Interacting with bionformatics functionality

Geir Kjetil Sandve,
institute of informatics, UiO

Alternative ways of doing bioinformatics

- Word
- Excel
- UNIX shell
- Programming language
- Tool-specific graphical interface (web page)
- Graphical analysis framework (web-based)
- Commercial analysis software

Word

- Does the DNA sequence "gagacagagg" exist within gene body of BRCA1?
- But, extremely limited for analysis..

Excel

- How long are the genes in human chromosome one?
- Can do a bit, have good control, but limited versatility

UNIX command line

- Must separate two things:
  - **A way of interacting with the operating system:** logging in to servers, managing files and running programs (alternative to Windows)
  - **Using specific unix tools:** knowing syntax and usage of tools like grep, wc, sek, awk

UNIX to run programs

- Flexible way to run programs, handle data
- But:
  - May be laborious to install programs, learn syntax etc
  - Often a bit clumsy to share data and collaborate on analysis procedures
  - May need access to a server

UNIX to run programs (cont.)

- In some ways a low-threshold approach:
  need only a minimum of infrastructure
  - **But:** in order to do it well, a deep competence is needed
    (since all best practices must be devised from scratch)
- Anyway not difficult to learn pure basics
  (a few commands like ssh, cd and program names)

UNIX commands and more advanced UNIX syntax

- Flexible, but with limitations in practice (don't even try to do something really complex)
- Often irreproducible analyses and hard to reuse previous code
- Sometimes hard to guard against hidden errors (hard to follow good development practices)
- Based on recollection, not intuitive and quickly forgotten - frustrating if only used intermittently

UNIX commands and more advanced UNIX syntax (cont.)

- After two weeks of using UNIX, how would you solve a simple problem like plotting gene lengths? (which was straightforward with Excel)
- Beware: Spend a finger (at a time), and you risk spending a full arm..
- May be appropriate with frequent usage and well suited problems, but not a pragmatic choice for all

Programming language

- Ultimate flexibility - you can do anything
- Handles complexity gracefully - you can do things well
- Python is easy to learn, efficient to code in, scales well
- *Invest a little more initial effort than UNIX shell,*
  to get something that you will remember your lifetime,
  and that adapts and scales very well

R

- Often used interactively, similar to unix shell
- Fantastic library
  - Arguably more extensive, more fine-grained, easier to install and better to interact with than UNIX commands/programs
- Also a general programming language (albeit a sucky one)
- Alternative to use of UNIX tools,
  not running large programs..
- *If you need interactive analysis and a little programming, this may give you most bang for the buck*

Tool-specific graphical interface

- Stand-alone program or custom web page
- No installation
- Typically better documentation (help, tutorials)
- You can browse and recognize possibilities, instead of having to recall them
- But:
  - Must learn peculiarities for each interface
  - Often covers only core functionality, not pre- and post-processing
  - Cumbersome interoperability between programs

Graphical analysis framework

- Stand-alone software suite or web-based platform
- Unified *(and typically powerful)* interface - learn once
- Often support for intermediate functionality
- Often scales well computationally
- Often excellent data handling, provenance, sharing
- Typically good for reproducibility and cooperation
- **But**: may not contain the functionality you need

Commercial analysis software

- Typically very user-friendly
- Good support
- But:
  - Lock-in is often an issue
    (often not interoperable - important to keep customers..)
  - Typically not transparent (crucial details may be hidden)
  - Often hard to customize

What's best for you?:

if bioinformatics is low priority

- Start using web-based solutions in an instant!
- If what you need isn't there, request it (and do something else in between)
- If your request isn't met, be careful not to spend too much effort in a side-track
  - Use colleagues or use R
  - Excel might also be useful for simple tasks
  - Give a quick try at installing and running program (give up quickly)
  - If you really need to: use UNIX tools (or programming language)

What's best for you?:

if bioinformatics is important, but intermittent

- If web-based solution is available - save time and use it!
- Look ahead and request functionality before you need it
- Use R or Python (depending on balance between quick analyses and general programming)
- Use basic command line to run programs, but probably avoid more advanced UNIX:
  - You may waste a lot of time on recollecting, trying and failing - looking efficient but not being effective

The Lifeportal at UiO

- Web-based framework based on Galaxy
- Offers thousands of tools and operations (Galaxy)
- Backed by Abel
- Can get help with using most productively
- Can request (and possibly get) new functionality
- May allow effective, collaborative, reproducible science

Conclusion

- Use the simplest (for you) tools suited for the job!
- Do what is effective, not what looks efficient (cool)
- It's always good to learn, but it's not necessarily best spent effort
- The Lifeportal might offer (or be extended with) what you need, allowing you to focus on your task