Published on **MacDevCenter** (http://www.macdevcenter.com/)
http://www.macdevcenter.com/pub/a/mac/2004/06/11/bioinformatics.html
See this if you're having trouble printing code examples

Introduction to Bioinformatics

by Robert Jones
06/11/2004

Bioinformatics is the intersection of molecular biology and computer science. For software developers, it's a fascinating and challenging area in which to work. During the coming months, the Mac DevCenter will touch on different areas of bioinformatics; what the hot topics are right now, how Mac OS X and open source software are playing a role, and how you can get involved.

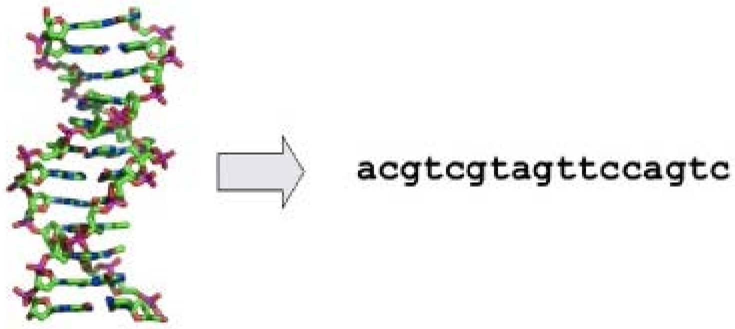
In this article I want to introduce this exciting field and set the scene for the articles that will follow.

What is Bioinformatics?

When molecular biologists started to generate DNA sequence data 27 years ago, it was natural that computer scientists and mathematicians would take a keen interest. Here in the messy, wet, analog world of biology was digital information: a linear string of four chemical groups encoding the entire blueprints for the protein machinery of the living cell. How could you not be interested in cracking that code?

This field of study gained a real identity, and the name bioinformatics, in the mid-1980s, as DNA sequencing became a fundamental tool for molecular biology and sequence data started to appear in significant volume. Right from the start, three concepts emerged that remain central to bioinformatics today.

The first is data representation. The DNA in the human genome is not neatly arranged in the pristine double helix we all recognize. It is coated with proteins that bind to specific sequences, which untwist the helix to allow gene expression and wind it up into tightly packed supercoils. Far from being a static archive of blueprints, DNA is a complex, dynamic, three-dimensional molecule. And yet we represent all of this as a simple string of the characters A, C, G and T.



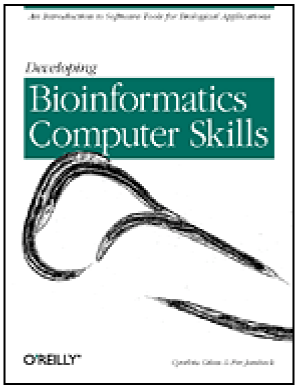
This is a remarkable abstraction. Most of the processes involving genes that we know about have been discovered using this grossly simplified representation of reality. It is the perfect representation for computer analysis, and without it we could never have approached a project on the scale of the human genome.

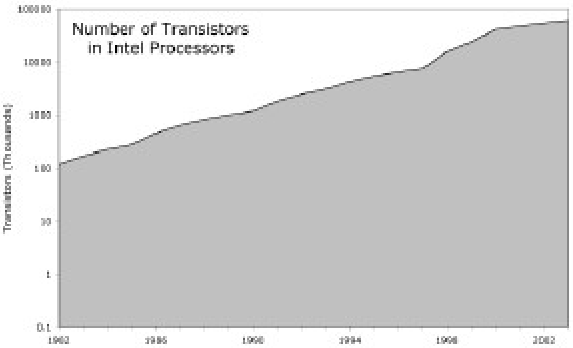
Second is the concept of similarity. Evolution has operated on every sequence that we see today. It conserves genes that encode important proteins and sequences that are involved in gene regulation. Sequences that encode useful functions are transferred, like code modules, from one organism to another. Because of evolution, similar sequences have similar functions.

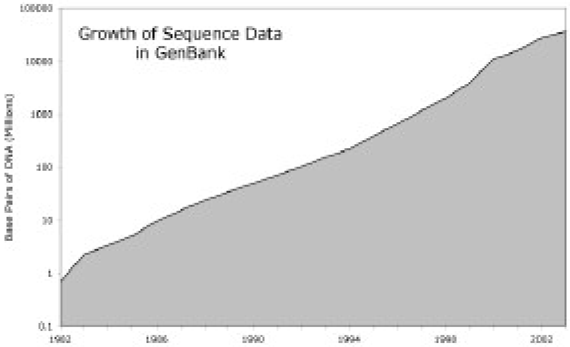
Algorithms for comparing sequences and finding similar regions are at the heart of bioinformatics. At many different levels, they are used to find genes, determine their functions, study their regulation and assess how they, and entire genomes, have evolved over time.

Third is the reality that bioinformatics is not a theoretical science; it is driven by the data, which in turn is driven by the needs of biology. Relatively few researchers have the luxury to develop algorithms and theories in the traditional academic sense. Most people are fully consumed in the day-to-day management and analysis of data.

We have a *lot* of data. The introduction of automated DNA sequencing in the early 1990s created what was, at the time, a torrent of sequence data. But it was the Human Genome Project, with its massive automation, production lines, and money, that really opened the floodgates in the past few years. Compare the rate of growth of sequence data in GenBank, the NIH sequence database, to Moore's Law, that well-known measure of technical advancement, and you will appreciate the challenge facing biology.



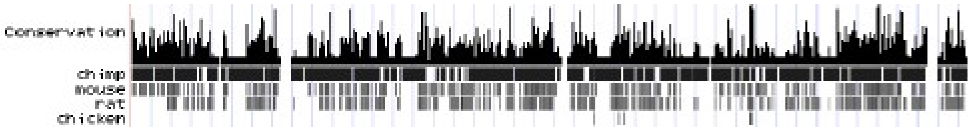




And those are just the sequences! Microarray technologies, able to measure the expression of thousands of genes in a single experiment, have developed over the past decade and now produce huge amounts of data. New techniques for looking at genetic variations in large human populations, and for identifying interactions between sets of proteins in cells, are pouring data onto file servers around the world. Bioinformatics is charged with managing and making sense of all of the data, keeping pace with both data production and technology development. There's plenty of work to go around.

What are the Hot Topics in Bioinformatics?

Comparative Genomics

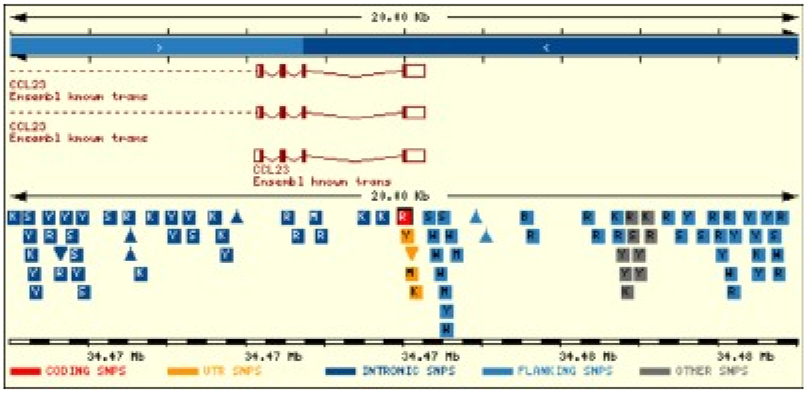


Edited screenshot taken from the UC Santa Cruz Genome Browser (genome.ucsc.edu)

This has the highest profile, thanks to the Human Genome Project. It has been, and still is, the focus of a huge amount of work. The first "tier" of genome sequences (human, rat, mouse, and fruit fly) is now complete, and the big sequencing labs are moving on to organisms like the chimpanzee, rhesus macaque, cow, chicken, and sea urchin.

Why this huge effort to sequence the entire contents of the zoo? Comparative genomics: the same approach to biology used by Charles Darwin, but based on sequences instead of the beaks of finches. By comparing the genomes of related species, we can learn a tremendous amount about how genomes are organized and how major evolutionary changes takes place. At the level of individual, genes we can uncover novel mechanisms for regulation that were hidden when we just had one sequence to work with. Similarity is everything!

Single Nucleotide Polymorphisms (SNPs)

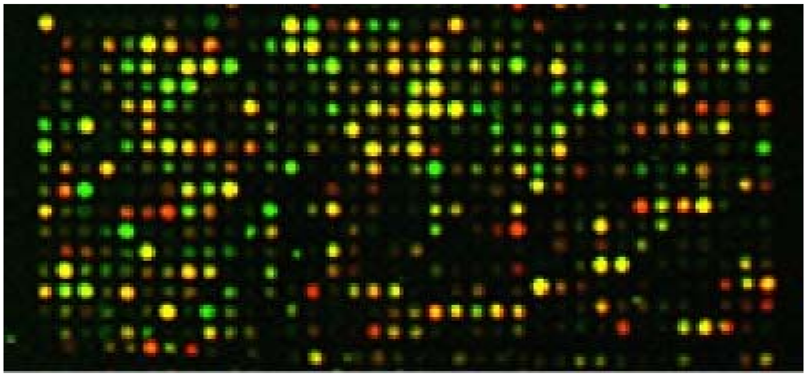


Edited screenshot taken from Ensembl SNPView (www.ensembl.org)

Another avenue that opens up once we have a "reference" human genome is the study of sequence differences between individuals -- the fine details: what makes you different from me. It turns out that the genome is full of single nucleotide differences, called *polymorphisms*, or SNPs for short. Most of these have no direct impact on anything. But their distribution throughout the genome, their frequency in the human population, and their patterns of inheritance make them extremely useful markers for differences that we do care about. By measuring sets of SNPs in thousands of individuals and correlating them with the incidence of a disease, we can identify which regions of the genome are involved and eventually pinpoint the genes themselves.

The combination of these molecular assays with large clinical studies of populations generates huge amounts of data and a whole new set of challenges for bioinformatics.

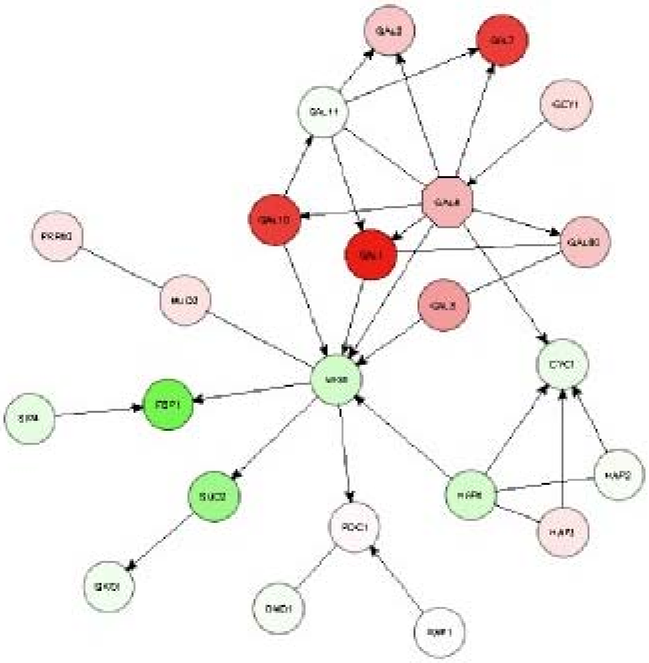
Microarrays



Section of a microarray image, courtesy of Eric Jeffery, Corixa Corporation

Microarray technologies show us which genes are turned on in different cell types in different circumstances. In response to infection, for example, certain cell types will express sets of genes and synthesize certain proteins that respond to the stress. Messenger RNA (mRNA) is like a photocopy of a blueprint that is used in the shop to build a specific type of protein. In a microarray, we can attach sequences from a range of genes to a glass slide in a series of dots, and then bind the mRNA extracted from a population of cells and measure how much binds to each dot. That gives us a snapshot of which genes are being expressed at any given time. Compare the patterns for mRNA from, for example, normal breast tissue and from a breast tumor, and you can identify proteins that are only present in the tumor. Those proteins are potential targets for cancer treatments, vaccines, and other therapeutics.

Systems Biology



Edited screenshot taken from CytoScape tutorial (www.cytoscape.org)

The genome gives us all of the genes in an organism, and microarrays tell us which subset is expressed in a particular biological process. Now the bottleneck in understanding biology is shifting to the world of proteins and the interactions between them. The traditional approach of dissecting out individual interactions with the help of mutations and inhibitors just doesn't scale. That is where systems biology comes in with a slew of novel technologies aimed at seeing the big picture of everything going on in a cell.

New advances in mass spectrometry have allowed this established chemical analysis technology to identify the components of complex mixtures of proteins. Inventive chemical labeling techniques provide insight into the transient interactions between different proteins in the cell. This bundle of new technologies is called *proteomics*.

The integration of all of these results with gene expression data and the collective knowledge of cell biology, contained in the scientific literature, becomes another huge challenge. This is leading to exciting work in textual analysis, pathway modeling, and network visualization.

Structural Biology

Nitrogenase structure 1CP2 displayed in MacPyMol (pymol.sourceforge.net)

While our abstraction of the DNA sequence works remarkably well, in the world of proteins the nuances of three-dimensional structure are everything. Structural biologists determine the structure of proteins using X-ray crystallography and nuclear magnetic resonance, a slew of heavy numerical methods, and a lot of computing. This is a huge field in its own right that predates bioinformatics by several decades. It focuses on the details of structure, the dynamics of molecular motion, and the specific interactions with drugs and other proteins. Bioinformatics, with its focus on huge volumes of data, has often had an uneasy interface with structural biology; "quantity versus quality" some might say, but that distinction is becoming every more blurred as all of these data sources become more integrated.

Software in Bioinformatics

Two main factors have shaped the current landscape of bioinformatics software. As already mentioned, the field has been driven by the massive amount of data and the research projects that generate it. As a result, most people in bioinformatics work on very focused projects and few have the luxury to sit back and write the ideal program for gene prediction, for example.

In addition, the technologies used in the lab, and the data they produce, have evolved very rapidly. That has made it very difficult to commit a lot of resources and time to specific pieces of software. The lifespan of a software project is often quite short and the lead time before deployment is minimal. Being able to understand the essence of a problem and hack up a quick solution that gets the job done are critical skills for a good bioinformatics developer.

A classic example is the genome assembler written by Jim Kent at UC Santa Cruz. Excellent software already existed for assembling the fragments of data produced by sequencing instruments into large blocks, but it could not handle the scale of the task that the Human Genome Project had created. Rather than try to modify existing code, it made sense for Kent to start from scratch and build something, in very short order, that was tailored to the task at hand. More than a quick hack, but a lot less than a complete, polished product, Jim's software assembled the human genome.

Refined, mature software packages usually emerge from research groups with a direct bioinformatics focus, as opposed to playing a support role in, say, a genome center. Of all of the software out there, the "killer app" in bioinformatics has to be BLAST, the suite of sequence comparison tools from NCBI, the National Center for Biotechnology Information at the NIH. The BLAST team built a very fast sequence-comparison engine that could search the entire contents of GenBank in seconds. Over the years, they have improved performance and extended their algorithms, but have always retained their focus on what they do well. As a result, every molecular biologist that has ever looked at a sequence has used the NCBI BLAST server.

What Role Is There for Mac OS X?

Molecular biologists have had a long history of involvement with the Mac, in part from a natural gravitation to the platform but undoubtedly helped by Applied Biosystems' choice of the Mac as the front end to its DNA-sequencing instruments. That changed in the mid-90s as the Windows interface improved and the price/performance ratio shifted in favor of the PC platform. Computer scientists and developers coming into bioinformatics, on the other hand, were used to using Unix. The rise of Linux has locked that preference firmly in place.

Mac OS X has the potential to be the ideal platform for bioinformatics development, with Unix under the hood, a great desktop, productivity applications, and integration with Windows systems. Porting existing bioinformatics packages from Linux is usually straightforward, and many are already available from the Fink project. Being able to expose command-line tools to the desktop user in a simple way will broaden their user base dramatically, and Mac OS X provides several ways in which to build this kind of interface.

Getting Involved in the Field

Bioinformatics is a very rewarding area for software developers to work in. There is something for everyone, whether you're into the minutiae of database design, complex user interface design, advanced statistical algorithms, or good old Perl script hacking. The technologies that produce the data you work on are amazing. The data and the biology behind them are fascinating. On top of that, biologists tend to be nice people to be around!

One topic that we will cover in the coming months is how you get started in bioinformatics: what programming skills you need, how much biology you should know, and how to build a lifelong career in the field.

Bioinformatics and the Mac DevCenter

Over the next few months we will cover these topics and others in more detail. Where possible, we'll also include short tutorials that introduce you some of the key software tools used in bioinformatics. These will guide you through analyses of real datasets from the Human Genome Project and elsewhere. They won't make you an expert, but I hope they will spur you in to further explorations of your own. Stay tuned!

Robert Jones runs Craic Computing, a small bioinformatics company in Seattle that provides advanced software and data analysis services to the biotechnology industry. He was a bench molecular biologist for many years before programming got the better of him.

Return to the Mac DevCenter