades, or even more time of combined effort from many groups around the world to be satisfactorily solved. The first six challenges will probably be solved with the aid of large amounts of data collected in the laboratories, analyzed manually or semi-automatically to generate curated data which will serve as "benchmarks" for hypothesis testing. Protein evolution and speciation (a topic in species evolution) are much more abstract. It is even hard to formulate them as input/output problems (a favorite format for computer scientists). One key issue here is to find ways of *measuring evolution*. Gene ontologies fall more into the domain of understanding a complex system via projections. The education challenge is of a completely different kind. It deals with dissemination of knowledge rather than with the generation of new knowledge.

Figure 1 is an attempt to link together these challenges. Each challenge is seen as a "black box", represented as a rectangle containing an abbreviated description of the challenge (based on the bold terms in their statements above), which can transform certain types of knowledge into others. Pieces of knowledge consisting of catalogs of various objects occurring in Nature are depicted as phrases without boundaries. Ovals delimit main bodies of knowledge, "knowledge on proteins" and "knowledge on biological processes". Therefore each challenge has "inputs" and "outputs" and together they form a knowledge network where the ultimate goal is to understand biological processes. I included a "translation" box to complete the network but this is not a challenge as it is relatively well understood. Challenges "protein evolution" and "speciation" where depicted as some kind on enhancers of the knowledge about proteins and biological processes, respectively, drawing information from them and giving back improved information. The "education" challenge permeates everything.

3 Bioinformatic Challenges for Computer Scientists

I will now try and capture what people trained primarily in Computer Sciences perceive as challenges in the field of Bioinformatics. I will start with very broad, practical issues and them proceed to more specialized ones.

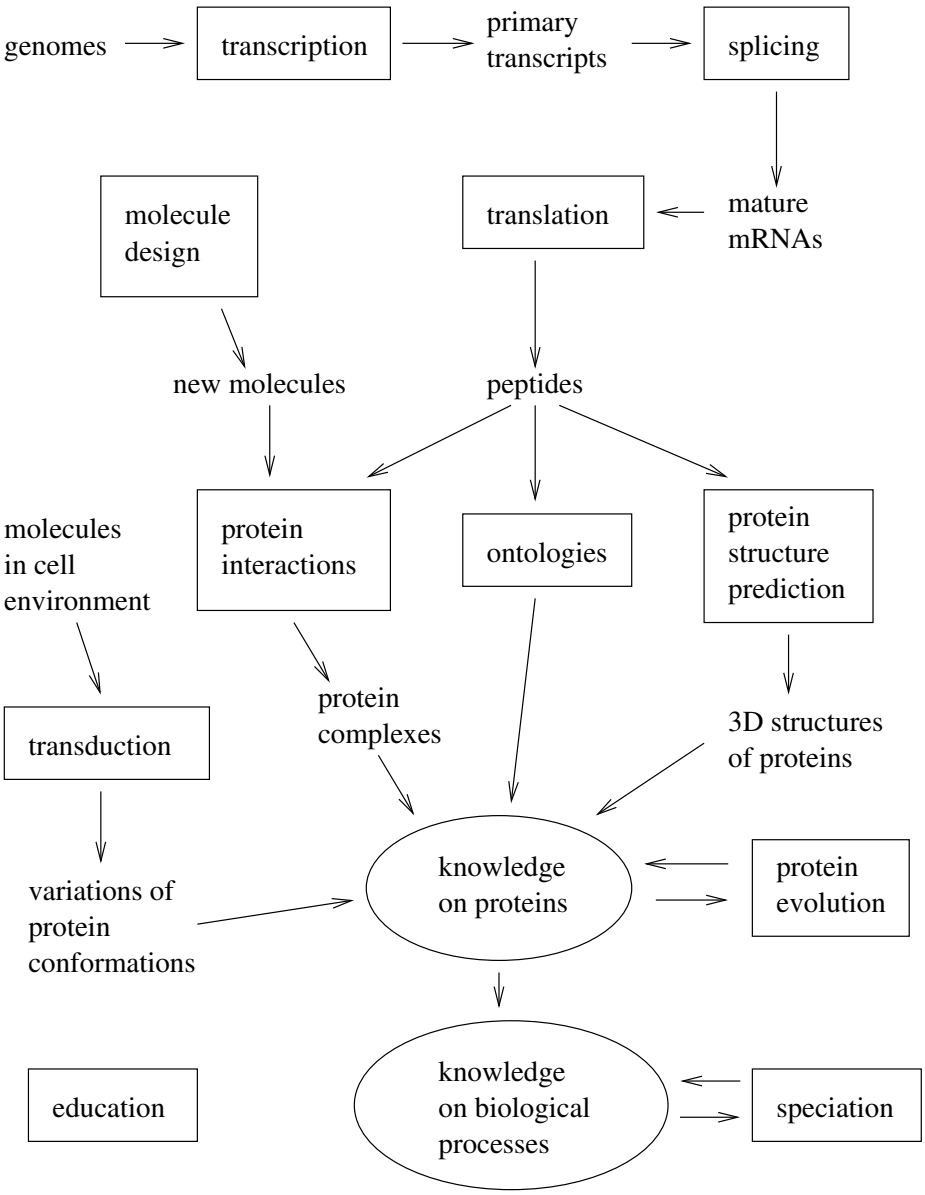

Fig. 1. Relationship among challenges. Each challenge is represented as a rectangle and can transform certain types of knowledge into others. Catalogues of natural objects are depicted as phrases without boundaries. Ovals delimit main bodies of knowledge. The challenges form a knowledge network where the ultimate goal is to understand biological processes. The "translation" box which is not a challenge since it is relatively well understood. The "education" challenge permeates everything.

Having been in charge of many Brazilian Genome projects, and now as CEO of Scylla Bioinformatics, I acquired certain impressions as to what the real issues are that need to be solved if we want to fully explore the amount of genomic data that is being generated today. I was glad to observe that these impressions are for the most part confirmed and substantiated by the testimony before the NSF Advisory Committee on Cyberinfrastructure of Gene Myers [12], a leading, world-renowned bioinformatician, who as vice-president of Celera Genomics had to deal with much more data and complex problems than myself.

According to Myers, current hardware is adequate. The problem is software. Software for data storage and sharing, software for parallel flows on computer grids, and, perhaps more significantly, software for writing software.

Better information management technologies are needed. The challenge is to find models and develop flexible data mining capabilities that can deal with huge sets, semi-structured data, experimental error, and integration of loosely coupled data.

Challenge 1 (Information management) *Create novel, paradigm-breaking systems for information management.*

Relational databases are the success that we all know, but for bioinformatics they lack some important characteristics. The entries have to be single-valued, and in biology many times we need multi-valued fields, for instance, when classifying a protein according to an ontology. Changing the schema is painful, and is an operation that researchers would like to do more often than, say, in commercial applications. The query language, SQL, is very basic and to write complex queries that execute efficiently can be very challenging and not intuitive.

Individual attempts have been made to alleviate some of these problems, with relative success, but still an integrated solution is needed. AceDB is a database specifically designed for biological data, and implements multi-valued fields as default [6]. It also has its own query language. The SRS (Sequence Retrieval System — see [20] for a recent development) provides capabilities to search multiple databases and goes a long way solving the integration problem. Perhaps that is the limit of what we can do with current technology.

The XML standard is an advance, but data grows too much with it, and it lacks semantics. Graham [7] points out that XML looks a lot like LISP, only not as prolix.

Challenge 2 (Parallelism) *Design expressive control systems for distributed, heterogeneous computing.*

Some issues become more apparent when we examine an application, for instance, data mining. Data mining can be loosely defined as a type of search where you do not really know in advance what you are looking for, but you can recognize when you stumble into something interesting. So you try many search approaches, each time learning from the previous one and producing improved searches, until you are satisfied with the result. If each approach requires a modification of the software, or of the database schema, as it often does, then

we have a bottleneck if people who use the tools are not the same that make the tools. A real revolution would occur when biologists are able to make their own tools and modify existing tools in a fast and effective way.

Challenge 3 (Programability) *Improve the programmability of computers.*

This challenge is a very general one and it is also important in a variety of other fields. As such, bioinformatics may benefit from advances in other fields, but it may also contribute to its solution, since in bioinformatics many users with no specific formation in computer science need to use computers in very sophisticaded ways.

4 The Specialized Challenges

Up to now we have focused in relatively abstract, broad challenges. But as we try and solve a big challenge, we often need to break it apart into a number of more detailed, specialized problems, some of which can still be difficult enough to keep researchers busy for many years. These specialized problems are just as important as the large ones. They are the bricks with which we can construct a tall building. Although they can be attacked by individual researchers or groups independently of the context, it is beneficial to know the context so that we can modify the challenge into a similar, more easily solvable one, but that is still useful in the context that motivated it.

In this section I mention some of the areas containing specialized in Computational Biology that would hopefully attract the interest of computer scientists, and can contribute toward solving the main challenges. This is but a small sample of the existing areas, whose choice was influenced by my own personal experience.

The starting point for this part will be the book written by Setubal an myself in 1997 [17]. This is an introductory textbook where the basic techniques and background are presented, divided by areas of interest. However, new developments spurted the appearance of new areas, or reshaped significantly old areas in the last years, and I will comment on some of those.

I made the decision of still keeping the discussion at a higher level, not going down to the precise definitions of problems, but rather commenting on the area in a more general form. I had to do that because otherwise this text would have become exceedingly long. I intend however to keep updating it until it reaches this stage. One undesirable effect of this decision is that the text is not autocontained, and readers are expected to be familiar with the basic notions of computational biology, as explained in ours or other textbooks in the area.

4.1 Sequence Comparison

Pairwise sequence comparison is a relative well-solved problem in the community. Everybody seems to be happy with the existing algorithms, although of

course people are always looking for faster implementations or approaches (see, for instance, the recent work by Myers and Durbin on accelerating the Smith-Waterman algorithm [11]). Score matrices have also apparently stabilized with the BLOSUM matrices (BLOSUM50 is the default in Fasta align, BLOSUM62 the default in BLAST). There is also a pretty stable agreement around gap parameters (gap open ranging from 10 to 12, gap extend from 0.5 to 3). Fasta align, Emboss align, BLAST, and SeqAln all use default values in these ranges.

One challenge that appeared in the last decade or so and is still with us is the matter of aligning long sequences. It is known that when one applies the standard alignment algorithms to a pair of very long sequences, for instance, genome size sequences, the resulting optimal alignment may contain long contiguous regions of very bad score, which is balanced in the final score by very long, very good flanking regions. This should in fact be reported as two very good alignments, dropping the bad part in between, and not as a single, longer alignment [1].

Other challenges here include multiple sequence alignment, and comparisons between genomic DNA and cDNA, and between DNA and protein sequences.

4.2 Fragment Assembly

Fragment assembly is one of the oldest problems in bioinformatics, because since the very beginning of DNA sequencing technology researchers wanted to reconstruct genome pieces larger than read length. The Institute for Genomic Research (TIGR) maintains an interesting page with the history of the "largest contig published so far" [9]. Current champion is human chromosome 14 with over 87 million base pairs.

Besides the traditional overlap-layout-consensus paradigm, which is still the norm in production work, both sequencing by hybridization and the Eulerian method have appeared as alternatives. Software package EULER by Pevzner and colleagues is a promising tool that explores these alternatives [16, 15].

4.3 Physical Mapping

To study long chromosomes in certain ways it is necessary to clone pieces of them, and to be able to map back these clones in the longer sequence. Physical mapping of DNA motivates several interesting combinatorial problems, both in the digest and hybridization versions. We refer the reader to the collection of open problems organized by Pevzner and Waterman [14], where several digest problems are mentioned, some of them still open.

In the absence of experimental errors the hybridization version is equivalent to the consecutive ones property for (0,1) matrices. PQR trees can be used to solve the problem in this case, and recently an almost linear time algorithm was presented to construct the PQR tree corresponding to a matrix [19]. Although PQR trees can pinpoint subsets of probes where experimental errors might have occurred, it is not clear how to solve the practical problem with errors.

4.4 Phylogenetic Trees

Phylogenetic trees are diagrams that describe the evolution of a set of species from a common ancestor of them all. It is related to the protein evolution challenge cited earlier, only here entire organisms are compared. The construction of phylogenetic trees is an old problem, but it gained additional momentum with the advent of high volumes of molecular data. Sequences of several species can be aligned and each column can be viewed as a character. Alternatively, distances can be computed from sequences and used as input for the tree construction algorithms.

It is often the case that the use of different methods and/or inputs for the same set of species yields different trees. One important problem is to combine various trees for the same species into a tree that is potentially better, in the sense that it has a higher probability of reflecting the real evolutionary scenario.

4.5 Genome Rearrangements

Genome rearrangements refer to the study of large transformations in a genome, such as the reversal of a large contiguous region, or a translocation between two chromosomes. Weights are given to a set of allowed operations and the goal is to compute, given two genomes, a series of operations that transforms one into the other with minimum total weight. This minimum weight can be used as a measure of distance for phylogenetic tree calculations.

This topic is relatively recent in bioinformatics, compared to others mentioned here. It received a lot of attention when the reversal problems for signed and unsigned (gene orientation known or not know, respectively), were proved polynomial time solvable and NP-hard, respectively, in the mid 90s [8, 3]. The proofs were rather involved, and many people have worked on these problems, simplifying proofs, improving algorithms, and trying to solve the open problems. The transposition operation has been resisting researcher's efforts for many years. The difficulty of this problem seems to extend to problems involving transposition as one of the operations as well.

To apply the genome rearrangement distance in real instances we need to solve the issues of duplicated genes, and genes appearing in just one of the genomes, either by fixing a one-to-one mapping between the sets of genes that will be considered, or by including gene duplication, creation, and deletion in the operation set. This problem is related to genome comparison (Section 4.6).

4.6 Genome Comparison

Genome comparison has many applications. It can be used to predict genes, because of gene conservation among species, or to measure the differences and produce input for phylogeny construction. The first challenge is to know which genes in one genome correspond to which genes in the other. Sometimes a gene in genome A does not correspond to any gene in genome B, and sometimes it corresponds to more than one. If a gene is present in many genomes, it can

be used to try and establish the evolutionary relationship between them, as ribosomal RNA has been [13]. This can be done using phylogenetic tree building techniques, but the challenge here is that using different genes we often end up with different trees, and we have to find a consensus of them all (Section 4.4). Of course some genes may be bad candidates for tree building, for instance, because they were acquired by horizontal transfer rather than by vertical inheritance.

Genome comparison can be seen as a collection of techniques that includes long sequence comparison, genome rearrangements, and others and is useful in gene prediction and phylogenetic tree construction.

4.7 Micro-array Experiments

This topic is among the newest in the scene, and it is getting a tremendous amount of attention. In a micro-array experiment we have a fixed set of (usually) DNA probes printed on a rectangular array. The quantity of such probe sequences can be of the order of thousands. Each probe is part of a gene that we are interested in. An RNA preparation containing messenger RNAs extracted from live tissue is then put in contact with the array. Those that hybridize (that is, have sequence complementary with) a probe will be caught in the array and this can be measured with the help of fluorescent dyes. Repeating the experiment with RNA preparations under several different conditions allows researchers to find out which of the array genes are expressed under which conditions, and how much they are expressed.

The key computational issue here is to cluster genes with similar expression patterns and use this information to draw conclusions on these genes. Hierarchical clustering (UPGMA) with the complement of the correlation coefficient seems to be the most widely used method in practice. Datta and Datta [5] compared several clustering methods and concluded that a divisive clustering method known as Diana [10] performs well in most cases, although other methods may outperform it depending on the data set and on the performance metric used.

The work of Spellman and colleagues [18] on yeast genes whose expression is regulated by the cell cycle has been very influential. One of the reasons is that the authors of this pioneering paper put together a web site where all the data can be retrieved. As a result, almost all papers on the subject use this data as a benchmark.

Since the output produced by laboratory instruments used in microarray experiments is in form of images, there is also the computational task of interpreting these images and generate the intensity values.

4.8 Protein Classification

The goal here is to predict the function of a given protein sequence, in general a previously unknown sequence obtained by translating a predicted gene in a high-throughput genome sequencing project, where the problem is also known as genome annotation. The definition of this problem is complicated because the meaning of “function” of the protein is far from clear. The function of a

protein can be very complex and multiple. Also, proteins are sometimes formed by two or more subunits, which are peptides translated from different genes. The SWISS-PROT database [2] is a key reference: an entry in this database is full of relevant information about the protein and contains links to several other web resources where a researcher can obtain even more information on the molecule.

With the plethora of genome projects around, and the need to annotate the data generated at a fast pace, automatic annotation becomes essential. Gene ontology (www.geneontology.org) is an initiative that aims at helping protein annotation by providing a standard vocabulary plus a classification of proteins or gene products into three orthogonal hierarchies:

- molecular function
- biological process
- cellular component (location)

For instance, a protein can have "transcription factor activity" as molecular function, "polarity of the adaxial/abaxial axis" as biological process, and "nucleus" as its cellular component. Each of these descriptions use controlled vocabulary and are in fact lower nodes in a hierarchy.

The hierarchies are not tree-like but rather the kind that permits a node to have more than one parent. It is therefore a connected, directed acyclic graph (DAG) with a single source.

5 Perspectives

We attempt to illustrate the relations between these problem areas using Figure 2. Each area is viewed as the developent of a set of techniques to help obtain data on three main domains: genomes, annotation, and evolution. The arrows indicate which areas are used by which other areas, offering a view that complements that of Figure 1.

The interconnections make clear the symbiosis between Biology and Computer Sciences. For computer scientists who want to work with bioinformatics, here is an advice: do not isolate, form interdisciplinary groups with bioscientists. Try to comprehend what "keeps them awake at night," i.e., the main issues in their research. Try to build a relationship where everyone is at the same time user and designer of the computing systems produced. Biologists *are* learning computer science at a fast pace.

For most of the problems, there are still not enough benchmarks. Work together to put those in place. It is an advance to at least specify their format, even if data will be filled in the future.

Quantify. Biologists are not very much used to think in numeric terms, but that is a necessity in the current state of affairs. With the advent of genomics, Biology is turning more and more into an exact science.

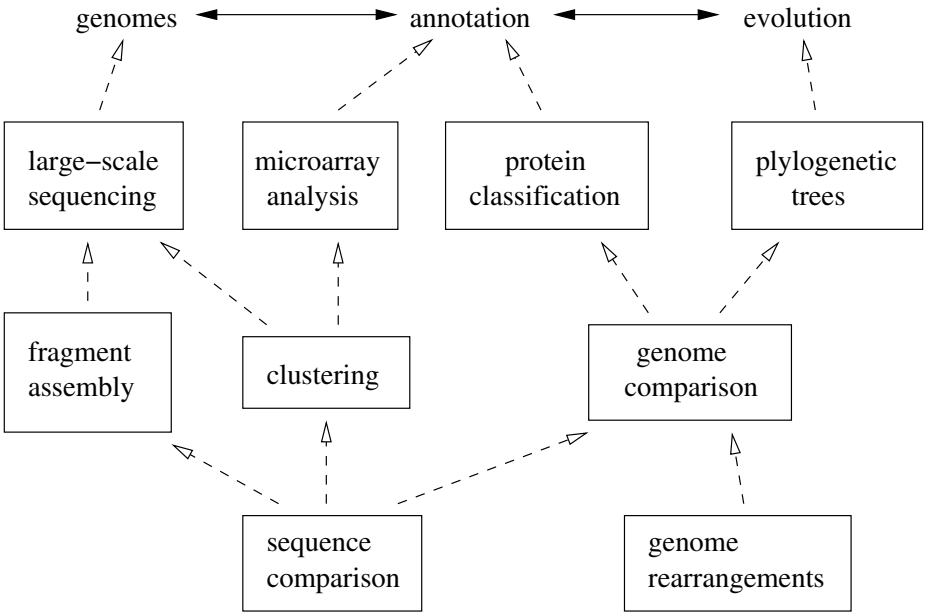

Fig. 2. Relationship among specialized problem areas in bioinformatics. The arrows indicate dependencies: an arrow from area B to area A means that A uses methods from B, or that problems in A create subproblems belonging to B. This diagram provides a view that complements that of Figure 1.

References

[1] Abdullah N. Arslan, Ömer Egecioglu, and Pavel A. Pevzner. A new approach to sequence comparison: normalized sequence alignment. *Bioinformatics*, 17:327–337, 2001.

[2] Boeckmann B., Bairoch A., Apweiler R., Blatter M.-C., Estreicher A., Gasteiger E., Martin M.J., Michoud K., O'Donovan C., Phan I., Pilbout S., and Schneider M. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Res.*, 31:365–370, 2003.

[3] A. Caprara. Sorting permutations by reversals and Eulerian cycle decompositions. *SIAM Journal on Discrete Mathematics*, 12(1):91–110, February 1999.

[4] Francis S. Collins, Eric D. Green, Alan E. Guttmacher, and Mark S. Guyer. A vision for the future of genomics research. *Nature*, 422:835–847, Apr 2003.

[5] S. Datta and S. Datta. Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*, 19:459–466, 2003.

[6] Richard Durbin and Jean Thierry-Mieg. A *C. elegans* database. Documentation, code and data available from http://www.acedb.org.

[7] Paul Graham. *ANSI Common Lisp*. Prentice Hall, 1995. ISBN 0133708756.

[8] S. Hannenhalli and P. A. Pevzner. Transforming cabbage into turnip: Polynomial algorithm for sorting signed permutations by reversals. *Journal of the ACM*, 46(1):1–27, January 1999.

[9] The Institute for Genomic Research. World's longest contiguous DNA sequence. www.tigr.org/tdb/contig_list.shtml, Jul 2003.

[10] L. Kaufman and P. J. Rousseeuw. *Fitting Groups in Data. An Introduction to Cluster Analysis*. Wiley, New York, 1990.

[11] G. Myers and R. Durbin. A table-driven, full-sensitivity similarity search algorithm. *Journal of Computational Biology*, 10(2):103–117, 2003.

[12] Gene Myers. Testimony for NSF Advisory Committee on Cyberinfrastructure. https://lapp1.cise-nsf.gov/rhilderb/index.htm, Jan 2002.

[13] G.J. Olsen and C.R. Woese. Ribosomal RNA: a key to phylogeny. *FASEB Journal*, 7:113–123, 1993.

[14] P. A. Pevzner and M. S. Waterman. Open combinatorial problems in computational molecular biology. In *Proceedings of the 3rd Israel Symposium on Theory of Computing and Systems*, pages 158–163. IEEE Computer Society Press, 1995.

[15] P.A. Pevzner and H. Tang. Fragment assembly with double-barreled data. *Bioinformatics*, Suppl 1:S225–233, 2001. Special ISMB 2001 issue.

[16] P.A. Pevzner, H. Tang, and M.S. Waterman. An Eulerian path approach to DNA fragment assembly. *Proc. Natl. Acad. Sci. USA*, 98(17):9748–9753, Aug 2001.

[17] J. C. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing Company, 1997. ISBN: 0-534-95262-3.

[18] Spellman et al. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 9:3273–3297, 1998. Web site with complementary material: http://genome-www.stanford.edu/cellcycle.

[19] Guilherme P. Telles. *An almost-linear time algorithm for PRQ trees and a scheme for clustering of expressed sequences of sugarcane*. PhD thesis, Istitute of Computing, University of Campinas, Campinas, Brazil, 2002. In Portuguese.

[20] E.M. Zdobnov, R. Lopez, R. Apweiler, and T. Etzold. The EBI SRS server — New features. *Bioinformatics*, 18(8):1149–1150, Aug 2002.