Crossroads
The ACM Student Magazine

Jump-Starting Your Bioinformatics Career as an Undergraduate: One Student's Approach

by ***Mike Verdicchio***

Introduction

Bioinformatics is a multidisciplinary research field where computer science, applied mathematics, informatics, and statistics are used together to solve complex biological problems [11]. Perhaps the most compact definition would be to say that, "bioinformatics is the science of refining biological information into biological knowledge" in a computational framework [4].

In this article, we will see how an undergraduate student with computer science and engineering background may jump-start his or her training in the field and contribute to the advancement of science, as well as open the door to even bigger contributions in the future. To illustrate this, I will describe my own bioinformatics research experience over a one year period of time. During that time, I developed software to graphically represent gene regulatory networks for biomedical scientists.

Bioinformatics Research: Graphical Representation of Gene Regulatory Networks

A gene regulatory network (GRN) is a group of specific genes in which individual genes influence or regulate the activity of others. GRNs represent an apex in bioinformatics and computational biology, allowing for the potential discovery of triggering mechanisms and treatments for high-profile diseases.

In an effort to graphically represent GRNs, I developed a Java program entitled

"ArrayViz"[10]. While specifics of the program will be discussed later in the article, in general, ArrayViz reads in gene expression microarray data, processes it statistically, and finally displays a graphical representation in the form of an undirected graph. The combination of skills required to complete this process draw from computer science and engineering, mathematics and statistical analysis, cellular biology, and especially the communication and interactions between those disciplines. In the end, the sampling of experience from each of these fields allowed me, as an undergraduate, to construct this tool.

Multidisciplinary Training Required for a Multidisciplinary Project

Through a mentor-guided undergraduate research program at Arizona State University [5], I was able to begin the multidisciplinary learning process in earnest. The following sub-sections detail my learning experiences in the fields of cellular biology, mathematics and statistics, and computer science, as well as their relation to ArrayViz, which is illustrated in Figure 1.

Figure 1: Multidisciplinary makeup of ArrayViz.

Cellular Biology

As a neophyte to cellular biology, I first had to learn the basics of cellular function, DNA, DNA transcription, RNA, genomics, the Human Genome Project, and finally how they all fit together to produce the data set I had in my hands.

I picked up a lot of jargon from various books and articles, but perhaps what helped the most was seeing it all put into practice at the Translational Genomics Research Institute (TGen) [9] in Phoenix, Arizona, where my aforementioned mentor works as a researcher.

The key to producing my data set is the DNA microarray. These microarrays are a large set of cloned DNA molecules spotted onto a solid matrix (such as a microscope slide) for use in probing a biological sample to determine gene expression, marker pattern, or nucleotide sequences of DNA/RNA. One DNA microarray chip can provide expression levels for anywhere from several hundred genes to hundreds of thousands of genes. After the microarray probes are analyzed and expression levels are determined, the final microarray data is made available in spreadsheet form.

Applied Mathematics and Statistics

The first step of the ArrayViz software involves reading in the DNA microarray data and performing some user-directed data transformation and normalization. Transforming and normalizing a data set aids in extrapolating the desired data while reducing statistical noise. Transforming any data set logarithmically, for example, will expand the scale of low numbers while compressing the scale of higher ones. In terms of gene expression data, this transformation will allow more accurate visualization of different kinds of data, since sets with a large range of intensities can be transformed into more usable numbers.

ArrayViz next compares every pair of genes in the data set to determine their influence on one another, be it insignificant or significant. Depending on the number of genes in a data set, this step can be considerable in size, since the number of gene pairs follows the binomial coefficient, or "n choose k." ArrayViz currently allows two different statistical metrics for determining inter-gene influence. In many data sets, the expression readings contain discrete values over a large range with many decimal places. In other data sets, the expression values might be binary (0 or 1) or ternary (-1, 0, or 1) in form. These different data benefit from different metrics for statistical comparison, and ArrayViz currently offers the correlation coefficient and mutual information as metrics of statistical comparison between genes.

Computer Science

Based on the sheer number of gene-to-gene comparisons needed for one graphical representation, it is easy to see why computer science plays such an integral part in bioinformatics in terms of efficiency alone. In the bigger picture, the computational demands of ArrayViz are nearly infinitesimal when compared with many of the jobs researchers need to run on a regular basis. While ArrayViz may pause for one or two seconds during computation, some programs require weeks or even months to process, and these programs run on some of the largest computers in the world. By adhering to fundamental computer science paradigms, ArrayViz was developed efficiently, with special detail paid to execution efficiency.

The urgency to write good code is perhaps best expressed by the resources in which people are willing to invest in order to run large-scale bioinformatics software. Arizona State University and TGen recently partnered up to create one of the fifty fastest and largest supercomputers in the world. The 512-node cluster of IBM eServer xSeries

1350 servers powered by 1024 Intel Xeon processors is housed on the ASU campus. It runs Red Hat Linux and will eventually support hundreds of users and researchers on the system [7], including those from TGen, the Biodesign Institute[2], and the Department of Biomedical Informatics[3].

ArrayViz Functionality

With sufficient training in the areas of cellular biology, mathematics and statistics, and computer science, I was ready to begin work on ArrayViz in earnest. Many aspects of the software have been briefly mentioned to supplement the descriptions of the multidisciplinary aspects of the project. However, in order to offer a clearer understanding of how the software works, I present the following walk-through, which is comprised of six steps. These six steps, summarized in Figure 2, encapsulate the multidisciplinary communication and interaction required for bioinformatics research in the context of the ArrayViz software.

Figure 2: ArrayViz program flow.

Step 1: Microarray Data Input

Once a biologist has completed a microarray-based experiment, he/she converts the data to an electronic format, such as a spreadsheet. In certain free online databases, such as Gene Expression Omnibus (GEO)[6], researchers can access microarray data from countless other experiments. When the desired data has been obtained, the researcher or biologist can hand it over to ArrayViz.

Step 2: Mathematical Data Transformation

We discussed how transforming and normalizing microarray data can be helpful in extrapolating the desired information while reducing noise. In the next step, ArrayViz allows the user to transform the data set using the log-base-two transform.

Step 3: Statistical Gene Comparison

After any desired transformations are complete, the next task is to find any present GRNs. To do this, it is necessary to compare every unique gene pair in the data set using one of the statistical metrics previously mentioned, namely, the correlation coefficient or mutual information.

Step 4: Layout Generated by GraphViz
After the interaction level of each gene pair has been calculated, ArrayViz creates a specially formatted list of the genes that meet the user-specified threshold for inter-gene influence. This list is passed to one of our third-party software components, namely, GraphViz[1]. GraphViz processes the list and returns to ArrayViz a specially coded layout map for our graphical representation of the GRN. This step is completely abstracted from the user.

Step 5: Layout Visualized by jGraph
Once ArrayViz has completed all of its internal computation based on user-specified preferences, and once GraphViz has given us a coded layout, the final output is generated by jGraph software[8] in the form of an undirected graph. The graph is a representation of a GRN, with each vertex representing a particular gene in the data set, and edges connecting gene pairs that directly influence each other. The edges are also colored and scaled to represent the sign and magnitude of the influence. The number of vertices displayed depends on a threshold selected by the user after being provided with several statistics on the data set. In Figure 3, an ArrayViz-generated GRN is shown for a melanoma data set originally containing 587 genes.

Figure 3: GRN for melanoma data set.

Step 6: Graphical Interaction in ArrayViz
In the end, the biologist is face-to-face with a graphical abstraction of patients, doctors, scientists, laboratory work, microarrays, spreadsheets, mathematics and statistics, and computer science. This picture, however, is more of a gateway than an end result. If the biologist sees, for instance, a vertex in the graph that appears to be "holding it all together," he/she might be interested in which gene that vertex represents, and might want to obtain more information. At this point, ArrayViz lets the user discover which gene the vertex represents by simply placing the mouse over it. In the next release of the software, right-clicking a gene will pull up a context menu linking the user to various biological databases where further information on that gene and its functions, as well as countless papers and other reference materials, can be found with relative ease.

Conclusions

At this point I have used the example of ArrayViz to demonstrate how the right mentoring and training can allow undergraduates with a computer science or computer systems engineering background to contribute to the field of bioinformatics. I have traced multidisciplinary training through each of the fields needed to begin the research, and have shown how that training was put into action with the example of the ArrayViz software. Hopefully, however, I have also shown the scope of the bioinformatics field, and that GRNs represent only one area in this vast and ever-evolving discipline. Perhaps as more and more young researchers realize that they possess much of the required knowledge and many of the required skills to work in this field, many of them will take the initiative in producing new knowledge, which will in turn contribute to the greater good.

Acknowledgments

None of the learning and progress described in this article would have been possible without the patient mentoring received from Dr. Seungchan Kim. His kind, well-timed advice and his consistent testing of my limits has pushed me to places I never knew I could go. Also, the Fulton Undergraduate Research Initiative at the Ira A. Fulton School of Engineering at Arizona State University has been instrumental in facilitating and funding the research project described herein. To that program, its founders, and its staff, I am eternally grateful. A heartfelt thank you is also directed to Mr. Ira A. Fulton for his selfless philanthropy, the Department of Computer Science and Engineering at ASU, as well as the Translational Genomics Research Institute in Phoenix.

References

1

AT&T Labs Contributors. AT&T Labs' GraphViz. 2006. http://www.graphviz.org/ (accessed 07 July 2006).

2

Biodesign Contributors. The Biodesign Institute at Arizona State University. 2006. http://www.biodesign.asu.edu/ (accessed 07 July 2006).

3

BMI Contributors. Department of Biomedical Informatics at ASU. http://bmi.asu.edu (accessed 07 July 2006).

4

Draghici, Sorin. 2003. *Data Analysis Tools for DNA Microarrays*. Boca Raton: Chapman & Hall/CRC.

5

FURI Contributors. Fulton Undergraduate Research Initiative. 2006. http://www.fulton.asu.edu/furi (accessed 07 July 2006).

6

GEO Contributors. Gene Expression Omnibus. 2006. http://www.ncbi.nlm.nih.gov/geo/ (accessed 07 July 2006).

7

Intel Contributors. TGen/ASU Advances Genomic Research with High-Performance Cluster based on Intel Processors and Linux. Intel Business Computing Case Study. 2003. (accessed March 25, 2006).

8

jGraph Contributors. jGraph Ltd., jGraph Software. 2006. http://www.jgraph.com/ (accessed 07 July 2006).

9

TGen Contributors. Translational Genomics Research Institute. 2006. http://www.tgen.org (accessed 07 July 2006).

10

Verdicchio, Mike. ArrayViz Software. 2006. http://www.public.asu.edu/~mverdic/arrayviz/ (accessed 07 July 2006).

11

Wikipedia Contributors. Bioinformatics. 2006. http://en.wikipedia.org/w/index.php?title=Bioinformatics&oldid=42672144 (accessed March 9, 2006).

Biography

Mike Verdicchio graduated from the Ira A. Fulton School of Engineering at Arizona State University in May of 2006 with a degree in Computer Systems Engineering. As an undergraduate student under the mentorship of Seungchan Kim, PhD, Verdicchio performed bioinformatics research under the Fulton Undergraduate Research Initiative at Arizona State.