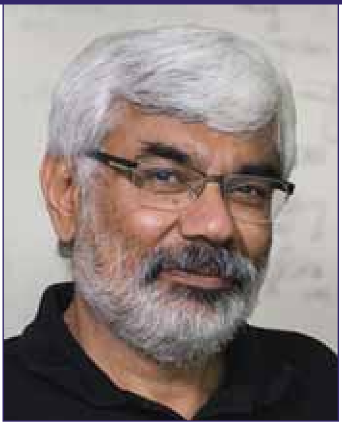
Subramaniam VER



Grant applications for the development and maintenance of software and infrastructure should:

Compete with basic research applications

Biomedicine has a strong tradition of being an experiment- and phenomenology-driven science. But over the course of the last century, the field has become both more quantitative and more interdisciplinary. And in the late part of the 20th century, it also became highly data-driven, owing in large part to genome sequencing and the emergence of high throughput and computer technologies that could measure components of living systems at large scale. Meanwhile, large advances in computers and computational methods inspired confidence in our ability to deal with biomedicine as a quantitative science. In other words, biomedicine began to be viewed as a constructionist science where data and knowledge could be integrated to reconstruct models of processes in living systems. This tight integration of biomedicine with computation and informatics is the fundamental reason grant applications for the development and maintenance of software and infrastructure should compete with basic research applications and be considered by study group panels that have sufficient biology expertise. I would also point out that:

> "Competing with basic research applications will keep the applicants 'honest' in learning the devil-in-the-details biology that serves as the underpinning for the software," says Subramaniam.

1. Biomedical research is highly context-specific and in order to be considered mainstream, software and infrastructure proposals should be biology-rich, useful and have content as opposed to mere form. Competing with basic research applications will keep the applicants "honest" in learning the devil-in-the-details biology that serves as the underpinning for the software.

2. While computer professionalism and a high degree of digital sophistication is needed to generate "good-practices" software and tools, sophisticated software that does not embed biomedicine integrally is at best irrelevant. It is easy to see that one of the most-utilized software infrastructures, namely *GenBank*, did not begin with best-practices software engineering paradigms.

3. The price of software engineering is often cited as a reason for a separate funding mechanism since traditional biomedical research salaries are modest compared to those of computer professionals. Again, some of this is true, but most biomedical researchers appreciate aspects of this argument and in itself this does not warrant a separate consideration for software engineering proposals.

4. The most important argument that has an intellectual basis is the fact that most biomedical researchers, while familiar with basic bioinformatics, are not computer science-savvy and hence would find it difficult to both assess and accept a higher proportion of software proposals. While acknowledging aspects of this argument, in the long run, having a "two-culture" paradigm would be detrimental for strong integration between computational sciences and biomedicine.

5. A significant argument that has been made repeatedly is that most basic research applications are "hypothesis-driven" while proposals that are computational and software in origin are "knowledge/discovery-driven." It remains to be seen whether researchers will accept that analysis of data and incorporation of biological legacy knowledge to generate novel hypotheses is as good as, if not superior to, hypotheses that are stated based on experience and intuition. This question warrants significant discussion and in itself is not a reason for considering software and infrastructure proposals outside of basic research applications.

None of the arguments above are intended against the need for and importance of allocating significant resources for further developments in computational/software/informatics research. However, they are meant as a challenge to the computation, data and informatics-inclined researchers to embrace biology in a more intrusive manner and become polymaths who can participate in building next-generation biomedicine. □

DETAILS

Shankar Subramaniam, PhD, professor of bioengineering at the University of California, San Diego

SEND US IDEAS Got your own opinions on this topic? Or have another topic you'd like to write about for these pages? Send us your thoughts on the Feedback page of our Web site: http://biomedicalcomputationreview.org/feedback.html.

SUS Stiles

Grant applications for the development and maintenance of software and infrastructure should:

Have a separate funding mechanism

Although I agree with many of my esteemed colleague's points—especially the need for strong biological input into the review process—I disagree with his conclusions.

Enormous challenges and opportunities confront us. The growing overlap of structural biology, molecular biology, cell biology, genetics/genomics, synthetic biology, and systems biology will lead to a new era of quantitative physiological understanding, with breakthrough advances in preventive medicine, drug design, and medical interventions of all kinds. Computation is now inseparable from all of these fields, and new hardware developments (e.g., multicore and specialized processors) are increasing the complexity of computer engineering, adding even more complexity to software development. This, together with the scope of scientific problems, has changed the landscape of scientific computing dramatically. In general, the NIH and NSF, despite their roles as primary academic funding agencies, have yet to fully address the human and monetary expense of large-scale/high-performance software development, maintenance, and training, despite centers devoted to biomedical computation, supercomputing, and an often-stated desire for "hardened," "integrated," and "near-commercial quality" programming. Even the DOD, DOE, DARPA, and the national laboratories, with a longer-standing focus on large-scale computation (albeit often classified), are struggling to find effective mechanisms of support for increasingly complex software and computational infrastructure.

"Shoe-horning development and maintenance of software and infrastructure into long-standing evaluation and funding mechanisms is not the answer," says Stiles.

Further compounding the biomedical problem, many experimental investigators still have no real frame of reference for software development and computing. Lab and home computers are seen largely as appliances, and software, regardless of origin, is expected to either run in a web browser applet or be downloadable, instantly installable, and intuitively usable. Satisfying these expectations is difficult even for large commercial software and computing firms. For research projects, this is feasible only in a limited number of cases, and at the cutting edge can lead to counterproductive expectations from potential users. The future of biomedical computation spans enormous ranges of space and time, and lies at the interface of biology, engineering, chemistry, physics, mathematics, and computer science. Far-sighted multidisciplinary development is critical to success, and suitable mechanisms of evaluation and support must be developed and maintained in and of themselves.

Shoe-horning development and maintenance of software and infrastructure into long-standing evaluation and funding mechanisms is not the answer. Alternatives have been tried, including a standing special emphasis panel on software development and maintenance at the NIH. While this is a step in the right direction, it has not yet become a chartered study section primarily because the number of submitted proposals has remained too small. Why? Lack of interest or need? Hardly. Instead it reflects the still incomplete assimilation of leading-edge computing into mainstream biomedical research, and the poor fit of the funding mechanism. All but the smallest projects exceed typical R01 scales, both in time and money. Even at the level of centers, a common point raised during program evaluations has been "underfunded." This is not to say that bigger centers are the answer. No, the answer is probably expanded stable funding at the level of collaborating laboratories, with the primary emphasis on far-sighted multidisciplinary development and less on short-term hypotheses. Regardless, a sustained emphasis and support mechanism is essential. No commercial computing enterprise can survive without professional software and hardware engineers, and thus must provide corresponding salaries and stability. The same is true for the integrated multidisciplinary computational research needed for the future of biomedicine. □

DETAILS

Joel R. Stiles, MD, PhD, director of the National Resource for Biomedical Supercomputing at the Pittsburgh Supercomputing Center, and associate professor of biology at Carnegie Mellon University