"Extreme Programming" in a Bioinformatics Class

Scott Kelley, Christianna Alger*, Douglas Deutschman

San Diego State University,5500 Campanile Dr. Mail Code 1153
San Diego, CA 92182
Email: calger@mail.sdsu.edu

*Corresponding author

Abstract: The importance of Bioinformatics tools and methodology in modern biological research underscores the need for robust and effective courses at the college level. This paper describes such a course designed on the principles of cooperative learning based on a computer software industry production model called "Extreme Programming" (EP). The classroom version of EP included: working in pairs, switching roles between labs, partner interdependence and individual accountability. New pairings were created at random each week and at the completion of each lab, students (n=18) indicated their satisfaction and frustration levels with working with partners, the materials, and the technology. We used a Repeated Measures-ANOVA (RM-ANOVA) statistical design to provide statistical power with a modest number of subjects. Students consistently rated working with a pair highest in terms of both ease and satisfaction, regardless of prior programming and technology experience. We found no differences in reported ease or satisfaction between undergraduate and graduate students, or between students with prior experience with technology. Surprisingly, we found that students rated the more difficult computer programming part of the course higher than the web-based exercises. The Extreme Programming cooperative model appears to be very appropriate for Bioinformatics classes, and can be easily implemented in computational labs to enhance student satisfaction and potentially maximize the use of computer workstations.

Keywords: bioinformatics, Python, analysis of variance (ANOVA)

Introduction

Bioinformatics has become an integral facet of modern biological research. Academics and biotechnology companies rely heavily on a vast assortment of bioinformatics tools to analyze a virtual flood of biological data, from genome sequence to x-ray crystal structures, being dumped into computer databases (Kaminski, 2000). Bioinformatics tools are used to perform DNA and protein sequence searching (Altschul et al., 1997), sequence alignment (Chenna et al., 2003), molecular structure prediction (Akmaev et al., 1999; Chivian et al., 2005), evolutionary relationship analysis (Ronquist and Huelsenbeck, 2003), gene expression (Slonim, 2002), and many other applications to generate or test hypotheses. The recent development of simple, yet powerful, programming languages (e.g., Perl and Python) has also opened the door for biologists with little formal computer science education to develop functional bioinformatics software (Gentleman et al., 2004). Biotechnology companies have invested heavily in bioinformatics research, and scientists trained in bioinformatics software tools and/or programming are often hot commodities in the biotechnology industry.

The importance of bioinformatics tools and methodology in modern biological research underscores the need for robust and effective courses in college level bioinformatics. In our experience, however, the typical biology student has limited exposure to computational biology and little or no programming background. Indeed, we often find that both undergraduate and graduate biology students express some distaste for computer work. Given the increasing emphasis placed on bioinformatics and technology in biological research, it is therefore important to provide an educational experience that maximizes learning and fosters student motivation.

In computer labs at the college level, students typically work on their own computers to learn software or write programming code. This is true of all the biology computer lab courses (e.g., bio-statistics, conservation ecology, and population genetics) at San Diego State University where the study took place. However, numerous studies of cooperative learning have clearly shown the advantages of working in pairs or groups in terms of both learning outcomes and interest levels for science and mathematics courses. Slavin (1996) described cooperative learning as 'one of the greatest success stories in the history of educational research' (p. 1) because so much research has tied cooperative learning to achievement gains. Slavin's review of 99 studies on cooperative learning and achievement in K-12 school environments found that

78% of the cooperative learning groups outperformed the control groups in terms of student achievement. In their meta-analysis of studies on cooperative learning in science, mathematics, engineering and technology (SMET) courses at the college level, Springer, Stanne, and Donovan (1999) found significant positive effects on achievement, persistence and attitude in students engaged in small learning groups compared to students who were not. They estimated that the effect of small group learning on achievement would increase a student's grade on a standardized (norm referenced) test from the 50th to the 70th percentile and the effect of group work on increased student persistence would reduce attrition from SMET courses and programs by 22%.

Given the clear potential benefits of cooperative learning, our aim was to develop and evaluate a novel cooperative learning approach for bioinformatics at the college level. In this study, we focused on the effectiveness of cooperative learning on student motivation, per se, rather than on learning. Motivation appeared to be a particular concern with biology students not naturally inclined towards computer work, and the students scored highly on all the course exams this semester and in previous years, indicating that they had mastered basic Bioinformatics concepts. We based our cooperative learning approach on a new software development model used in the computer industry called 'Extreme Programming' (EP). The EP model, described as a 'deliberate and disciplined approach to software development' (Wells, 2001), is characterized by a set of simple rules and practices associated with all phases of development from planning to execution. What makes this model different from others is that programmers work in pairs, with several pairs working to find solutions to the same project/problem or pieces of the problem. The process stresses communication and teamwork and appears ideally suited for a hands-on bioinformatics lab course, in which students could be paired at a single computer.

EP claims several key advantages to solo programming approaches: 1) increased problem-solving capacity; 2) higher likelihood and greater rapidity of error-catching; and 3) more engaging and productive work experience. These touted advantages in workplace productivity appear remarkably similar to the educational benefits observed in cooperative group learning approaches.

Many instructors assume that when students are working in groups or with partners that the students are engaging in cooperative group work. In fact, to reap the benefits of group work, attention to the structure of the group and the type of task required is critical. According to Johnson and Johnson (1994), cooperative learning has four basic elements: 1) group members work toward a common goal, resulting in interdependence; 2) students interact to solve problems; 3) a component of individual accountability is built in to the lesson or course to assure that all students master the content being taught; and 4) interpersonal and small group skills are developed. Cohen (1994) added two more necessary elements. First, all individuals must have opportunities to hold high status academic positions, such as facilitator. And secondly, for maximum learning to occur, the task assigned to groups should be open-ended, meaning that a variety of solutions are possible, and difficult enough so that students experience a 'healthy level of uncertainty'.

The structure of the bioinformatics class run by one of the authors (Kelley) was designed to encompass almost all of the requisite elements of effective group work. Interdependence was established by having both members of each pair earn the same grade for each lab. The success of one student was determined by the success of the partnership. The students were provided considerable opportunity to talk face-to-face to solve problems. In addition to group grades, each student took quizzes and wrote papers independently, creating individual accountability. Each pair worked together on two labs a week and they shared a computer to accomplish each task. One student worked at the computer while the other observed as they problem-solved. The students were required to switch roles for each lab. In the first half of the course, students learned how to use a series of complex, but highly useful, bioinformatics tools for analyzing biological data. In the second half of the course, the students were taught the fundamentals of computer science in the Python programming language and applied this language to the analysis of sequence data. These first labs were more 'cut and paste' as opposed to the labs in the second half of the semester, which were open-ended, and by the students' own admission, more difficult.

After designing the course based on best teaching practices, we developed a survey given after every lab to answer to the following questions:

1.) What was the satisfaction of working with a partner relative to lecture and technology?
2.) How effective was the paired learning approach under increasingly high levels of uncertainty?
3.) How did past experience with technology and student grade level (undergraduate or graduate) affect the learning experience?
4.) Did a decrease in comfort level with the material or the technology decrease satisfaction of working with a partner?

Due to the limited number of student respondents, we used a statistical design known as a

Repeated Measures ANOVA (RM-ANOVA; see Materials and Methods), a methods routinely used with studies including small sample sizes, such as clinical trials. Statistical analysis of survey responses answered all of the above questions in a straightforward manner and helped us determine the effectiveness of the EP cooperative learning model for Bioinformatics.

Materials and Methods

Data Collection and Participants

Data were collected using lab evaluation surveys (Table 1) during S. Kelley's bioinformatics course in the spring of 2005 at San Diego State University. The course participants included 8 female students and 10 male students (45% female). Of these, 11 out of 18 students (>60%) had non-European ancestry, and 7 were undergrads, while the rest were Master's students. The course was taught in a "lecture/lab" format. Prior to the lab, the teacher (Kelley) would teach a lecture on the algorithms or concepts underlying the particular exercise. For example, in the non-Python section the students might be taught a DNA sequence comparison algorithm and then use the algorithm to compare two novel sequences on pen-and-paper. In the Python section, the students might be taught a basic programming concept, such as the logic behind an "if/else" statement. Following this short lecture and exercise (usually lasting about 30-45 minutes) the students would then pair up at a computer and complete an exercise written by the instructor related to the lecture material. After the lecture on sequence comparison, the students would complete a lab exercise using web-based software implementing the algorithm for comparing two sequences, and after the "if/else" lecture, the students would write a Python program that used "if/else" statement.

Table 1. Sample survey completed by students after each lab.

Name or Red ID ____________ Partner Name ___________Lab # ____ Date _____
Place an X next to your student status: Undergraduate ___ Graduate ___ student.

I. On a scale of **1 (extremely frustrating)** to **10 (not frustrating at all)** rate your frustration level with elements of the lab. Please write the rating in the space provided.

Extremely Frustrating **Not Frustrating at all**
1 2 3 4 5 6 7 8 9 10
_____ material being studied _____ working with a partner _____ technology

II. On a scale of **1 (extremely dissatisfying)** to **10 (very satisfying)** rate your satisfaction with the lab experience. Please write the rating in the space provided.

Extremely Dissatisfying **Very Satisfying**
1 2 3 4 5 6 7 8 9 10
_____ material being studied _____ working with a partner _____ technology

III. Place an **X** next to the statement that best describes your familiarity with the software
_____ I am very familiar with the software used for this lab.
_____ I am not familiar with the software, but have successfully used similar software.
_____ I am not familiar with the software.

IV. Is there anything else you would like to communicate about your lab experience?

After each lab students were asked to complete a short survey indicating their level of ease and their level of satisfaction with the study material, the computer technology, and their partner. The "material" part referred to the written exercise the students worked on with the partner at the computer, while the "computer technology" referred to the web-based software or the Python programming environment. An example of the survey is shown in Table 1. The surveys were placed in an envelope which was stored unopened until the end of the semester after all the grades for the course had been assigned. Students were assured that no one would look at the survey results until after assignment of final grades.

Statistical Methods

We used one-way ANOVAs to test for significant difference in over all mean scores among labs, between undergraduate and graduate students, and between students with previous experience or no previous experience in overall mean scores. Survey scores were also analyzed using a 3-way RM-ANOVA. RM-ANOVA methods provide a powerful means of providing statistical power with a modest number of subjects. Many published RM-ANOVA designs use modest numbers of subjects. Case studies provided by Quinn and Keogh (2002) include samples sizes comparable to the present study: n=12, 20 and 24 subjects. According to Quinn and Keough, "The main aim of these [RM] designs is to reduce the unexplained variation (MS residual) …They offer more powerful tests of the null hypothesis of interest, with no increase in the overall resources needed for the experiment (p.262)." According to Munro (2004), "Each subject [serves] as his or her own control, and the within or error variance [is] decreased. This [results] in a more powerful test and [decreases] the number of subjects needed for the study (Page 214)." The proven ability of Repeated Measure approaches to provide statistical power in studies with modest samples sizes similar to our own, gave us confidence in interpreting our statistical results.

Lab exercises were highly variable in content and were treated as the repeated measures. Data normality and homogeneity of variances were tested and confirmed using graphical methods. We used an Expectation Maximization (EM) algorithm, based on the work of Little and Rubin (1987), to impute missing values in student survey responses. Missing values comprised approximately 15% of the dataset. The EM method used a maximum likelihood approach to estimate the expected values based on the observed data (i.e., student responses for other labs). The 3-factors in the RM-ANOVA included: (1) Lab Type (Non-Python vs. Python); (2) Education Component (Materials vs. Pairs vs. Technology); and (3) Questionnaire (Ease vs. Satisfaction). Paired T-Tests, in which survey data for each student was kept as a separate response variable, were used to compare mean differences in survey responses overall scores for Material, Partner and Technology. These tests were divided by lab type (Python and Non-Python) and question type (Ease and Satisfaction). The Paired T-Test approach is especially useful for situations with high among-subject variability, such as patients in clinical drug trials.

Results

This study made 96 observations on each of the 18 individuals (6 measures for each of 16 labs). This means that a total of 1728 observations were collected, a sizeable number by any measure and an indication of how Repeated Measure designs allow for strong conclusions with modest subject numbers. The analysis used the average of 8 labs for each metric. Thus we have 16 (size n=8) averages in the RM analysis (288 averages). The averages are more normally distributed than the raw value (central limit theorem) providing better fit to the assumption of normality. One-way ANOVAs found significant differences in overall scores among labs, but no significant differences between undergraduate and graduate students or any effect of previous experience on survey responses. For main effects, we found highly significant differences in the survey responses between Python and Non-Python labs (Table 2: $F_{1,17}$=14.348; P=0.001) and among the different types of educational components (Table 2: $F_{2,34}$=15.906; P<0.001) Materials, Pairs and Technology). We did not find significant differences between the survey response in terms of question type (Ease and Satisfaction). There were also significant 2-way interactions between lab type and educational component (Table 2: $F_{2,34}$=11.728; P<0.001), as well as between educational component and question type (Table 2: $F_{1,17}$=14.348; P=0.001), but not between lab type and questions type. No significant 3-way interactions were detected.

Table 2. Three-way repeated-measures ANOVA on student survey scores.

Repeated Measures ANOVA

Source	Sums-Sq	df	Mean-Sq	F	P	H-F† P
Main Effects						
Lab Type (Lab)[1]	10.893	1	10.893	14.348	0.001	.
Error	12.907	17	0.759			
Educational Component (Comp)[2]	31.812	2	15.906	15.284	< .001	< .001
Error	35.384	34	1.041			
Questionnaire Type (Ques)[3]	1.423	1	1.423	0.936	0.347	.
Error	25.846	17	1.520			
2-way Interactions						
Lab * Comp	5.819	2	2.910	11.728	< .001	0.001
Error	8.435	34	0.248			
Lab * Ques	0.680	1	0.680	3.083	0.097	.
Error	3.751	17	0.221			
Comp * Ques	6.973	2	3.486	3.518	0.041	0.041
Error	33.694	34	0.991			
3-way Interaction						
Lab * Comp * Ques	0.482	2	0.241	1.850	0.173	0.173
Error	4.427	34	0.130			

[1] **Lab Type** (Python, Non Python)

[2] **Educational Component** (Material, Pairs, Technology)

[3] **Questionnaire** (Satisfaction, Frustration)

† Huynh-Feldt corrected P value

Plots of 4 individual student responses illustrated the tremendous student variability in survey responses over the course of the semester (Figure 1). Paired T-tests found significant differences in the mean responses for Materials, Pairs and Technology in both Python and Non-Python labs (Fig. 2, 3). In general, the scores for Pairs were highest, followed by Technology then Materials. However, Technology and Pairs scored almost equally well in their Satisfaction scores for the Python labs and students also found the Non-Python technologies less satisfying than the lecture materials for the Non-Python labs. Figure 2 shows a transition graph for all 18 students, along with the mean scores and standard errors, for one of the Paired T-tests (Non-Python, Satisfaction survey scores), while figure 3 reports the mean responses for all the Paired T-tests without individual student responses.

Figure 1. Graph showing the Satisfaction scores for four representative students for all 16 labs. This subset of students spans both the Grad/Undergrad and the Level of Familiarity before the class. The chart illustrates the considerable variability among students and labs.

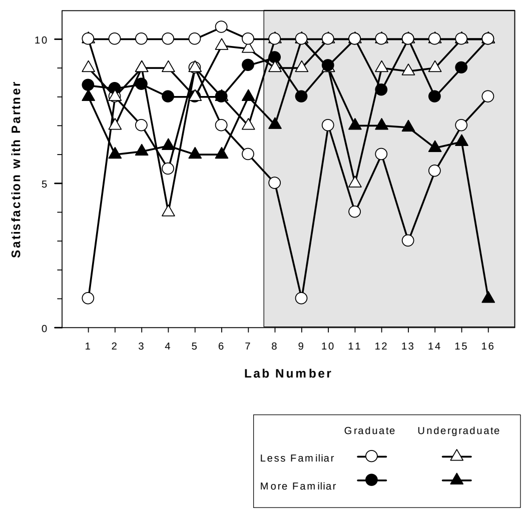
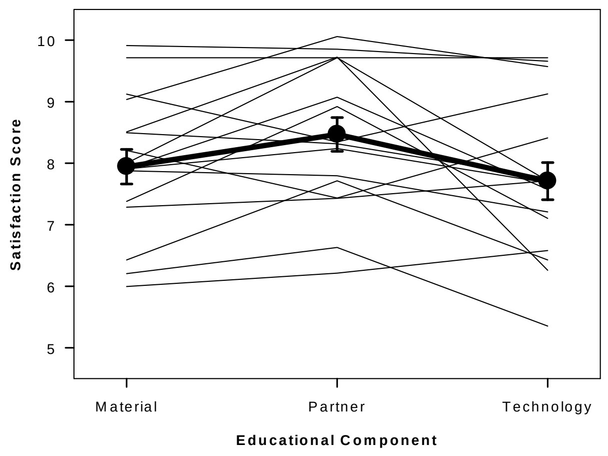


Figure 2. Transition graph showing average Satisfaction with NonPython Labs. Responses of all 18 students are represented by the thin lines, and the thick line connects the mean and standard errors for the groups, indicating how they differ among Materials, Partner and Technology.



Discussion

The survey was a highly sensitive indicator of student frustration and satisfaction with the course, despite the apparent simplicity of the survey design. Most of the students scored all aspects of each lab above 50%. However, within this range there was a considerable variability and strong differences among both students and labs in terms of survey scores. Figure 1 illustrates the typical responses of four individual students over the course of the semester. As expected, there was highly significant lab-to-lab variability, which reflected the wide diversity of exercises presented to the students, particularly in the Non-Python exercises.

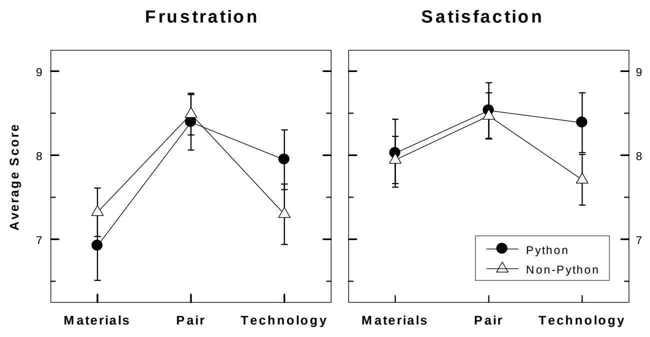
Encouragingly, we found no differences in responses between students with or without previous experience with the technology (either the web tools or programming experience) or between undergraduates and graduate students. This finding mirrors personal observations made by the instructor in the course. Many of the bioinformatics novices were just as good with the bioinformatics tools and at programming as the 'experts' and the undergraduates performed as well on tests as did the graduate students.

The RM-ANOVA found highly significant differences between student responses to the lecture material, the technology and working with a partner (Table 2). A closer look at the data using a paired T-test identified the strongest trend in the study: students consistently rated working with a partner highest in terms of both Ease and Satisfaction, and in both Python and Non-Python labs (Figure 3). Clearly, the aspect of collaboration was highly valued and the EP cooperative model appears to be very appropriate for Bioinformatics classes. Although we did not directly evaluate the impact of the EP model on student learning, cooperative learning has a long track record of boosting student achievement in SMET courses and student satisfaction is also correlated with performance. From an instructor perspective, there were two other enormous advantages to using the EP model. First, the students had someone to help troubleshoot problems, reducing their reliance on the instructor to answer questions. Second, EP effectively doubles the number of students who can take the course, which is an important concern given the extremely limited computer resources on campus.

The RM-ANOVA analysis also uncovered highly significant differences in student responses between Python and Non-Python labs (Table 2). The survey results appear to reflect the extremely different character of the material taught in the two halves of the course. Somewhat to our surprise, we found that these biology students tended to favor the Python programming section of the course over the Non-Python section (Figure 3). Although many of the students expressed trepidation about the programming section prior to the start of the course, and many expressed considerable frustration about programming during lab exercises, overall they seemed to find working the simpler 'cut and paste' labs more frustrating than the open-ended Python labs. This would suggest that students want to engage in more demanding work and that doing so with a partner enhances the experience.

The students appeared to be especially pleased with the Python technology (Figure 3), a fact that speaks well for the Python language as a learning tool since many students had no prior programming experience. The high overall survey scores appeared to confirm the students' general interest in bioinformatics and programming per se. This is especially important in light of the fact that 45% of the class was female and 60% had non-European heritage. Indeed, many of the students appeared to have considerable latent abilities with programming (S. Kelley, pers. obs.) and apparently just needed an opportunity and the right environment to express their talents. We suggest that the non-competitive EP cooperative learning model, combined with easy syntax of the Python programming language and an interesting application (Biology), opens the door to computer education for students who otherwise might never try such a class.

Figure 3**.** Transition graph showing the average scores with standard error bars for Materials, Pairs and Technology. The scores are broken down by Ease and Satisfaction, as well as by Python and Non-Python labs.



Even though we have 30 years of research on the positive effects of group-work (Slavin, 1995), higher education has 'yet to respond to calls for greater opportunities for collaboration and cooperation in SMET (science, mathematics, engineering and technology) courses and programs' (National Science Foundation, 1996). Professors continue to implement teacher centered teaching styles that focus on transmitting knowledge to passive learners. This traditional lecture model of teaching does not engage students or reflect what it is scientists will be expected to do once they enter the workforce, whether it be on a campus or out in the field (Arch, 1998; Springer et al., 1999). By adapting the Extreme Programming model to the bioinformatics class, we believe we have created a student-centered class that required the learner to engage with the material and his or her classmates. In the process of learning the content, the students learned the value of collaboration in problem-solving that will be needed in the workplace.

Limitations of the study and directions for further research

The two greatest limitations of this study were the lack of a control or comparison group. Given the small number of students enrolled in the course, we believed that dividing the class into experimental and control groups would have yielded insufficient data and would have deprived students of potential benefits of cooperative learning. Also, there already exists a plethora of research connecting achievement and cooperative learning, so the first priority was to establish a cooperative learning model that works well in the context of a bioinformatics class (or college computer classes in general). Controls are certainly the best way to judge learning and achievement, per se, but the focus of this study was on the effect of cooperative learning strategies on student motivation in a computer class. Thus, we plan to carry out a long term study on this continuing course that uses "fortuitous controls", which would be times when someone's partner does not show up for a lab and they are forced to work by themselves. We are still working out the details of how this might be accomplished. Additionally, qualitative data such as student interviews and longitudinal follow up with participants could yield greater understanding of the long-term effects of the use of the Extreme Programming model in Bioinformatics classes.

Acknowledgements

We thank the SDSU College of Sciences Computer Support division for their help with the installation of computer software. We also thank the Bioinformatics students for their helpful participation with the surveys.

References

AKMAEV, V. R., KELLEY, S. T., AND STORMO, G. D. 1999. A Phylogentic Approach to RNA Structure Prediction. Proc. Int. Conf. Intell. Syst. Mol. Biol., *7*, 10-17.

ALTSCHUL, S. F., MADDEN, T. L., SCHAFFER, A. A., ZHANG, J., ZHANG, Z., MILLER, W., AND LIPMAN, D. J. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucl. Acids Res. *25*, 3389-3402.

AMERICAN ASSOCIATION FOR THE ADVANCEMENT OF SCIENCE .1989. Science for all Americans: Project 2061, New York: Oxford University Press.

ARCH, S. 1998. How to teach Science. Science, *279*, 1869.

BLOSSER, P,E., 1992. Using cooperative learning in science education, ERIC Clearinghouse for Science, Mathematics, and Environmental Education, http://www.stemworks.org/Bulletins/SEB92-1.html (accessed on 13 April 2006).

CHENNA, R., SUGAWARA, H., KOIKE, T., LOPEZ, R., GIBSON, T. J., HIGGINS, D. G., AND THOMPSON, J. D. 2003. Multiple sequence alignment with the Clustal series of programs. Nucl. Acids Res. *31*, 3497-3500.

CHIVIAN, D., KIM, D. E., MALMSTROM, L., SCHONBRUN, J., ROHL, C. A., AND BAKER, D. 2005. Prediction of CASP6 structures using automated Robetta protocols. Proteins *61 Suppl 7*, 157-166.

COHEN, E.G. 1994. Restructuring the classroon: Conditions for productive small groups. Review of Educational Research, *64*, 1-35.

GENTLEMAN, R. C., CAREY, V. J., BATES, D. M., BOLSTAD, B., DETTLING, M., DUDOIT, S., ELLIS, B., GAUTIER, L., GE, Y., GENTRY, J., *et al.* 2004. Bioconductor: open software development for computational biology and bioinformatics. Genome Biol. *5*, R80.

JOHNSON, D.W., AND JOHNSON, R.T. 1994. Learning Together and Alone:Cooperative, competitive, and individualistic learning, (4th Ed.) Boston: Allyn & Bacon.

KAMINSKI, N. 2000. Bioinformatics. A user's perspective. Amer. J. Respir. Cell Mol. Biol. *23*, 705-711.

LITTLE, R. J. A., AND RUBIN, D. B. 1987.Statistical analysis with missing data. New York: Wiley.

MUNRO, B., 2004. Statistical methods for healthcare research. New York: Lippincott Williams & Wilins.

NATIONAL SCIENCE FOUNDATION 1996. Shaping the Future: New expectations for undergraduate education in science, mathematics, engineering, and technology. Washington, DC: National Academy Press.

QUINN, G.P. AND KEOUGH. N.J., 2002. Experimental design and data analysis for biologists. Cambridge: Cambridge University Press.

RONQUIST, F., AND HUELSENBECK, J. P. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics *19*, 1572-1574.

SEYMOUR, E. AND HEWITT, N.1997. Talking about leaving: Why undergraduates leave the sciences. Boulder, CO: Westview.

SLAVIN, R. 1995 Cooperative Learning: Theory, research and practice (2nd ed.) Boston: Allyn & Bacon.

SLAVEN, R. 1996. Research for the future: Research on cooperative learning and achievement: What we know, what we need to know. Contemp. Educ. Psy., *73*, 651-653.

SPRINGER, L., STANNE, M.E., AND DONOVAN, S.S. 1998. Effects of small group learning in science, mathematics, engineering, and technology: A meta-analysis. Review of Educational Research, *69*, 21-51.

SLONIM, D. K. 2002. From patterns to pathways: gene expression data analysis comes of age. Nat. Genet. *32 Suppl*, 502-508.