Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

Bio-Linux as a Tool for Bioinformatics Training

Timothy Booth, Mesude Bicak*, Hyun Soon Gweon, Dawn Field
Molecular Evolution and Bioinformatics Group
NERC Centre for Ecology and Hydrology
Wallingford, United Kingdom
tbooth@ceh.ac.uk, mbicak@ceh.ac.uk, hyugwe@ceh.ac.uk, dfield@ceh.ac.uk

Enis Afgan
Center for Informatics and Computing
Ruđer Bošković Institute
Zagreb, Croatia
enis.afgan@irb.hr

***Abstract*—Because of the ever-increasing application of next-generation sequencing (NGS) in research, and the expectation of faster experiment turn-around, it is becoming unfeasible and unscalable for analysis to be done exclusively by existing trained bioinformaticians. Instead, researchers and bench biologists are performing at least parts of most analyses. In order for this to be realized, two conditions must be satisfied: (1) well designed and accessible tools need to be made available, and (2) researchers and biologists need to be trained to use such tools in order to confidently handle high volumes of NGS data. Bio-Linux is a fully featured, powerful, configurable and easy to maintain bioinformatics workstation and helps on both counts by offering well over one hundred bioinformatics tools packaged into a single distribution, easily accessible and readily usable. Bio-Linux is also accessible in the form of virtual images or on the cloud, thus providing researchers with immediate access to scalable compute infrastructure required to run the analysis. Furthermore this paper discusses how bioinformatics training on Bio-Linux is helping to bridge the data production and analysis gap.**

Keywords—bioinformatics; next-generation sequencing; training; cloud computing.

I. INTRODUCTION

A. *The analysis and tools problem*

Bioinformaticians are familiar with the current analysis & tools problem in bioinformatics, exemplified by the observation that a genome sequenced for $1,000 might require $100,000 analysis [1] in a current clinical setting. Researchers, especially PhD candidates and Postdocs, find that they can run sequencing analyses quickly and cheaply but the effective analysis of the resultant data remains hard. Fixed processing pipelines quickly become obsolete in the face of new sequencing technologies, new lab protocols, new questions to ask, new reference databases, and increasing data volumes.

There is an ongoing need for a multitude of flexible, ready-to-use tools, as well as a need for bioinformatics training - empowering researchers with the software and the knowledge to plan and perform analyses - creating the bioinformaticians of the future. The Bio-Linux platform [2] provides researchers an easy way to: (1) set up a Linux-based bioinformatics workstation and (2) get the tools and data installed and configured on the system. Beyond that, the process of performing bioinformatics tasks is difficult, where one needs to deal with errors and subtleties in data and understand the tools, as well as their strengths/weaknesses for a given problem. But with the system set-up taken care of, the researcher is free to focus on these problems.

B. *Learning hurdles in bioinformatics*

Anyone wishing to develop bioinformatics skills and to analyse NGS data effectively faces many learning challenges. Here, we identify two particular hurdles - steep learning curves that a user must overcome to progress. As illustrated in Fig. 1, these are: 1) the move from manual, interactive data manipulation to programmatic manipulation, e.g. from manually editing a batch of files to writing a simple shell loop to make the edits, and 2) the move from working on a single system to working on a Grid or Cluster system, e.g. from running a big set of BLAST searches in series to splitting the query and submitting it to a compute cluster.

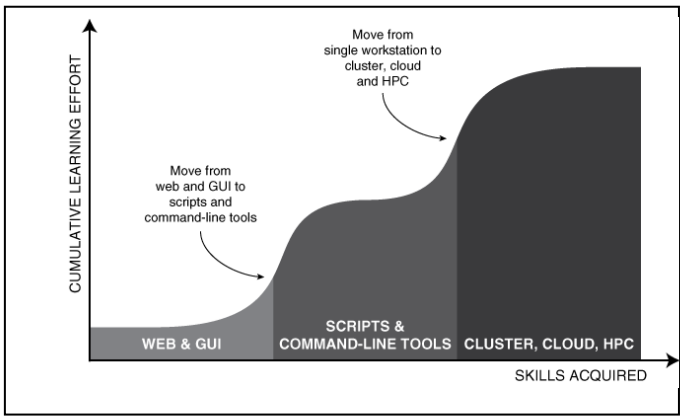

Figure 1. Illustrative depiction of two learning hurdles encountered on the path to acquiring technical skills necessary to analyse NGS data. We contend that these constitute significant barriers to progression for a new bioinformatician.

Both steps are hard because they involve shifts in the way of thinking about the analysis accompanied by a need to work in a new computing environment. In the first case, the user moves from hands-on interactive manipulation of data to using or writing scripts that get the computer to do the repetitive work. Typically, in bioinformatics, this means moving to a Linux environment where scripting is fully supported. In the second case, the move is from a linear way of working to

scheduling jobs in parallel and breaking down problems to "divide and conquer", and also from a single workstation or server to a cluster setup. If the user is required to set up the unfamiliar environment for themselves (i.e. to install unfamiliar tools on an unfamiliar Linux system, as is often the case, or to build or configure a compute cluster) then the problem is compounded. Training is required to move people past these learning hurdles, but tutors find that providing students with a suitable training environment is tricky. Here we will show how Bio-Linux can help with all of these issues.

C. *Other approaches to overcoming the problem*

That the learning hurdles outlined above are real and significant is evidenced by the extensive and varied efforts to avoid or offset them. In the case of the first hurdle, we have seen many efforts to give users a way to do increasingly sophisticated tasks via simple, or what are perceived as simple, graphical user interfaces (GUIs). Galaxy [3, 4, 5], Mobyle [6], and UGene [7] are prime examples that give access to many underlying command-line tools via a GUI. Clearly, a major motivation in creating these is to extend what a researcher can do in interactive mode without making the leap into programmatic data manipulation. However just as clearly the need for bioinformaticians to work on the command line, or otherwise to write high-level code, is not close to being eliminated by existing software, and it is well known that most bioinformaticians having become used to the command line then find it invaluable.

Likewise, there are concerted efforts to enable and seed greater use of Grid and Cloud computing resources (eg. http://www.ngs.ac.uk/campus-champions) and commercial consultancies who get most of their business from enabling access to the Cloud (eg. http://bioteam.net/) because even qualified bioinformaticians face a steep learning curve in grappling with these approaches.

II. Deploying an effective training platform

A. *An overview of Bio-Linux*

The Bio-Linux project was initiated by Professor Dawn Field in 2002 at the NERC Environmental Bioinformatics Centre (NEBC, http://nebc.nerc.ac.uk). Bio-Linux distribution provides easy access to a powerful computing environment (Ubuntu Linux, http://ubuntu.com) pre-loaded with bioinformatics tools from basic data manipulation tools (eg. the EMBOSS suite) to viewers and editors (eg. Artemis, Archaeopteryx) to programming environments (R, BioPerl) to integrated analysis packages (eg. Qiime, Geneious free edition). Installation is approachable even to novice users and updates of the core system and all packages are automated by the APT system, which is native to Ubuntu. Many of the packages are maintained specifically for Bio-Linux, but we also draw on and contribute to the work of Debian-Med [8].

Many of the tools on Bio-Linux can be accessed via the Galaxy interface, which as of Bio-Linux 7.0 is installed as standard. Due to the many steps involved in setting up and maintaining a Galaxy server and the tools it depends on, users wanting to work in a local Galaxy installation were previously in a bind that the effort needed to get Galaxy working could well be greater than the effort saved by using it. Thus it was most practical where a core bioinformatics facility could maintain a central server for many users, which is not a luxury everyone enjoys.

In addition to being an analysis platform, the Bio-Linux project, since its initiation, was also conceived as a teaching platform that would be effective for tutored and untutored learning. In this paper we show how some of the newer features of Bio-Linux make for particularly effective teaching options, and describe how they work in practice.

B. *Difficulties encountered in bioinformatics training*

In the context of classroom-based computer training, there is a need to have access to a consistent system environment, including all the tools and the data, as well as sufficient computing power to handle the analyses being performed by all students. Establishing a fully configured and functional system environment at teaching locations can often be a challenge: installing one new package on the training machines can be troublesome, never mind a full suite of tools. Local machines are likely to be protected by security policy from any new software installations, and most such machines have the MS Windows operating system under which a large number of important bioinformatics tools do not run. Furthermore, visiting researchers will likely not have access to the training resources beyond the training period.

To overcome the steep learning curve associated with beginning bioinformatics on Bio-Linux, it is desirable to allow researchers to learn in an environment that is as similar as possible to the one they will be using in their actual research. Ideally, they should be able to take an exact replica of the training environment to work on after the course in order to continue learning and to start applying their new skills in anger.

C. *Practicalities of running a course on Bio-Linux*

We make use of the in-built ability of the Ubuntu Linux distribution to run "live" from a USB stick. This ability is inherited by Bio-Linux. A teaching machine runs directly from the stick, bypassing the normal operating system on the hard drive while providing a full Bio-Linux environment. Under the hood, this is achieved by having a read-only *squashfs* (http://squashfs.sourceforge.net) root filesystem, overlayed with a second read-write filesystem. The unified filesystem is managed by *aufs* (http://aufs.sourceforge.net) to appear as a single volume, with the result that all changes (home directory, preferences, system configuration, installed packages) will be preserved to the USB stick. This overlaying approach means that the original image can be highly compressed, allowing the full Bio-Linux distribution to fit in around 2GB space, as opposed to ~6GB uncompressed, and this keeps the media costs down since 4GB capacity USB sticks are sufficient. Also, in the unlikely event that a student manages to break their personal system, the USB stick can be quickly reverted to a pristine state simply by erasing the overlay data. Thus we can give everyone full root-level access to the system and assure them that they are free to experiment in a safe, sandboxed environment.

Bio-Linux USB sticks are prepared in batches at the NEBC. It is also possible for researchers to make their own "live" sticks from scratch by downloading the latest image from the NEBC Bio-Linux project website (http://nebc.nerc.ac.uk/tools/bio-linux) and following instructions provided.

In the scenario where a single user is completing self-taught training, they need a regular PC or laptop and a Bio-Linux USB stick. The PC is rebooted so as to boot from the stick, and the Bio-Linux environment will load. The user will immediately see a familiar desktop environment with a "Bio-Linux Tutorials" icon on the desktop.

With co-operation from the system administrator, a room of PCs can be booted into Bio-Linux in around 30 minutes prior to the course, with a USB stick running on each PC. At the end, the USB sticks are removed and the machines are rebooted back to the hard drive, unchanged from their previous configuration. Each attendee retains their USB stick, which can be booted on another PC with all working data in-place, as outlined above. Furthermore, when the live system becomes too limiting, Bio-Linux can be installed onto the hard disk of a target machine, and once again files and settings are preserved and transferred to the installed system so the user is left in the same familiar environment. This is all handled by the *Casper* component of the Ubuntu system.

The limitations of working from a live stick are to do with capacity, speed and reliability. The course is designed so that users do not hit the storage limit of the USB stick, and in the event of hardware failure a stick is simply replaced. For longer courses to be run on the "live" platform, higher capacity or more sturdy USB stick models should be considered.

III. CloudBioLinux and CloudMan

A. The utility of cloud services in bioinformatics

Upon receipt of a large batch of NGS data, a typical Bio-Linux user might prepare to analyse the data using familiar tools on their Linux workstation or local department server, either with command-line tools like Qiime or Velvet, or by using the Galaxy workbench environment in Bio-Linux.

As the size of a typical analysis grows, the data processing is likely to need speeding up by running it in parallel on a cluster. A user is now faced with two problems: firstly, setting up or otherwise gaining access to a suitable cluster environment, and secondly transferring the programs and data to work effectively in this environment. This is compounded by the fact that the working environment for the cluster, typically a shell prompt on an unfamiliar Unix machine, presents a fresh learning curve to the user. This is a particular issue in older style grid systems (e.g., the UK National Grid Service), where the user must go through an involved process to obtain access to the compute resource, then go about ensuring that whatever software and reference data they need is installed. Finally, the data is moved onto the grid system for analysis, where most likely standard bioinformatics tools and file viewers are not available. So, the user has no easy way to manipulate and view the data within the Grid environment. This is before any parallel analysis approach is even planned.

Luckily, an alternative model to acquiring computational resources is now available: Cloud computing. With the advent of cloud computing, computational capacity can be hired as a service from a cloud computing provider in the form of virtual servers, disks etc. These resources can then be customized to compose specialized computational systems in a matter of minutes, so the user can work in their preferred computing environment. What is particularly interesting in this model is that the resources can be hired and configured only for the duration and the scale of computation; once the required computation is completed, the resources can be released just as they were acquired. This offers a suitable platform for training and presents a significant opportunity for researchers who have periodic computational needs because they can gain access, and pay, only for the resources they actually consume.

However, the resources provisioned by the cloud providers, like any newly installed computer, come in the form of generic virtual machines with block storage disks; out of the box, these resources are not properly configured for bioinformatics analyses or training - they first need to be set up and composed into purposeful systems.

B. Making Bio-Linux available in the cloud

The CloudBioLinux project (http://cloudbiolinux.org) [9] builds on the features of Cloud computing and Bio-Linux, and focuses on making the entire Bio-Linux context readily available in the cloud. Namely, CloudBioLinux takes advantage of the ability to provide pre-configured machine images in the cloud context. This allows one to instantiate, on-demand, an exact replica of the system that is preconfigured with the operating system, tools, and data. The end result is that such deployment can be equally used for training purposes as well as later analysis directly by a researcher while obtaining the underlying compute and storage infrastructure directly from the cloud provider.

Technically, CloudBioLinux is a set of well-designed scripts, available as open source software (https://github.com/chapmanb/cloudbiolinux) that is used to generate and configure the machine image with all of its dependencies. The result is captured as an image; a CloudBioLinux image is currently available on Amazon Web Services (AWS, http://aws.amazon.com). As a result, in addition to being available as ready-to-instantiate image on the AWS cloud, the set of scripts allows one to build an exact replica of the system on a different commercial cloud, a private/academic cloud, or in a local virtual machine. Once made available, the machine image can simply be instantiated through a web-based control panel (http://biocloudcentral.org) and accessed via SSH to provide a command line or used as if it was a local computer via the graphical remote desktop interface (FreeNX, http://freenx.berlios.de). Once the need for the particular instance diminishes, at the end of training or upon completed analysis, the instance can simply be terminated.

C. Scaling the infrastructure via the CloudMan platform

Access to a pre-configured system and the required infrastructure represents a big step in the direction of making the training and analysis platform accessible. However, as the scale of problems grows (either by having multiple researchers simultaneously use the system or by running larger analyses), the underlying compute infrastructure also needs to scale. To support such a deployment model, CloudBioLinux is built with CloudMan (http://usecloudman.org) [10, 11] support. CloudMan is an extensible platform and a framework for managing compute clusters in the cloud; it allows one to create a fully functional compute cluster that provides a scalable computational backend for the tools running on top. CloudMan is compatible with a range of cloud providers (AWS, OpenStack, and OpenNebula) and thus continues to build on the accessibility of the Bio-Linux solution.

One of the unique CloudMan features that is particularly useful in the training and reusability context is the ability to share a complete deployment. Namely, each CloudMan instance can be shared as a point in time configuration (in terms of tools, data, and configurations) with individuals or made public. Once shared, a user can start their own instance of the shared CloudMan instance and all of the customizations performed on the shared instance will be automatically available on the derived instance. In addition to sharing instances whose toolset has been customized, researchers can share instances that have had data uploaded and analyzed. This makes it possible to share partial or complete analysis environments, allowing multiple analysis directions to be considered in parallel as well as analysis results made accessible without needing to maintain a live and accessible instance.

C. Training on the cloud

In the context of training, this allows one to create a custom template containing all the required tools, reference data, and sample data. Once created, it is trivial to instantiate multiple replicas of such template. The templates can be instantiated, via a web browser, at a visiting location for the purposes of the training.

Thus, CloudBioLinux provides a fantastic platform for teaching. Just as a teaching room of Bio-Linux workstations can be created with Bio-Linux USB sticks, a teaching cluster can be created on a cloud computing provider such as Amazon EC2 with just a few clicks. Unmodified, the standard CloudBioLinux image is ready to meet a range of teaching needs. Furthermore, the course tutor has the option to configure the build scripts and add any additional files and packages, enabling full control over the system that is being used.

If course attendees would like to have access to the exact training environment even after the training, it is possible to easily instantiate their own instance from any EC2 account - in the same way that they can take home a live USB stick and boot it on their own PC.

IV. Current and proposed training

A. "Introduction to Bioinformatics on Bio-Linux" Course

Our introductory one-day tutorial introduces the new user to the Bio-Linux system. It includes a tour of the desktop, information on finding programs and documentation, and provides experience running graphical and command line tools. The course is made available as a set of notes and a collection of sample files which the students work on. These are pre-loaded onto the default Bio-Linux system (via the *bio-linux-tutorials* package).

Part 1 is an introduction to the Linux desktop and system, emphasising use of the command line. As well as basic file navigation and use of pipe and flow control features, users are shown how standard core utilities (http://www.gnu.org/software/coreutils/) can be applied and combined to manipulate their data files. In practice, an ability to work confidently in a variety of environments (CLI, GUI, web) will best equip a researcher to perform effective bioinformatics analysis. The command line is, for most, the least familiar of these, and providing a platform where all environments are available and can be demonstrated and tried together by course attendees is ideal for learning.

Further components in part 2 of the course take the user into bioinformatics programs on Bio-Linux and provide the information they need to make intelligent choices about the programs and interfaces used to run bioinformatics analyses. This section of the course is modular, and as well as adding our own content we have invited the authors of various bioinformatics tools (Artemis, EMBOSS, Velvet, FastQC) to contribute modules on their software. Contributors found they could easily modify existing tutorials to run within the Bio-Linux environment.

Other courses have been successfully taught on Bio-Linux: An introduction to databases (PostgreSQL), introduction to Perl, and several courses for masters students. All benefit from running off a "live" system as outlined above, and even where course attendees are not familiar with a Linux environment, only a short introduction is needed before they are equipped to start working on the proper course material.

B. Proposals for future training

As with the system itself, all course materials are freely available online (http://nebc.nerc.ac.uk/support/training/course-notes/past-notes/intro-bl6). We are also in the process of making tutorial videos available to encourage independent remote Bio-Linux training. Over the coming year, we will also start to leverage the potential of CloudBioLinux for training activities as outlined above.

Bio-Linux will also host the *Code Catalogue* depicted in Fig. 2, a platform to hold collective scripts in an organized manner, sorted against categories ranging from analysis to visualisation scripts for different domains of biology, to workflows and cloud computing related scripts. The unifying properties of the collection will be that each script will perform a single defined task and be ready to run straight away within the Bio-Linux environment.

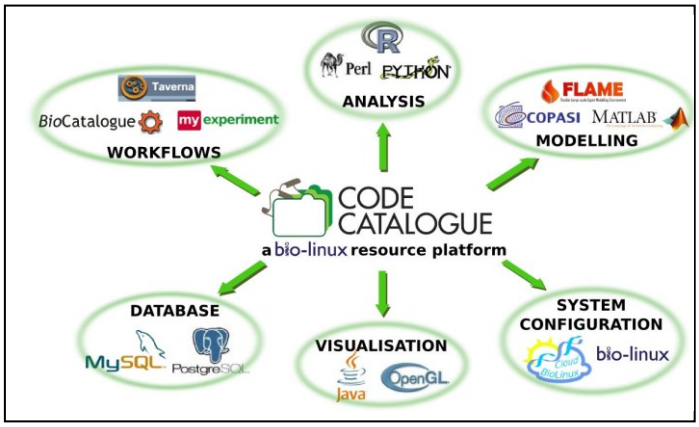

Figure 2. Code Catalogue, a community resource platform.

V. Community Outreach and Usage Figures

At NEBC, we greatly value community outreach and feedback, and provide support for any Bio-Linux related queries via the NEBC Helpdesk (helpdesk@nebc.nerc.ac.uk). There are also Bio-Linux User and Developers discussion lists, and annual User Group Meetings take place at Hacketons organized prior to the Bioinformatics Open Source Conference (BOSC, http://www.open-bio.org/wiki/BOSC_2012).

Over the 9-year lifespan of Bio-Linux, it has become a globally popular workstation. We monitor uptake of Bio-Linux via a variety of metrics, where the figures are as follows:

- 3400 registered ISO downloads (since 6.0 release). The total number of downloads is larger but unknown.
- 182 people on the discussion list, 50 subscribed to developers list
- 5000 distinct IP hits on package repository (typical month)
- 1200 page views on main website (typical month)
- 180 course attendees per year

One other aspect of providing publicly available course material and videos is to encourage the formation of "Bio-Linux Teaching Groups". As a result, Bio-Linux servers have started to emerge, which are being used for teaching and training purposes at universities.

All software and materials described in this paper are freely available and freely re-distributable. Most code is also open-source, under various OSI-approved licenses. It is possible to obtain Bio-Linux and CloudBioLinux from the project websites, http://nebc.nerc.ac.uk/tools/bio-linux and http://cloudbiolinux.org.

Acknowledgment

The authors acknowledge the many past Bio-Linux developers, as well as CloudBioLinux developers Ntino Krampis and Brad Chapman, the Galaxy Team (http://wiki.g2.bx.psu.edu/Galaxy_Team), the Debian-Med Team (http://wiki.debian.org/DebianMed), and course material contributors Peter Rice (EMBOSS), Matthias Haimel (Velvet), Felix Krueger (FastQC), Daniel Pass (Qiime).

References

[1] E. R. Mardis, "The $1,000 genome, the $100,000 analysis?" Genome Medicine, vol. 2, pp. 84, 2010.

[2] D. Field, B. Tiwari, T. Booth, S. Houten, D. Swan, N. Bertrand, M. Thurston, "Open software for biologists: from famine to feast" Nat. Biotechnol, vol. 24, pp. 801-803, 2006.

[3] J. Goecks, A. Nekrutenko, J. Taylor and The Galaxy Team, "Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences" Genome Biol, vol. 11, pp. R86, 2010.

[4] D. Blankenberg, G. Von Kuster, N. Coraor, G. Ananda, R. Lazarus, M. Mangan, A. Nekrutenko, J. Taylor, "Galaxy: a web-based genome analysis tool for experimentalists" in Current Protocols in Molecular Biology, Chapter 19, Unit 19.10.1-21, 2010.

[5] B. Giardine, C. Riemer, R. C. Hardison, R. Burhans, L. Elnitski, P. Shah, Y. Zhang, D. Blankenberg, I. Albert I, J. Taylor, W. Miller, W. J. Kent, A. Nekrutenko, "Galaxy: a platform for interactive large-scale genome analysis" Genome Research, vol. 15, pp. 1451-5, 2010.

[6] B. Neron, H. Ménager, C. Maufrais, N. Joly, J. Maupetit, S. Letort, S. Carrere, P. Tuffery, C. Letondal, "Mobyle: a new full web bioinformatics framework" Bioinformatics, vol. 25, pp. 3005-3011, 2009.

[7] D. S. Bennett and A. Stam, "EUGene: A Conceptual Manual." Int. Interact, vol. 26, pp. 179-204, 2000.

[8] S. Möller, H. N. Krabbenhöft, A. Tille, D. Paleino, A. Williams, K. Wolstencroft, C. Goble, R. Holland, D. Belhachemi, C. Plessy, "Community-driven computational biology with Debian Linux" BMC Bioinformatics, vol. 11, pp. S5, 2010.

[9] K. Krampis, T. Booth, B. Chapman, B. Tiwari, M. Bicak, D. Field, K. E. Nealson, "Cloud BioLinux: Pre-configured and on-demand bioinformatics computing for the genomics community" BMC Bioinformatics, vol. 13, pp. 42, 2012.

[10] E. Afgan, D. Baker, N. Coraor, B. Chapman, A. Nekrutenko, J. Taylor, "Galaxy CloudMan: Delivering cloud compute clusters" BMC Bioinformatics, vol. 11, pp. S4, 2010.

[11] E. Afgan, D. Baker, N. Coraor, H. Goto, I. M. Paul, K. D. Makova, A. Nekrutenko, J. Taylor, "Harnessing cloud computing with Galaxy Cloud" Nat. Biotechnol, vol. 29, pp. 972-974, 2011.