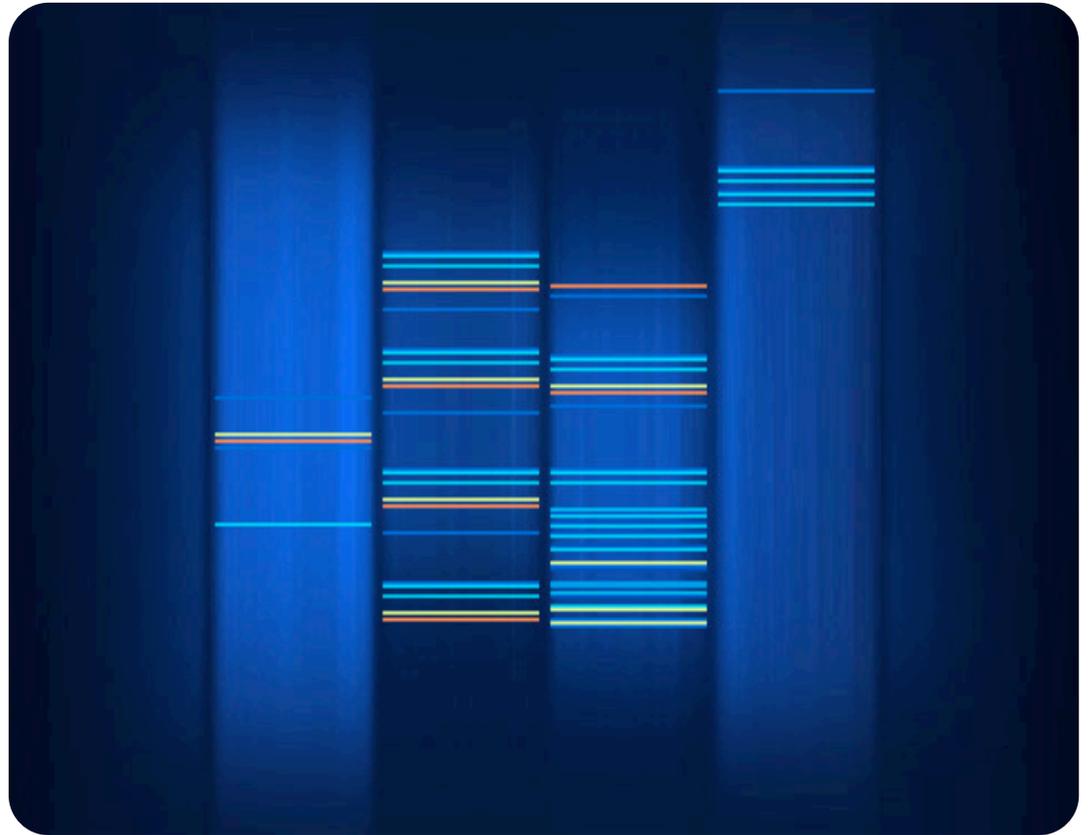




bioinformatics blog journal **bio::blogs**

```
use Bio::Blogs;  
$blogID_obj = Bio::BlogID->new(-file => "blog.fasta", -format => "wp" );  
$blog_obj = $blogID_obj->next_blog;
```

Tips & Tricks



thanks

This document is made up entirely of content generated by scientific bloggers, active researchers who are happy to donate their ideas and time freely to a continuous, online dialogue about science, biology and bioinformatics.

Each article is written by someone who is a bioinformatician, but also has a website where they occasionally, or more than occasionally, post their thoughts and observations.

So if you found any this useful go and take a look at their blogs, there's plenty more to be found. Even better, create your own blog on wordpress or blogger and join the discussion.

michael barton

pedro.beltrao

Importance of the literature

Try to keep up to date on the literature. It is very important to know what questions can be tackled with currently available information.

Reading other people's work is, for me at least, the best way to get a sense of where the boundaries of current knowledge are and what are the most interesting questions we can ask. In particular for bioinformatics, it is important to pay attention to new data sets as they are being published since, each new data set gives us more possibilities to explore a question.

To keep up I have RSS feeds to automatic Pubmed queries that I am interested in, RSS feeds to interesting tags in Connotea and Citeulike and the usual RSS feeds or emails with content alerts for journals that interest me.

dan.swan

Interactions

When, as bioinformaticians, we think about interactions, it's probably related to an organisms interactome or what proteins do when they get up close and personal. However the social interactions that we have with our colleagues are what really defines the usefulness of the discipline.

Bioinformatics is a great uniter, bringing together maths, computer science, biology and even philosophy. We are long term 'interdisciplinarians' in an age where interdisciplinary science is still almost a buzzword.

Unavoidably, a bioinformatician talks to biologists. Whatever your background it is the quality of these interactions that are the basis for success in your chosen career.

How can you make the most of these interactions?

Know your technology

This might sound obvious, after all this is your speciality right? However there is always the temptation to recommend the latest and greatest to your collaborators. Ask yourself some questions first. Have you actually used it yet? Does it deliver what you expect? Are you chasing technology solutions for technologies sake? Can you explain it clearly to the biologists and outline why you have made your choice?

Know your biology

Even more fundamental one might assume. This is harder if you're dealing with people from all kinds of lab backgrounds. One day you're talking to a microbiologist, the next you're talking to a clinician. It is essential that you have some appreciation of their branch of science. Reviews are your friends, but requesting relevant papers from the biologists in advance will save you hours of grubbing around in Medline. A cursory glance at the labs website for recent publications will ensure that you can hit the ground running. You're never going to acquire the knowledge they have in a short space of time, understand the limitations of this, and be prepared to leverage their expertise.

Do they know your background

This is incredibly important. To the biologist, you're a bioinformatician. All too frequently they will make an assumption about your background. You might have a biology background yourself, but they will assume you're a computer scientist and talk to you like they were lecturing a high school student. Or you are a computer scientist, and the biologist will assume that you're intimately familiar with the regulation of pathway X in organism Y. Make your experience in the field clear, this helps manage expectations and avoid false starts.

Manage expectations

A lot of biologists don't appreciate the back-end complexity of bioinformatics work. The applications they're used to are simply black-box solutions to them. Often they think that bioinformaticians have a big red button hidden under their desks called 'Analyse' that will return the data they want in the format they want. Explain clearly the limits of what bioinformatics can achieve and how much work it actually is going to be for you to do what they're asking. Let them know when their data is not suitable for rigorous analysis and why. Suggest solutions to these kinds of issues for their future work. This also maintains the position of you having intellectual input into the project, and the benefits that will bring.

Maintain communication

All too often interactions that start promisingly can dwindle to inactivity as both you and your biologists get sidetracked by other things. If you're serious about the work you do, it is up to you to keep the interaction going. Personal organisation is the key here. Tracking your projects effectively, and knowing when to gently prod the biologists for the next round of interactions is all important. Don't rely on them to come to you, go to them.

euan.adie

Revision control

Take the time to learn subversion and then use it for your code and (equally importantly) your data sets.

Creating applications

If you're writing a web app or database, assume that you'll no longer be around to support it next year (you'll be too busy, or you'll have moved job, or changed email addresses). Make plans to cover this eventuality while you've still got access to everything.

Blogging

I got more collaboration requests through my blog than I ever did through talks or conferences.

bertalan .meskó

Genetic disorder databases

Diagnostic delay poses a problem for rare diseases. We can't expect physicians to know everything about all the cc. 4000 genetic conditions. But we can help them how to find relevant information and quickly understandable material on genetic conditions.

- Online Mendelian Inheritance in Man
- Single Gene Disorders and Disability at Centers for Disease Control and Prevention
- Specific Genetic disorders at National Human Genome Research Institute
- Genetic disorders at Wikipedia
- GeneReviews
- Ask the Geneticist
- National Organization for Rare Disorders
- Center for Inherited Disease Research
- Open Directory Project: Genetic Disor-

pierre.lindenbaum

The social life

- blog your work, your company, the papers of interest, use it as a notebook.
- register your blog on technorati
- register your blog on postgenomic
- share your slides on slideshare
- share your bookmarks/papers on del.icio.us and connotea
- share your relations and register one or more social networks - LinkedIn, Nature Network or Xing
- share your knowledge - Wikipedia, Freebase and Openwetware

The bibliographic life

- use a feed aggregator to read your blog, journal, and company feeds.
- use the RSS services from the NCBI to survey your subjects of interest.

The lab life

- write a RSS page for your lab - new papers, new openings
- open a public calendar for your lab to notice the next meetings, e.g. calendar.google.com
- publish the FOAF description of your team on your web site
- publish the DOAP description of your projects on your web site
- share your data - manyeyes

The bioinformatics life

- learn a low level programming language - C, java, ...
- learn a database language - mysql...
- learn a scripting language - python, perl...

- be an expert, as much as it is possible, in these languages.
- look at others' code.
- share/save your code - sourceforge, code.google.com
- do not create another new format: use xml and learn xslt
- do not create another new XML schema: use RDF
- publish your web pages using strict xhtml and/or RDFa
- embed your graphics in your web pages with SVG.
- do not create another identifier for your biological entities: use LSID.
- some user defined C functions can be implemented in mysql

The human life

- go outside: the real life is here

roland.krause

Biological data sets

Visualize the data early in the process in every which way. Make yourself familiar with the way the data is structured before trying to "mine" it. If you start drawing histograms or scatter plots only when a publication draws close you are likely to miss important aspects.

Inspect the data manually in depth. Pick a few examples that you follow through completely.

Searching biological relevance, a common task for a bioinformatician, is something that you cannot write a SQL-statement for. Finding the right question is essential and be prepared to do this tough part yourself - delegating it back to the bench-side is easy but not an option for a fruitful. Typically, the bench-side is more knowledgeable in the matters regarding the data generation process, so ask them questions but don't expect them to pose a phrase you can translate easily.

ryan.castillo

Working in bioinformatics, I've found that there is more of a need for clearer communication between collaborators and less room for personal pride. I emphasize the throwing out of pride because Bioinformatics is a field that is an integration of biology, computer science and statistics. This results in very intelligent people, who are masters in their respectable field, trying to collaborate with people of other scientific paradigms. Thus you have numerous papers coming out where researchers do not fully understand the work they are publishing.

An example of this is the constant use of average linkage hierarchical clustering in gene expression analysis. While this is an efficient clustering practice there are numerous methods out there that are as good or better that have little light shone upon them due to the lack of understanding of the analysis.

Coming into Bioinformatics people need to understand that they will never master all the aspects of the field. What can be done is to understand general concepts and master your own respective skill set then use effective communication skills to come up with the best collaborations.

paras .chopra

Academic Summers

If you don't have much experience in research as of now, browse journals on your field of interest, pick up problems which you identify to be tractable by you and then just start cracking the problem. Aim for a publication during these summers. Only that way you will be able to do a good project.

Then next year, you can expect to get a summer position at a good lab, just aim at publications for now.

neil .saunders

Code from articles

Neil wrote a great post on recreating the code used in publications, however there's not enough space to fit here, but the article can be found at tinyurl.com/2vmbck

konrad.förstner

Set up a tool box containing

- a programming language (or several) that suits your needs and lets you write functional and readable code quickly, e.g. Python
- the knowledge of the most important/used Unix commands
- a powerful editor - syntax highlighting and completion
- a reference management system like Connotea/citeulike or JabRef.
- a browser equipped with handy extensions
- a proper version control system for your code and documents - CVS or Subversion might be the options that come to your mind first, but if you would ask Linus Torvalds, he would recommend git

Experiment documentation

- Think about your future ego that will thank you for the possibility to travel back in time.
- Write a lab journal/book
- I highly recommend a blog or wiki based system, but for sure flat files or even a hand written notebook (unlikely) might do the job.
- Write down what you did, how you did it, why you did it, and where the input and output are stored. Also some of your key graphics, article summaries, and overviews of the results should be placed in there.
- By articulating the thoughts in your head you define aims and become aware of them - "Begin with the end in mind" as Stephen R. Covey says.

Code Documentation

- A short description at the beginning of the program is the minimal documentation.
- Give meaningful names to functions and variables. If you cannot find a proper name, you don't know what the thing really does.

File documentation

- Use clear file names, file headers and folder structures to find things easily
- I do this by putting the date in the beginning of folders where I organise my analyses e.g. 2007-05-28-foo_analysis_in_bar_data.
- I put a master shell script into these folders which contains all the calls and sometimes, as a comment, the names of the resulting files.
- The output files of my programs have a short header that tells by which program they were created, at which time and with which parameters the program was called (I use a little library for that).

Empathy and compassion

- Think about which information in your head could be needed by other people in your lab or in your scientific community. A wiki for example can be an excellent tool for organizing generally needed knowledge and avoiding the drain of it if somebody leaves the lab.

Other small tips:

- Be aware that demands will change so program from the beginning on in a way that make it easy to extend/adapt the program.
- As soon as there is the smallest possibility that a function might be used twice write a proper library.
- Often flat files are a handy format for data, and grep and others are good companions. If data sets are constantly changing and built of many components a proper database might be a better solution. For many purposes you don't need a client-server-based database like MySQL but a simple library based one e.g. SQLite that does the job without server set up and user management.
- Code your graphics, as this will give you more control. Especially if you have graphics that consist of different panels, it will save you a lot of time if changes will be needed (and this will be the case). I personally create primarily SVG files and convert them to other formats if needed.
- If you set up a web-server think not only about functionality but also about security. Make the internet a safer place and don't feed the spam/botnet industry with your server.
- Regarding sustainability: We are not evolved to work for hours in the same position in front of a computer. Know when to take breaks, stretch, and try to focus as often as possible at distant objects. Hopefully this will help to maintain proper eyesight long enough until neuronal interfaces are on the market.

michael.barton

Databases

Use a database to store all your data, something along the lines of mysql. The advantage is that you are always using the same method to access the data - which is not the case if it's spread across flat files.

Secondly, when you start to analyse larger and larger data sets, use indexes and the EXPLAIN command to keep query times low. Data queries that take more than a few minutes to run will really slow down how quickly you can test and run the required analysis.

R

Generally if you have quantitative data you're going to need some kind of test to prove and answer a question. So I recommend getting a decent book and learn the programming language R, as well as basic statistics. This should include classical one, two, multiple sample tests. Tests where the variables are nominal or continuous. Learn the various types of statistical modelling, as well as the assumptions involved. Learn to use the lattice and grid packages to communicate your results simply and effectively.

Distractionless breaks

When you work all day on a computer with internet access, there a plenty of opportunities for distraction. The best way I've found to avoid these distractions is to focus on working for fixed periods without checking email or surfing the net. Forty minutes works about best for me. After this period I check my email and feeds, get some water, have a stretch, then get back to working. I've that by doing this you can really power through, and get enough stuff done to leave a long lasting sense of satisfaction.