Review Article

Network-centric computing in genomics

Brian Fristensky
Department of Plant Science, University of Manitoba,
Winnipeg, MB R3T 2N2 Canada.
frist@cc.umanitoba.ca

Abstract

Biology is becoming increasingly dependent on information resources. Large-scale projects generate enormous amounts of mapping and sequencing data. Complex analytical methods are required to work with these data. Lab groups create databases which encode combined expertise of lab members into object-oriented knowledge bases. The desktop-centric personal computer is an impediment, rather than an asset, for most of these tasks because it is oriented towards a single user on a single computer, rather than multiple users sharing resources. BIRCH is a network-centric resource providing a comprehensive set of databases and software for molecular biologists. BIRCH is integrated into a larger campus-wide Unix system so that all computing, including molecular biology, Internet access and office tasks can be done in an easy-to-use graphic desktop, accessible anywhere on campus, at home, or when travelling. The network-centric approach makes all resources accessible in a consistent fashion, minimizing costs and software administration, and promoting reliability, data integrity, security and scalability. See http://home.cc.umanitoba.ca/~psgendb.

Desktop computing: what we've gained and what we've lost

Starting up a new lab is an opportunity to make a fresh, start, to do things right the first time. This idea was at the front of my mind when I set up my lab at the University of Manitoba in 1990. The late 1980s was the period when the PC became a necessity for most biologists. Resources like GenBank were only available if enough members of a department had the will and the money to keep the databases on a minicomputer or mainframe. Among biologists, only hard-core computer aficionados like myself knew about the Internet, and almost no one appreciated why email was useful. Communication between PCs was by modem. The desktop PC, thorough its relative ease of use and moderate cost enabled biologists to take advantage of computers for data analysis, writing, and making figures. The small group of us who had learned computing on mainframes appreciated the "user-friendliness" of PCs. However, we also noted that PCs lacked the discipline, consistency and reliability of mainframes. Of fundamental importance was the fact that mainframes and minis allowed any user to do any task from any terminal. Much like the Protestant Reformation, in which every believer became a theologian, every PC user became a system administrator. And, as in the Reformation, numerous denominations sprang up, peppering labs with PCs, Macs, some UnixPCs, Amigas, which still had to talk to the Vax and Unix minis or workstations. People learned to take for granted that a program that would run on one PC could not run on another. Data, documents, and drawings became scattered across these machines like samples in a -70ºC freezer. Files accumulated with no one in charge, most older files being of little value, but at the same time not easily distinguishable from those few that were of great value, due to inconsistent or incomplete labelling. Since most biologists had never used mainframes, things such as having to reboot several times a week were just a normal part of using a PC.

The demands of 21st century biotechnology

In recent years, the isolated lab with a group leader and a few students has given way to megaprojects with many collaborators sharing data, clones, cultures and strains. At the same time, the data from publicly-funded projects is increasingly used by others in the research community: complete genomic sequences, genetic maps, and expression data. Researchers need access to complex methods, such as gene prediction, phylogenetic analysis and recognition of patterns among groups of sequences or expression datasets.

Because each PC is a unique combination of hardware, software, configurations and data, it is actually an impediment to progress. In particular, fragmentation of data across many PCs allows for inconsistent organization of data and makes it difficult for data to be shared, or even located. Differences in software and configuration from machine to machine results in competition among lab workers for use of specific computers. When expensive software is installed on a single machine, workers can only use it on that machine. When many users use a given machine, changes or deletions made by each user affect all other users. Like the -70ºC freezer, no one is responsible for keeping the PC organized. In summary, the standalone PC makes what you can do strongly dependent on which machine you're using.

Keeping it simple

Though the term "system" is often used indiscriminately, it implies an organized set of components that work together. My strategy was to build a laboratory computer system meeting the following requirements: (a) tasks included a broad spectrum of DNA sequence analysis methods, general office tasks, Internet access, and a laboratory database. (b) These tasks should all be available on a single operating system, so that we didn't have to support more than one computing platform. (c) Software should be centrally-administered to ensure that everything works for everybody. (d) Any worker should be able to do any task from anywhere. No task or piece of data should reside on a standalone computer.

Although today's Web-based tools bring some of these capabilities to the PC desktop, the Web interface is relatively inflexible compared to user interfaces in typical application software. As well, lack of integration into the user's local filesystem makes web-based tools more awkward to use, compared to locally-running applications.

Paradoxically, as our research demands more sophisticated tools, the need for a simple, clean solution becomes imperative.

How it all works

Network computing is best summarized by the expression, "the network IS the computer", meaning that resources such as CPUs, disk drives, printers, and mail are handled by one or more servers connected through a network backbone (Fristensky, 1999a; http://home.cc.umanitoba.ca/~psgendb/nc/_). Many of these components are redundant, providing greater reliability. Services could be provided by different hardware (e.g. Sparc, Intel, Alpha, RS6000) running different operating systems (e.g. Unix, VMS, OS400). As illustrated in Figure 1, four users are logged into four identical X-terminals, which display windows generated by applications, using X11 protocols (Open Group, 1998). The applications themselves run on any number of identically-configured servers which do all the actual processing other than drawing windows on the screen. The users' home directories, as well as publicly-available files and software, reside on a central file server, mounted across the network to all servers. Thus, any user can log into any server, from any terminal, and perform any task. X-windows emulators for PCs allow users to work at home, with exactly the same desktop that appears on their X-terminals at work. Because all configuration occurs at the server, X-terminals require no configuration or upgrades. Failure of any one server or terminal is not a problem because servers and terminals are interchangeable. Automated backups are done for all users. While the file server is, in principle, a single point of failure, the fact that it serves a large number of users makes it cost-effective to invest in highly reliable file servers, such as RAID servers. (To put this point in perspective, the hard drive in your PC, probably the cheapest available, is also a single point of failure.) Performance for multiple users can be as good or better than performance on a standard PC by making sure that network bandwidth and the server to user ratio is adequate.

Universities and corporations typically already have clustered servers. For example, at the University of Manitoba, the Biological Research Computing Hierarchy (BIRCH) system was built to take advantage of an already-existing Unix server cluster (Fristensky, 1999b). In Figure 1, publicly-readable directories contain software (bin), data (dat), documentation (doc), and administrative files (admin), in addition to complete DNA (GenBank) and protein (PIR) databases. The only setup done by the user is to run a script called 'newuser', which configure their account to read a central BIRCH configuration file each time they login. Since administration is centralized, changes and additions are immediately available to all users. In 9 years of administering BIRCH, it has never been necessary to make separate configuration changes on each user's account.

The original motivation behind BIRCH was to provide a campus-wide user base with a diverse set of computing resources that could grow over time. BIRCH has been created entirely using free software from numerous labs. Although there are too many programs in BIRCH to list here (currently over 300), categories include:

sequence format interconversion
sequence printing and display
restriction site search and mapping
RNA structure
protein structure
primers and oligonucleotides
sequence similarity
multiple sequence alignment
sequence database manipulation and search
laboratory database management
pattern recognition and matching
phylogeny
gel imaging
genetic mapping
gene expression analysis (under development)

Rather than being locked into a single program or package from a single vendor, BIRCH has taken advantage of the Genetic Data Environment (**gde**) (Smith *et al*., 1994). (Note: A commercial version of **gde** is included with GCG package. See www.gcg.com). While **gde** by itself is primarily a multiple alignment editor, it has the unique ability to launch other programs. The list of programs launched is read from a centrally-administered file called .GDE-menus. To add a program to **gde**, all that is required is the addition of a short specification of parameters to be set by the user, which are used to build a Unix command that launches the application. **gde** creates menus that let the user set these parameters.

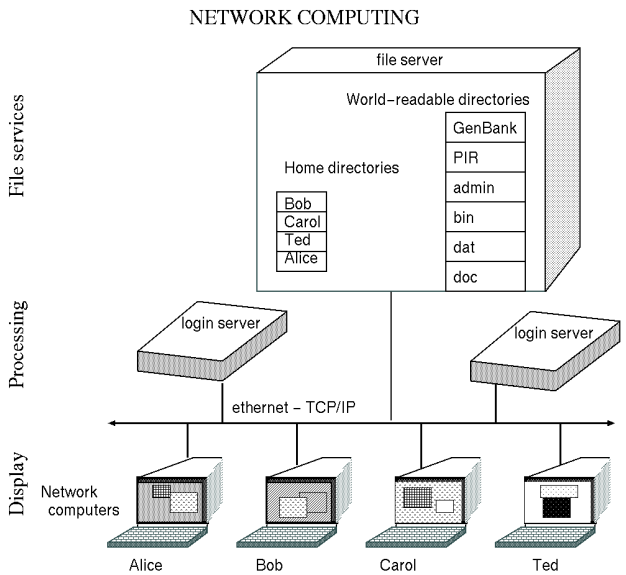

Figure 1. System schematic.

Sequence analysis often involves many steps and a number of programs, often written by several different authors. **gde** is unique by making it easy to chain programs together seamlessly. Because **gde** takes care of conversion of file formats and the actual running of programs, the user does not have to learn the intricacies of running each component program. Equally important, as new methods become available, they can be plugged into the existing .GDEmenus file, expanding the choices available to the user.

Complete information on BIRCH, including documentation for all programs, as well as information on how to create your own BIRCH site, can be found at

http://home.cc.umanitoba.ca/~psgendb.

Example: Phylogenetic analysis

Figure 2 illustrates how BIRCH simplifies the complex analytical tasks requiring many different programs used in concert. The **gde** window at top of Figure 2 contains a set of plant chitinase III genes, whose protein coding sequences are extracted by **features** (Fristensky, 1993) into a new **gde** window (e.g. all sequences in this window begin with an 'atg' start codon). Sequences are translated by **ribosome** (Fristensky, 1993) (3rd window from top) and protein sequences aligned using **pima** (Smith and Smith, 1990) (4th window). The protein alignment is used to guide a DNA alignment of the coding regions using **mrtrans** (Pearson, W.R., unpublished). (Note the correspondence between gaps in the protein alignment with those in the DNA alignment in the 5th window.). Finally, **fastDNAml** (Olsen, *et al*., 1994) is run to produce a maximum likelihood tree, with a report appearing in a text editor and the tree appearing in the **treetool** tree editor (Maciukenas, 1994). By storing the results of each step in a separate **gde** window, the intermediate results can be independently saved or analyzed, and the user can choose which program and parameters to use at each step, as indicated in Figure 3.

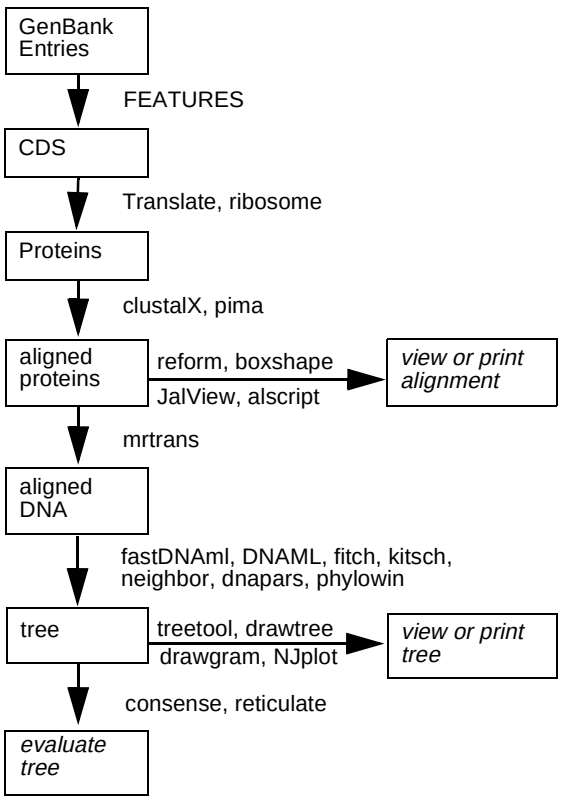

Figure 2. Screen-shot of semi-automated phylogeny using GDE to launch programs at each step.

Figure 3. Schematic of program choices at each step in phylogenetic analysis.

Any sequence analysis package is limited in functionality to what was included by the vendor. The BIRCH approach is to integrate new programs from many sources to provide users with choices at each step of the analysis. In the analysis shown in Figure 3, there is a choice of several alternative programs at most of the steps. For example, construction of the tree from the DNA alignment could have been done by any of 7 programs implementing parsimony, maximum likelihood or distance methods. Choosing **phylo_win** (Galtier *et al*., 1996) at this point would transfer the analysis to **phylo_win**, which implements several of these methods as well as providing a separate tree viewer.

Since the overhead involved in installing a program can be substantial, the network-centric model means that the job of installing each of the component programs used in this analysis need only be done once to provide these capabilities to everyone in the lab. In the PC model, each lab typically installs a different set of programs, scattered across several PCs. Not every program is available on all PCs, and each PC must be upgraded and debugged separately.

Example: Laboratory databases

Creation of a database for an EST project is illustrated in Figure 4 with **AceDB**, now the most commonly used program for genome databases (Durbin and Thierry-Meig, 1991). At top left, cDNA MB56-1G is selected in a **gde** window, and a **fasty** (Pearson *et al*., 1997) search compares it with the GenPept protein database. The output for MB56-1G (Fristensky *et al*., 1999) is at centre left. The main **ACeDB** window appears at top right, and the data for MB56-1G are entered in the window at lower right. The **ACeDB** Tablemaker (center right) is used to organize ESTs into biochemical categories (e.g. defence) and gene families (e.g. pathogenesis-related protein gene Cxc750).

The PC model is especially problematic for databases. As illustrated above, it would be convenient to have both sequence database searches and the AceDB software and database on one computer. However, on PCs, if one user is running **fasty** searches, which can take substantial time, no one else can access the database. In any case, the PC model would only allow one user to access the database at any one time. With network-centric computing, any number of **fasty** searches or **ACeDB** clients can be run by any number of users, simultaneously. Where usage is heavy, the load can be spread out across different servers.

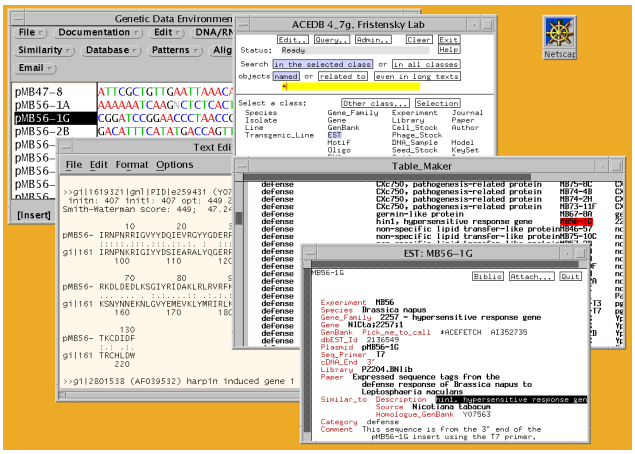

Figure 4. Example of an EST project, managed using ACeDB.

For users interested in implementing an **ACeDB** database for their own lab, a sample database along with documentation can be found at http://home.cc.umanitoba.ca/~psgendb/acedb/acedb.html

Unifying the computing environment

The next step in our chain of logic was to realize that the network-centric model could simplify not only the computation-intensive molecular biology tasks, but all of what are now thought of as common desktop tasks. Essentially all aspects of the Internet, such as Web browsers, the File Transfer Protocol (FTP), USENET newsgroups, and many aspects of email were originally developed under Unix, and later ported to PCs. More recently, office applications have been appearing in increasing numbers for Unix. In my lab, we do our computing in a 100% Unix environment. Every computing task, from record-keeping, data analysis, writing papers, drawing figures, and preparing presentations are all unified seamlessly on a single easy to use graphic desktop via X-windows. As illustrated in Figure 5, this paper and the accompanying figures were written on the same Sun Unix system that BIRCH resides on. The example shows that common desktop programs, including web browsers, mailers, file managers, word processors, and drawing programs all run in the Unix environment. It is also worth noting that an equally comprehensive laboratory computing environment is available at the NIH (Advanced Laboratory Workstation).

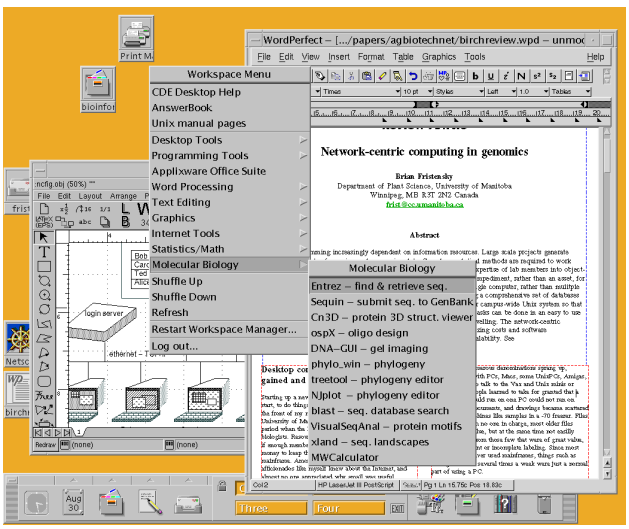

Figure 5. Screen shot of a typical X-windows session. A customized CDE can be opened from anywhere on the screen. Submenus, organized by category, make it easy for users to find and launch programs.

We have avoided Windows NT for reasons of reliability, scaleability, and security (Kirch, 1999), and poor support for terminal sessions. While most major Unix systems already support 64-bit processing, NT will not be a 64-bit system for at least several years. As well, software written for Windows is seldom configurable for users with separate home directories. In most cases, Windows software requires that all users have write access to central configuration directories for each program, meaning that any user could disable the program, and users can't have separate configuration files.

Another advantage of X11/Unix systems is that X-windows sessions can be run remotely. Even in places where X-terminals are not available, X-windows software packages are available to allow PCs to emulate X-terminals. Links to various commercial X-windows packages can be found at http://home.cc.umanitoba.ca/~psgendb/nc/xterm.html. A 'light-weight' X11 protocol is implemented in Virtual Network Computing (VNC), which offloads most of the screen-drawing tasks normally done at the terminal or the PC to the server. At the server end, **vncserver** creates a screen image, which is updated by **vncviewer** running on the remote PC. **Vncviewers** exists for all PC platforms, and there is even a Java-applet version of **vncviewer** that runs in any Java-enabled web browser. Given a fast Internet connection, if **vncserver** is installed on your Unix system, you can run an X-windows session on your home server from a Java-enabled web browser, anywhere in the world. VNC is distributed free by AT&T Research, Cambridge (http://www.uk.research.att.com/vnc/).

The popularity of the laptop points to the need for people to have a consistent set of programs and files no matter where they go. This convenience comes at the price of frequent transfer of data between laptops and PC at home and the lab, upgrading the software, and the fact that it is impractical to keep connecting the laptop to the network as you travel from one place to another. As more of what we do depends on networks, laptops become less useful. With VNC, all you need is a web browser in an airport kiosk or hotel room to access all the same files and applications that you use in the lab.

Finally, training of the user base is a critical component of any computer resource. The network-centric model allows the user to work on their own account during training sessions, and guarantees that the system will work the same way in their lab as it did in the session. The consistency of the computer system across all desktops also makes it possible to have central, web-based documentation that is up-to-date and accurate for all users.

Conclusion

Unix has made it easy to integrate all computing, from common tasks such as word processing, to database access and sequence analysis on a single, easy to use desktop, accessible simultaneously to all workers in the lab from any terminal. In a recent project, our lab sequenced 278 cDNAs from oilseed rape plants inoculated with a fungal pathogen, to identify genes that are induced by fungal attack (Fristensky *et al.*, 1999). Since this was our first EST project, we learned by doing. As the dataset grew, we wrote Java software to automate conversion of **fasty** results into database entries, submission of the results to GenBank, integration of resultant accession numbers back into our database, and generation of a hypertext table of ESTs for online publication. This process also led to several cycles of refinement of the data objects in our lab database, as we learned more about how to conceptualize our data. The coexistence of **ACeDB**, **gde**, word processors and drawing programs all on the same filesystem on the same server made it easy to run analytical programs in **gde** windows to answer our questions about the data as we wrote the manuscript. There is a momentum to writing a paper,

or analysing data, that is interrupted whenever you have to move from one PC to another to work on different parts of a problem. BIRCH, and the larger Unix system of which it is a part, allowed an uninterrupted flow of work for everyone involved in the project. And unlike the -70 freezer, in which samples could belong to anyone, and contain anything, each BIRCH user keeps their files in their own accounts. When a lab worker leaves the lab, their files can be transferred to a new account on the same system. Whoever picks up where they left off will not have the problem that things that worked for one person on one machine won't work for a new person on a different machine.

We so often use the expression "don't reinvent the wheel". Yet, in practice that is exactly what each PC owner does when installing a new PC. As we demand more sophisticated computing resources, it makes sense to move to a more network-centric approach, allowing biologists to simply use computers, and not manage with them.

References

Advanced Laboratory Workstation (ALW) System, Center for Information Technology, National Institutes of Health. http://www.alw.nih.gov

Durbin, R.; Thierry-Meig, J. (1991) *A C. elegans Database*. ftp://lirmm.lirmm.fr/pub/acedb/

Felsenstein J. (1989). PHYLIP Phylogeny Inference Package. *Cladistics* **5**, 164-166.

Fristensky, B. (1993) Feature Expressions: Creating and Manipulating Sequence Datasets. *Nucleic Acids Research*. **21**, 5997–6003.

Fristensky, B. (1999a) Network computing: Restructuring how scientists use computers and what we get out of them. In *Bioinformatics Methods and Protocols* [edited by Misener S.; Krawetz, S.], Totowa, NJ, USA: Humana Press. pp. 401–412,

Fristensky, B. (1999b) Building a multiuser sequence analysis facility using freeware. In *Bioinformatics Methods and Protocols* , Totowa, NJ, USA: Humana Press. pp 131–145.

Fristensky, B.; Balcerzak, M.; He, D.-F.; Zhang, P. (1999) Expressed sequence tags from the defense response of *Brassica napus* to *Leptosphaeria maculans*. *Molecular Plant Pathology Online* http://www.bspp.org.uk/mppol/1999/0301FRISTENSKY.

Galtier, N.; Gouy, M.; Gautier, C. (1996) SeaView and Phylo_win, two graphic tools for sequence alignment and molecular phylogeny. *Computer Applications in the Biosciences* **12**, 543-548.

Kirch, J. (1999) Microsoft Windows NT4.0 Server versus Unix. http://www.unix-vs-nt.org/kirch/

Maciukenas, M. (1994) Treetool. ftp://rdp.life.uiuc.edu/rdp/programs/TreeTool/

Olsen, G.J.; Matsuda, H.; Hagstrom, R.; Overbeek, R. (1994). FastDNAml: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. *Computer Applications in the Biosciences* **10**, 41–48.

The Open Group (1998) *X Window System***.** http://www.camb.opengroup.org/

Pearson, W.R.; Wood, T.; Zhang, Z.; Miller, W. (1997) Comparison of DNA sequences with protein sequences. *Genomics* **46**, 24–36.

Smith, R.F.; Smith, T.F. (1990) Automatic generation of primary sequence patterns from sets of related protein sequences. *Proceedings of the National Academy of Sciences, USA* **87**, 118–122.

Smith, S.; Overbeek, R.; Woese, C.R.; Gilbert, W.; Gillevet, P.M. (1994) The Genetic Data Environment: an expandable GUI for multiple sequence analysis. *Computer Applications in the Biosciences* **10,** 671–675. ftp://megasun.bch.umontreal.ca/pub/gde/