GOLDEN HELIX
Accelerating the Quest for Significance™

"Dammit Jim, I'm a doctor, not a bioinformatician!"

Academic Software, Productivity, and Reproducible Research

by Christophe Lambert, CEO & President of Golden Helix

Do you ever feel like Dr. McCoy on Star Trek, where your job and expertise is to do x, but to achieve your goals you also have to do y and z, which you either don't want to do or don't have the skills to do? Genetic researchers are faced with this every day as they are expected to design experiments, develop methods, write software that will perform these methods, teach students, write grants, work towards tenure, make life-changing discoveries, follow statistical best practices, and, in their spare time, publish regularly. What are the causes and symptoms of this reality? And what is the effect these demands have on productivity?

This whitepaper is wide ranging, so I'll first summarize it at a high level and then drill down into the many facets of the systemic problems our field faces for research productivity. We'll then look at the obstacles to productive bioinformatics-driven genetic research through the lenses of skillset, mindset, toolset, and their interactions. Then we'll explore how facets of the academic toolset and mindset are at cross-purposes with reproducible research and explain how this, in turn, inhibits our practice of the scientific method and its concomitant productivity. Within this framework, I will examine intriguing issues such as:

- The degree to which bioinformatics expertise has become rate-limiting for our field.
- Cycle-time and the cognitive impact of inserting bioinformaticians between researchers and their data.
- Consequences of reputation as the prime metric of academia.

Finally, I'll share my thoughts on a solution—making the case that improved tools and practices can dramatically increase productivity for all stakeholders.

Skillset, Toolset, Mindset

Productivity in any field is a function of its skillset, toolset, and mindset. A master carpenter can't accomplish much without his tools. A box full of tools without the skills to use them will not get a house built. And if a skilled carpenter with a great toolset wakes up one morning realizing he never liked carpentry and has been doing it only to please his father, then he doesn't have the mindset to be a productive carpenter in the long run. These three productivity components interact, support, and modify each other. The advent of power tools advanced carpentry and required new skillsets and mindsets to obtain productivity gains.

For those of you not familiar with the constant dillema of Dr. McCoy check out the following youtube video: http://www.youtube.com/watch?v=pGMLCxKPMSE

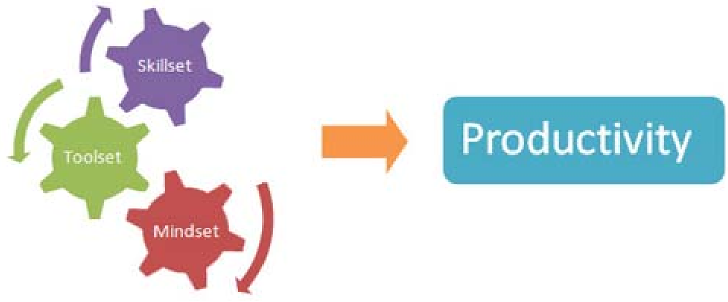
New skills and techniques arose to use the new tools, and in turn, catalyzed toolset innovation to support continuously improving skillsets. The mindset of the artisan also gave way to embracing process and automation.

What, then, does the skillset, toolset, and mindset look like in our field of genetic analysis, and how do they interact to impact our productivity?

Skillset

In academia, the free software tools used today require highly skilled bioinformatics professionals, which are often in short supply. A recent survey of International Genetic Epidemiology Society members (Krishnan & Wilson 2010) showed that 73% reported an inability to find qualified domestic applicants to fill positions. As we will examine later, almost every academic package in genetic analysis requires statistical, programming, command-line, and data manipulation capabilities that fall outside the competency of many small to mid-sized labs and their researchers who seek to utilize genetic information for biological research. Even in the case where these labs have the resources to hire the expertise, it simply may not be available.

To perform research in this field, one must have competence in several disciplines: computer science, statistics, and genetics. Why does someone virtually have to be a computer programmer in order to perform genetic research? We need only to look at the state of our toolset.



Toolset

The NIH recently reached out with both an academic and small business (SBIR) RFP to address some pressing needs in software tools and infrastructure for high-throughput sequence research. The RFP does an excellent job of characterizing the large deficits in toolset:

> *…[Laboratories] face a serious challenge because they do not have access to readily usable software tools or the informatics expertise necessary to take best advantage of the new sequencing capabilities. There is thus a need for robust, well-documented, and well-supported software tools for processing and analyzing the data that individual labs can now generate, and the demand for such software tools will only grow.... While the sequence analysis informatics tools [large centers] have developed are technically available to others, there are significant practical barriers to their use by the wider community. Most of the "in-house" informatics tools developed so far are optimized only for local applications. Furthermore, the software may not be well-documented, and it does not generally have adequate documentation or user support. It may only run on large, local computational clusters (something many researchers do not have), and may not, for example be useful for cloud applications, or operate on multiple platforms. It may require a dedicated group of local bioinformatics experts to maintain or update. In general, the centers have not been supported to develop their tools to make them readily transferrable to other groups, especially groups operating at a smaller scale and/or that do not have extensive dedicated local bioinformatics support. This situation has the potential to create a bottleneck to the ability of the growing number of investigators who wish to analyze sequence data that they have generated in pursuit of their own projects. The issue promises to become even more serious because the costs of producing sequence data continue to drop rapidly, and will soon be exceeded by the costs of analysis.*

Unfortunately, the problem of unusable software and the need for experts to drive them is not new, nor is it unique to the field of high-throughput sequencing. Rather, it is a systemic issue that has plagued the broader field of genetic research for perhaps as long as the field has been producing software. Foundational to this problem is the fact that academia is the birthplace of most new statistical and computational methods in genetic research. Yet, for the most part, these methods are embodied in simple standalone programs that are not supported, maintained, or well documented, and have an unusable or non-existent graphical user interface (GUI).

Further, with a plethora of data formats, it becomes painful to use multiple programs in combination. This results in long learning curves, inefficient workflows, sporadic technology transfer to industry and, ultimately, delayed lifesaving research. Also, the uncertainty scientists have that they are getting accurate results with a new program creates a barrier for them to use new and potentially powerful innovations. Finally, when professors who produce software don't get tenure, when their research interests shift or lose funding, or when students leave, software support ends. This innovation "abandonment," along with the reinvention of the wheel that takes place as innovators unnecessarily re-create software infrastructure (such as data management, graphical components, and statistical libraries) represents a large waste of time (and federal funds).

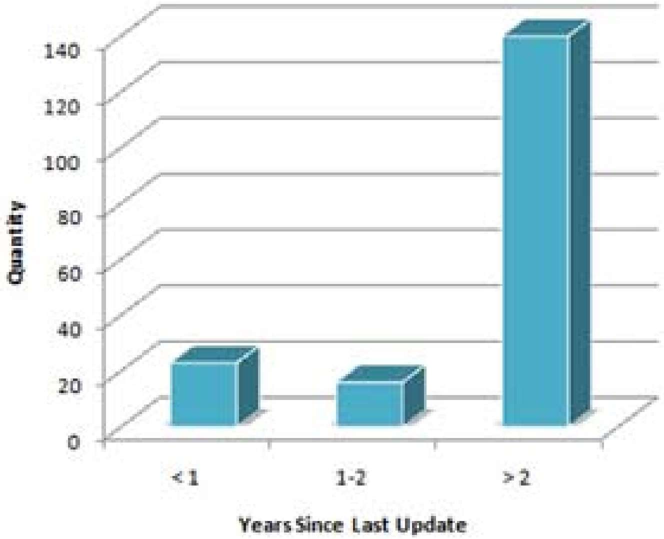
In January of this year, we set out to assess the level of ongoing maintenance and upgrades of academic genetic analysis software as well as their usability. To do this, we performed a careful search of the Laboratory of Statistical Genetics at Rockefeller list of genetic analysis software, containing 564 packages. We examined roughly half of the software packages (those beginning with letters A through L) to establish the last date of update, determined by the most recent public release of the software or documentation as a measure of active maintenance and development. In some cases the links were out of date, and Google was used to find the new package location, if any. The most common and accurate way to find the date of last update was to download the zip or tar file of the latest installation or source files and look at the timestamp of the most recent file in the archive. In some cases we had to look at the date stamp of the PDF file of the manual, in others the authors were thorough and listed dates of release of their software on their website.

Of the 280 programs listed alphabetically from A-L, 23 were designated by Rockefeller as obsolete because they were "unmaintained web-less or merged programs." At the time of our search, we found an additional 45 that had no web link to the software nor could one be found with a Google search, for a total of 68 obsolete programs. The number of programs that had been updated within a year (less than 365 days) of the time of our assessment was only 28. Those 28 contained 5 of the 6 commercial programs listed among the 280. If this is a reasonable sampling of genetic software, only about 9% of academic programs developed for genetic analysis are maintained with some form of update within the last year.

Given that a well-developed user interface is requisite for the adoption and usability of genetics packages by mainstream researchers, we assessed how many of the programs updated within the last year have a menu-based graphical user interface (GUI). Only six of the 23 actively maintained academic genetics programs had a GUI (two additional ones were form-based web page apps). Thus from this sampling, the estimated fraction of academic programs that have the potential to be highly usable and are well-maintained is **approximately 2%** (6 of 274). (This is probably an overestimate, as the programs making the Rockefeller list would over-represent the more well-known efforts and under-represent less well-known programs and those that are abandoned soon after being created. Also, user interface quality is a continuum—some programs that have a GUI are still undocumented or in other ways difficult to use.)

There are a few examples of academic software tools in the field of genetics that have endured for years and have a broad user base. Generally, such efforts tend to be platforms for the integration of methods from multiple authors and in many cases survive without a user interface. One example of this is the Bioconductor project which has enough of a community of contributors who have kept a software framework alive with no single point of failure. However, one needs to be an R programmer to use the framework.

Looking at the rule instead of the exception, why do only 2% of academic software programs have a GUI and get regularly updated? Why is all this innovation and work simply left behind? To understand the structural reasons for the limited academic software toolset, we must understand the role of mindset.



Mindset

Each individual has his/her own mindset, which I'll define as the amalgamation of values and mental models of reality with which a person engages in goal-directed behavior. The values determine the goals, and the mental models determine the means of moving towards those goals. Or to put it another way, either implicitly or explicitly, measurement drives behavior: our actions are directed to closing the measurable gaps between our goals and our present state. It is useful then to characterize a given culture by shared dimensions of mindset as embodied in its measures.

In academia, the prime metric that drives behavior at all levels, from the university as a whole down to the individual professor, is reputation. Yes, academia is about creating new knowledge; yes, it is about educating students; yes, it is about benefit to society; and yes, it is about bringing in the revenues to support all the other efforts. Cash consciousness has certainly been increasing in most universities in recent years, making it a close second, but at the end of the day, reputation is the score-keeping system. Universities recruit students on the strength of institutional reputation, recruit faculty and staff based on reputation, and tenure is granted based on reputation. Grant funding goes to those with a reputation for past achievement, which creates a virtuous cycle of more research, publications, and grants to continue the process of converting cash into reputation. The expectations put on individuals within a system whose prime metric is reputation has been captured aptly in the cliché: "publish or perish." Reputation is a rather intangible quantity, namely what your peers think of you, thus a proxy for reputation is the body of peer-reviewed literature a researcher has produced.

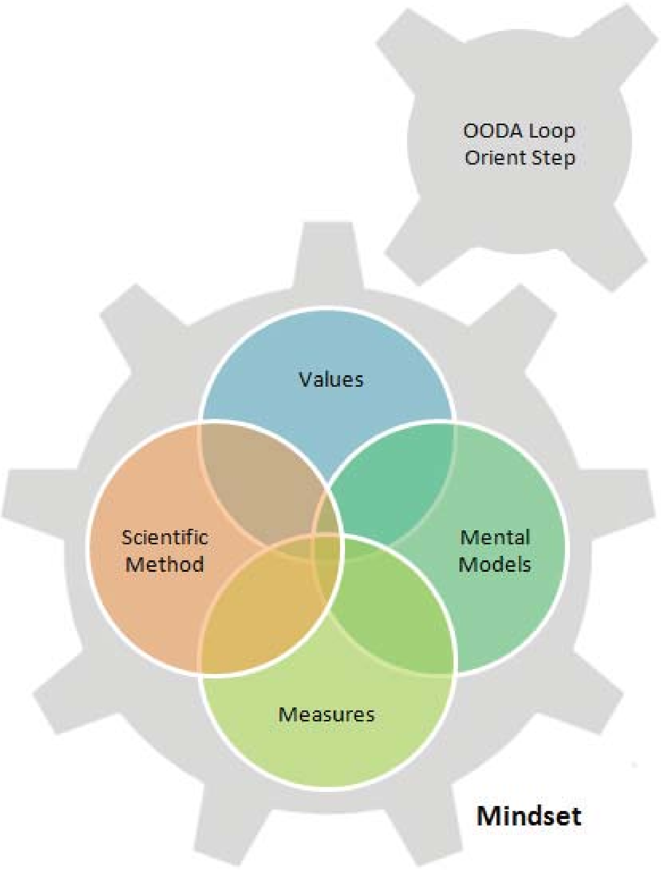
Impact of Mindset on the Toolset

What does this discussion of mindset and measurement mean for productivity? Academic researchers are measured on the quality and quantity of their papers, not on the quality of the toolset (such as the software) they produce. Given scarce time and resources, and given measurement drives behavior, it should be no surprise that only modest effort will be put into software quality by most academic researchers. While the quality of mathematical and algorithm innovation may be high in newly published methods, the power of these approaches does not get transferred into tools usable by the majority of researchers, and so the innovation is unable to be leveraged fully.

In general, a tool should be assessed on the degree to which it reduces the dimensionality of a complex task to simpler steps, increasing both efficiency and effective capability for the user. When a tool can only be used by the toolmaker, its contribution to collective productivity is sharply diminished.

As suggested by our data collection efforts above, the vast majority of academic software tools are likely to become unmaintained, and ultimately be relegated to the graveyard of unsupported, obsolete, and, finally, unused programs. The method developer has incentive to disseminate the software for others to use so that their methods paper will be cited. However, the most reputation is gained by having the methods developer also be the operator of the software, earning co-authorship on subsequent biological research papers. While I do not believe it is intentional, a pattern operates where the scientific promise of the freely downloadable software is touted in the paper, and then the "consumer," due to time pressures and challenges of learning the software, programming/statistical skillset deficits, and/or fear of improperly driving the software and making embarrassing mistakes in publications, must call in the software author and/or his students as consultants on projects to get the research done.

While the cause of academic software quality problems is nothing so Machiavellian as methods developers purposefully making their software hard to use just so they can obtain services work and get their name on publications, it is ironic to what degree that actually may be a productive career move. That is, in a system measured by reputation, it seems to be in the best interests of the author to make their software cryptic so users are dependent on them to co-author their papers. Now, I see no evidence of this being the case, but the main point is that writing user interfaces, creating extensive documentation, and providing tech support does not add a single publishable unit to one's CV, and so is neglected in a system of "publish or perish."

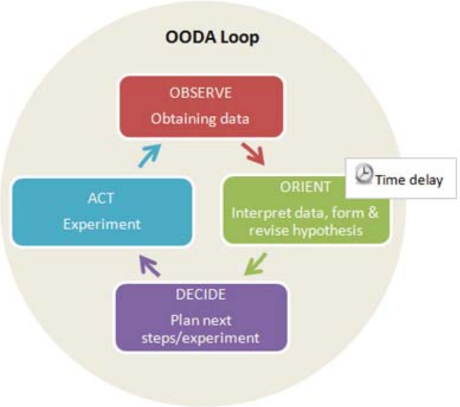


Consequences

In the long view, however, there are negative consequences for everyone. While successful software authors benefit from coauthored journal articles that result from joint work, they also suffer from being inundated with tech support requests from non-collaborators. For this they might get gratitude and citation in an article, but in general, achieve a much lower return in academic credit than other uses of their time. Also, in this system, bioinformaticians have become the constrained or bottleneck resource, and we often hear about the frustration research labs have with their dependence on bioinformaticians to get their data analysis done. Because bioinformatics is a scarce resource, data can sit for months unanalyzed, blocking the next step of a research program.

Perhaps even more frustrating is the impact of outsourcing on one's learning loop. Research, like many goal-directed activities, can be represented well by the OODA loop model where we observe, orient, decide, and act, iteratively moving towards our goals. The velocity of observing the consequences of experiments, re-orienting mental models (mindset), making decisions, and acting on a new experiment sets the pace of research productivity. When an external person is required for us to complete the orient step, it is easy to see how the cycle time of the OODA loop can be delayed by an order of magnitude or more. This, and the associated loss of focus and context that occurs as researchers multi-task on other activities while waiting for external analysis to complete, is devastating to productivity.

All is not rosy for the bioinformaticians either. Despite having job security, they suffer from the stress of being the constrained resource. Further, there is also a balancing feedback loop, whereby the successful bioinformatician has diminishing capacity to develop more novel methods because their project-oriented work for biologists takes up more and more of their time. Thus, we've seen an accelerated effort by bioinformatics cores in the last couple years to empower "end-users" to perform many of the basic analyses themselves in order to reduce the support burden.


OODA Loop
OBSERVE
Obtaining data
ORIENT
Interpret data, form & revise hypothesis
Time delay
DECIDE
Plan next steps/experiment
ACT
Experiment

Reproducible Research

While not immediately obvious, the consequences of the system on reproducible research, especially as it relates to unusable and unsupported academic software, are staggering. Reproducible research has gotten a lot of attention lately as good science rests upon the notion that one researcher can reproduce the approach and methods of another researcher and build upon or find contradictions in the previously generated findings.

Genetic research has become extremely data-driven. Great strides have been made in making data public through various repositories such as dbGaP, the Gene Expression Omnibus (GEO), 1000 Genomes Project web pages, and many public annotation databases. However, less progress has been made on making public the actual steps of data analysis. That is, the fully reconstructable audit trail of the analysis including software (and version), parameters, and the sequence of steps is absent, making the knowledge derived from the data woefully irreproducible.

If inadequate or missing documentation, lack of support on appropriate platforms, buggy or unstable code, and generally low usability are the norm for academic software, how can one reproduce research when the tools are unusable? Worse, as quantified earlier, a large fraction of software tools that are developed and used by a researcher to produce a publication cease to exist within a few years. Reconstruction of the steps of another researcher ranges from tedious to impossible.

The current modus operandi in academia is for analysis to be done with a cobbled together set of command-line tools. Disparate data sources are stored in various places of the investigator's hard drive and reproducing the analysis is problematic. When published analyses represent the juxtaposition of tools of unspecified version, run with unspecified parameters, and with unspecified data cleaning and transformation steps, reproducibility by other researchers is severely compromised.

Over the past few years, awareness has been increasing about the pressing problem of reproducible research. Ioannidis (2005) has described some of the methodological causes leading to false or non-reproducible findings, including small study and effect sizes, experimental design biases, multiple testing, data dredging, conflicts of interest, and selective reporting of results. Though more work has to be done, we at Golden Helix have striven to address many of these issues through education and training, with a particular focus on good experimental design and avoiding uncorrected multiple testing and data dredging.

Keith Baggerly has published on the need to disclose all of the data in publications (Baggerly 2010) and on the need to provide, as part of the publication process, full access to the analytical steps used to construct the analyses (Baggerly & Coombes 2009). Perhaps an instructive example is the high-profile story of Baggerly's team spending thousands of hours of investigation (Baggerly 2010) performing "forensic bioinformatics" on Duke University publications purporting to have constructed a gene expression signature to select therapies for cancer patients. Baggerly writes:

> *The independent reanalysis of these signatures took so long because the information accompanying the associated publications was incomplete. Unfortunately, this is common: for example, a survey of 18 published microarray gene-expression analyses found that the results of only two were exactly reproducible. Inadequate information meant that 10 could not be reproduced.*

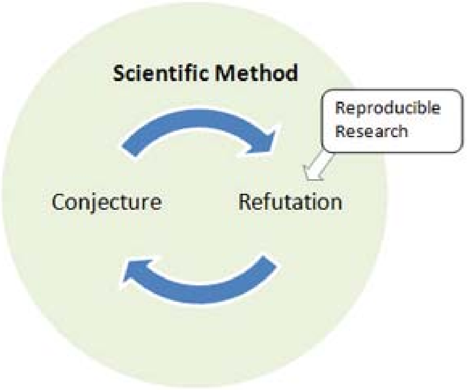
While Baggerly's team struggled to have the flaws in the studies recognized by the institutions that performed them, three clinical trials commenced whereby potentially life-threatening decisions were allegedly being made on patients based on faulty analyses. Also, multimillion dollar venture investments had been made to commercialize this research. The trials are now halted, but there is likely to be costly litigation and finger pointing for years to come, eroding the trust patients have in their doctors, and the trust investors have that peer-reviewed publications can be counted on for commercial decisions.

Keith Baggerly

Baggerly has advocated the use of the Sweave (Leisch 2002) framework to address this problem, and the MD Anderson Cancer Center has mandated its use for their publications, which is a great step forward. Sweave is an overlay to the LaTeX document typesetting system that allows the incorporation of R statistical source code into LaTeX documents so that the analysis performed to construct graphs, tables, and other results for publications is preserved for posterity. Sweave is a good solution if analysis has been done in R, but is only available to programmers using that language for bioinformatics research.

Touching back on the discussion of mindset, consider that reputation as the prime metric of academia, in some ways, is actually inimical to reproducible research. Reputation takes years to build and yet can be destroyed almost overnight by one public faux pas. Unfortunately, despite our aspiration to consider mistakes as learning experiences, when they become public in a system of competition for reputation, those very mistakes can be devastating. While reproducible research is in the best interest of scientific progress of the field as a whole, for individual researchers to expose their work to criticism and falsification exposes an author to the threat of lost reputation.

Nevertheless, reproducible research is essential to scientific productivity, being a cornerstone of the scientific method. Application of the scientific method has led to the incredible advances in technology and quality of life in the last century and more. The scientific method's process of conjectures and refutations is like a binary search into the nature of reality, leading to exponential productivity and generation of knowledge. The demarcation between science and non-science rests on falsification, and the scientific method is all about replacing false or incomplete hypotheses and theories about the nature of reality and converging to truer ones of higher explanatory and predictive power. That is, we form falsifiable hypotheses, do our level best as good scientists to refute them ourselves, and then, if unsuccessful, present them to a broader community for potential refutation. If a hypothesis stands up to all of this scrutiny, others may build upon it, integrating corroborated hypotheses into bodies of theories that provide powerful causal models as we move towards our goals. If reproducibility is inhibited, then refutation by our peers is inhibited. Without this selection process, a body of science becomes a muddled mess and stagnates. If we obstruct refutation, we inhibit the exponential engine of productivity that the scientific method provides.



Thoughts on a Solution

If productivity in our field is measured not only by volume of publications, but also by the quality of the causal theoretical models for biological processes, we have a number of systemic and interrelated obstacles to productivity in our field:

1. Bioinformatics has become the constrained resource limiting the pace of genetic research—there is a skillset deficit in the field as a whole.
2. The software toolset for genetic research, produced and broadly used in academia, has serious shortcomings for productivity. For the most part, it can only be operated well by the constrained resource.
3. The mindset embodied in reputation as the prime metric of academia reinforces the toolset deficit.
4. The toolset and mindset inhibits the reproducibility of research, a cornerstone to the scientific method and the productivity that method provides us.

Of course Golden Helix sells commercial software for genetic analysis, and it would be convenient if we could claim just buying our software would solve all these problems. It won't. Nevertheless the content of this article informs our own product and services innovation process, and we believe we have some key components in the direction of a solution.

Good software helps researchers overcome limits of their skillset in statistics and computer science and provides a visual processing capability to enable biologists to make sense of their data. Further, even for experts, it collapses complex multi-step processes into simpler ones, amplifying the level of abstraction and minimizing the drudgery of working down "in the weeds" with the data. In fact, some of our biggest fans are bioinformaticians who have spent years with command-line tools and see our user interface as liberating them from the heavy lifting of mundane data transformation tasks and keeping track of their work.

Nevertheless, there are legitimate fears we've heard voiced by some bioinformaticians; specifically, that it would be irresponsible to provide user-friendly bioinformatics software to the uninitiated. For certain kinds of analysis even easy-to-use software workflows are complex enough that mistakes can be made. For instance, associations between biomarker and phenotype may be found that can be explained away by confounders as a result of problems in experimental design or other sources of variability that even often experts miss. Thus, there is a significant risk that a novice will jump to report an exciting finding based on an incomplete analysis.

Publish or perish

The conflict can be formulated as whether or not to encourage a biologist to analyze his or her own data. The reason not to is to avoid embarrassing mistakes. However, almost every bioinformatician started off lacking skills in either statistics, computer science, or biology and had to learn a domain-appropriate subset of the rest generally through experience and, perhaps, being paired with a capable mentor.

Clearly it is not viable or sustainable to divorce the biologist from their data. How, then, can we ease the transition, allowing the novice to learn by doing without making egregious errors in analysis? One solution is an audit function where non-bioinformaticians analyze their own data, possibly more slowly, and making more mistakes, but then have their work audited by peers and, ultimately, a bioinformatician before publication. In the long run, biologists will become more self-sufficient on more and more complex analyses and enhance the productivity of the whole field. (Additional education at the undergrad and graduate level would help as well.) Meanwhile, bioinformaticians will be freed up to work on higher level problems that engage their passion and skillset, stepping in to help the biologist only on truly challenging problems. And there will be no shortage of work anytime soon.

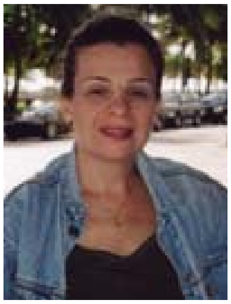
Still, with the aforementioned academic software toolset and associated reproducible research problem, it is difficult for a bioinformatician to audit even another bioinformatician's work. This could be made possible with software tools that automatically log all analysis steps and parameters and saves the intermediate steps of analysis, including graphical output, for inspection at a later date. Further, the ability to annotate one's work as one might do in a laboratory notebook would allow an analyst (expert or novice) to share his or her work with a colleague for collaboration, review, and audit. Additionally, it would be tremendously beneficial if a researcher could deposit an entire analysis project that documents the methods used for a publication along with the data in public repositories such as dbGaP. (Last year, Golden Helix did just this, as we assisted in depositing an Alzheimer's GWAS study to dbGaP, including not only the raw data, but a full audit trail of quality control and association analysis.)

In fact, with bioinformaticians in short supply, Golden Helix has provided more and more analytical services for our customers over the past several years, including auditing data analysis results. We've fine-tuned our services process for our clients at various levels of engagement and found that having shared analysis artifacts and interactive web meetings makes for a rich collaboration and learning environment for both parties. We've learned through dozens of engagements that collaboratively defining a clear high level statement of work, getting a complete kit of all necessary data before starting a project, and adhering to a frequent meeting schedule with highly visual artifacts leads to highly engaging and educational collaborations.

What about the methods developers? Should we advocate they spend more time writing GUIs, providing extensive documentation, and performing tech support for no academic credit? In the current measurement system (which we do not propose to change—not yet, at least), it would be at the expense of one's academic career to do this. Further, as has been noted in a recent Nature Editorial (Merali 2010), software engineering and developing GUIs is not a core competency in most scientific fields. Traditionally, one of the weaknesses of commercial offerings for genetic analysis is that they lag behind academic innovation. Unfortunately, creating a robust and highly usable product and supporting it takes away resources from method innovation (which is why academia focuses on the latter at the expense of the former), making companies generally less innovative than academia in terms of pure statistical and genetics methodology. This is especially true at the beginning of a new field, but the gap closes as a field matures and analysis methods standardize. We believe the marriage of academia and industry where each stands to gain on their respective prime metrics (reputation and revenues) is an excellent working model.

Some of the richest collaborations Golden Helix has had as a company have been integrating algorithms and command-line tools of leading academic innovators into our product. Notable examples include the integration of the PBAT family-based association testing package from Christoph Lange's group at Harvard,

Christoph Lange



Suzanne Leal

an ongoing collaboration since 2006, and our recent integration of Suzanne Leal's CMC (Li and Leal 2008) and KBAC (Liu and Leal 2010) collapsing methods for the analysis of rare and common variants. In such collaborations, we bring our experience in software engineering, human interface development, and high performance computing, and synergize them with the incredible innovations in statistical genetics that come from these researchers. The idea is for each party to do what it does best and be rewarded in the goal units of our respective systems. Such collaborations open up the opportunity for large scale educational events that promote these methods and their use so that the author gets credit and prominence. Further, the author is freed up from technical support, the community is educated on how to use the methods most effectively, and the software is maintained and sustained, contributing to scientific productivity for scientists around the world.

When you consider that reputation is the lifeblood of academia, a collaboration that engages the marketing resources of a company that actively reaches people in this niche market is a benefit in and of itself. Add to that the possibility of giving an innovation that has a 90+% chance of obscurity the opportunity to impact the research productivity of hundreds, while at the same time generating reputation and grist for the grant mill—it seems like a win-win for everyone.

Conclusion

Vast opportunities for the improvement of research productivity exist at the leverage points of skillset, toolset, and mindset. By providing tools that collapse complex multidimensional problems into highly usable software components, we reduce the bioinformatics skillset needed for researchers to get their work done, accelerating by an order of magnitude or more the OODA loop of the research process by allowing biologists to analyze their own data with safeguards, and addressing the systemic bioinformatics constraint in our field. We propose maximizing the utilization of scarce bioinformatics expertise, empowering bioinformaticians to work at even greater heights of complexity and facilitating their toolsets to be leveraged by mainstream researchers through commercial partnerships and/or infrastructure. Further, by enhancing reproducible research, we remove obstacles to application of the exponentially productive scientific method.

Whether you are a genetic researcher or a doctor on the starship Enterprise, the potential for orders of magnitude more productivity is within your grasp—the only question is whether the mindset, the mental models of reality and value systems we hold, will allow us to embrace the necessary changes. It is my hope that this whitepaper has transformed your mindset in some small way towards considering the leverage points for enhanced productivity in your own research processes—just as writing it has transformed mine. …*And that's my two SNPs.*

References

Baggerly, K and Coombes, K. (2009) "Deriving Chemosensitivity from Cell Lines: Forensic Bioinformatics and Reproducible Research in High-Throughput Biology" Annals of Applied Statistics 3(4) pp. 1309-34.

Baggerly, K. (2010) "Disclose all data in publications." Nature 46 p. 401.

Gentleman, R, et al. (2004) "Bioconductor: open software development for computational biology and bioinformatics" Genome Biol. 5(10):R80. Epub 2004 Sep 15.

Ioannidis JPA (2005) "Why Most Published Research Findings Are False" PLoS Med 2(8): e124. doi:10.1371/journal.pmed.0020124

Krishnan, M and Wilson, A. (2010) "A characterization of the training needs in genetic epidemiology". International Genetic Epidemiology Society. Boston, MA. 11 Oct. 2010. Conference Presentation.

Lambert, C. (2010) "Stop Ignoring Experimental Design (or my head will explode)" Our 2 SNPs. Golden Helix Inc, 29 September 2010.

Leisch, F. (2002) "Sweave: Dynamic generation of statistical reports using literate data analysis" Compstat 2002 Proceedings in Computational Statistics 69 pp. 575-580.

Li B, Leal S (2008). 'Methods for Detecting Associations with Rare Variants for Common Diseases: Application to Analysis of Sequence Data' Am J Hum Genet 83:311–321.

Liu DJ, Leal SM (2010) "A Novel Adaptive Method for the Analysis of Next-Generation Sequencing Data to Detect Complex Trait Associations with Rare Variants Due to Gene Main Effects and Interactions." PLoS Genet 6(10): e1001156. doi:10.1371/journal.pgen.1001156

Merali Z. (2010) "Computational science: ...Error." Nature Oct 14;467(7317):775-7.

About Christophe Lambert

Dr. Christophe Lambert is the President and CEO of Golden Helix, Inc., a bioinformatics software and services company he founded in Bozeman, MT, USA in 1998. Dr. Lambert graduated with his Bachelors in Computer Science from Montana State University in 1992 and received his Ph.D. in Computer Science from Duke University in 1997. He has performed interdisciplinary research in the life sciences for over twenty years.

Dr. Lambert is currently the co-chair of the Food and Drug Administration's Genome-Wide Copy Number Variation Data Analysis Team of the Microarray Quality Control Consortium and is also an active participant on the consortium's gene expression and SNP diagnostic teams.

Considered a thought-leader in genetic and predictive research, he has given numerous talks, trainings, and presentations on best practices and leading methods based on his experience in conducting dozens of whole-genome studies.

About Golden Helix

We are inspired by significance. Not only statistical, but technological, scientific, and personal significance. It's embodied in everything we and our customers do. And we believe the only way to achieve significance is by transcending the status quo. Every day we strive for extraordinary analytic and technological advancements that empower scientists around the world to pursue that which is truly significant: from uncovering the genetic causes of disease and transforming drug discovery to developing genetic diagnostics and advancing the quest for personalized medicine. To learn more about Golden Helix, visit our website at www.goldenhelix.com or our blog at blog.goldenhelix.com.