

Keynote Address: The Role of Algorithmic Research in Computational Genomics

Richard M. Karp
University of California at Berkeley and
International Computer Science Institute, Berkeley, California

June 10, 2003

Abstract

In the early 1990s, after more than three decades of studying algorithms within the framework of theoretical computer science, I shifted my focus to algorithmic problems arising in genomics. There is a fundamental difference between the views of algorithms in the two fields: in theoretical computer science the input-output behavior of an algorithm is rigorously specified in advance, whereas in computational biology an algorithm is merely a vehicle for discovering Nature's ground truth. In order to be effective in computational genomics I have had to radically change my approach to research. On the occasion of this keynote address I will share some of the lessons I have learned, in the hope of making the way easier for computer scientists and mathematicians entering this field. These lessons will be encapsulated in a list of aphorisms, accompanied by illustrative examples.

Learn the biological background of the problems you work on, and beware of overly simplistic problem formulations.

A case in point is the shortest superstring problem (given a set of strings, find the shortest string that is a superstring of each of the given strings), which was suggested as a formulation of the sequence assembly problem. This formulation is elegant, but it does not allow for errors in the sequence reads and, even on data from which the errors have been eliminated, gives overly compressed assemblies because it does not take into account the repetitive nature of genomic sequence.

Combinatorics and optimization theory are highly relevant.

Dynamic programming is a pervasive tool in computational genomics, shortest-path algorithms and network flow algorithms are widely used, integer programming is less frequently used but has been applied successfully to multiple alignment and physical mapping (Kececioğlu), and semidefinite programming has recently been applied to protein sequence similarity search (Buhler et al., ISMB 2003). Techniques from graph theory have been applied to sequencing by hybridization (Pevzner), and matroid algorithms have been applied to haplotyping problems (Gusfield et al.). Many further examples could be given.

Deep and elegant combinatorial algorithms arise only occasionally in bioinformatics.

Examples of elegant algorithms arise in the areas of genome rearrangement (Hannenhalli and Pevzner), phylogeny construction, approximate string matching, supervised learning and design of universal probe sets (Ben-Dor et al., RECOMB 2000), but more commonly the key to success lies in the appropriate application of simple mathematics.

Avoid problems of transient interest.

Problems such as multiple alignment, phylogeny construction, sequence similarity search and feature selection will always be relevant, whereas problems arising from specialized measurement technologies may quickly become irrelevant because of technological change, as I discovered in the course of my work on physical mapping.

Identify problems that are on the verge of becoming central to the field.

In the current post-genomic era problems of growing importance include the analysis of cis-regulation, the modeling of signal transduction pathways, SNP haplotyping and the use of multi-species comparative methods to find genes and regulatory signals in DNA.

Develop a user community for your algorithms.

However beautiful an algorithm may be, it will have no impact unless it serves the needs of biologists. To get your algorithm used it is necessary to work closely with your clients, the experimental biologists, and to provide an easy to use implementation. User experience will often suggest corrections and improvements as well as unexpected uses of an algorithm. As one example, an algorithm originally intended for creating physical maps from clone fingerprint data failed because of errors in the fingerprint data but, in modified form, proved extremely useful for pinpointing the locations of those errors (Thayer et al., *Genome Research*, 1999).

Accurate models of data are critical for algorithm design.

The implementation of an accurate quality measure for base calls in Phil Green's program Phred was a major advance in sequence assembly. The success of Celera's whole-genome shotgun approach hinged on insights into the nature of genomic repeats and recognition of the value of mated reads. Recent work on the detection of transcription factor binding motifs hinges on empirical observations about the "shape" of the motifs (Xing et al., this conference).

The use of diverse sources of data is often the key to success.

As one example, a recent approach to identifying cis-regulatory modules used an alignment of the human and mouse genomes, and databases of position weight matrices for transcription factor binding site motifs, binding site positions in the human genome, human genes, expression data for cell-cycle regulated genes and annotations of gene functions (Ben-Hur et al., ISMB 2003).

Stochastic models are of central importance.

All biological measurements are subject to random error which must be modeled probabilistically. In examples

such as the lysis-lysogeny decision in *E. coli*, the global behavior of a cell may depend on whether the number of copies of a protein is zero or small but nonzero, a chance event.

Statistical models are preferable to combinatorial optimization models.

In problems of identifying a sequence or structure from observations, statistical models make it possible to assign likelihoods to different solutions and establish confidence levels for inferences, whereas combinatorial optimization models focus on picking a 'best' solution according to an artificial scoring function.

An organism is best understood in the light of its evolutionary relationship to other organisms.

For this reason, comparative studies of genes or pathways in several related organisms are growing in importance. Virtually every biological mechanism that is investigated in computational genomics can be illuminated by studying its evolution using comparative data from several species.

Computational genomics can suggest new paradigms for computer science and statistics.

Living cells can adapt to changes in their environments, but large computer programs are brittle. Perhaps software engineering can benefit from a study of the sources of robustness in living systems.

Algorithm analysis in computer science typically studies the worst-case performance of an algorithm over all possible inputs, but examples from genomics suggest the construction of algorithms that adapt to the special characteristics of the data that will actually arise.

The work of the Celera group on whole-genome shotgun sequencing is an instance of a general approach to combinatorial puzzle solving in which constraints on the solution are enforced in an order determined by the strength of evidence for them. This approach has not yet been studied within theoretical computer science.

The trend toward using multiple sources of data to explain a biological phenomenon motivates renewed interest in statistical inference based on observations of many random variables of diverse types. More generally, biology has become a major application area for machine learning theory, a rapidly developing subject that bridges statistics and computer science.