CS 175: Introduction to Bioinformatics for Genomics and Transcriptomics

Lecture Slides

Pipelining

Jan Michael C. Yap

Algorithms and Complexity Laboratory
Department of Computer Science
University of the Philippines, Diliman
janmichaelyap@gmail.com

Lesson 9

Pipelining

Pipelining

- Motivations
- Setting Up A Pipeline
- Final Notes

Pipelining

Pipelining

Motivations

Setting Up A Pipeline

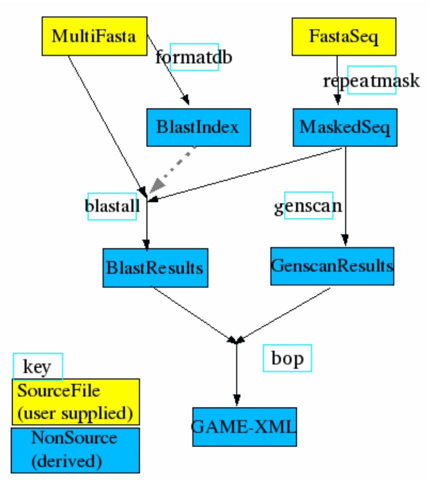
Final Notes

On pipelining

"In bioinformatics it is very common to end up with a lot of small scripts, each one with a different scope - plotting a chart, converting a file into another format, execute small operations - so it is very important to have a good way to [g]lue them together, to define which should be executed before the others and so on." - Giovanni M Dall'Olio

https://www.biostars.org/p/79/

Pipeline



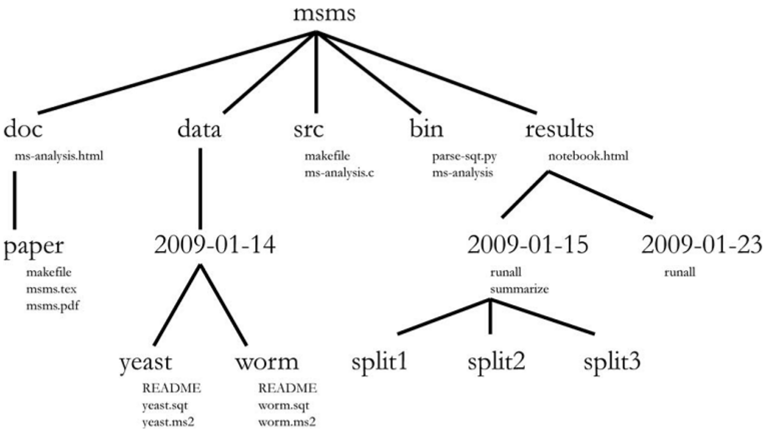
http://skam.sourceforge.net/skam-intro.html

Setting Up A Pipeline: A Quick Guide

- "Someone unfamiliar with your project should be able to look at your computer files and understand in detail what you did and why. This "someone" could be any of a variety of people... Most commonly, however, that "someone" is you."
- Main concerns
  - File and Directory Organization
  - Documentation
  - Carrying out an Experiment
  - Error Management
  - Software Development

Noble WS (2009). A Quick Guide to Organizing Computational Biology Projects. PLoS Computational Biology 5(7): e1000424.

File and Directory Organization



msms

doc
ms-analysis.html

paper
makefile
msms.tex
msms.pdf

data

2009-01-14

yeast
README
yeast.sqt
yeast.ms2

worm
README
worm.sqt
worm.ms2

src
makefile
ms-analysis.c

bin
parse-sqt.py
ms-analysis

results
notebook.html

2009-01-15
runall
summarize

split1

split2

split3

2009-01-23
runall

Noble WS (2009). A Quick Guide to Organizing Computational Biology Projects. PLoS Computational Biology 5(7): e1000424.

Documentation

- "Entries in the notebook should be dated, and they should be relatively verbose, with links or embedded images or tables displaying the results of the experiments that you performed. In addition to describing precisely what you did, the notebook should record your observations, conclusions, and ideas for future work."
- "Particularly when an experiment turns out badly, it is tempting simply to link the final plot or table of results and start a new experiment. Before doing that, it is important to document how you know the experiment failed, since the interpretation of your results may not be obvious to someone else reading your lab notebook."

Noble WS (2009). A Quick Guide to Organizing Computational Biology Projects. PLoS Computational Biology 5(7): e1000424.

Carrying Out An Experiment

- Record every operation that you perform
- Comment generously
- Avoid editing intermediate files by hand
- Store all file and directory names
- Use relative pathnames to access other files within the same project
- Make the script restartable

Noble WS (2009). A Quick Guide to Organizing Computational Biology Projects. PLoS Computational Biology 5(7): e1000424.

Error Management

- Write robust code to detect errors
- When an error does occur, abort
- Whenever possible, create each output file using a temporary name, and then rename the file after it is complete

Noble WS (2009). A Quick Guide to Organizing Computational Biology Projects. PLoS Computational Biology 5(7): e1000424.

Software Development

- Four program/script categories
  - Driver
  - Single-use
  - Project specific
  - Multi-project
- "[E]very script or program, no matter how simple, should be able to produce a fairly detailed usage statement that makes it clear what the inputs and outputs are and what options are available."
- Version control for back-up, historical bug tracking, and facilitating collaborative projects

Noble WS (2009). A Quick Guide to Organizing Computational Biology Projects. PLoS Computational Biology 5(7): e1000424.

Other Words of Wisdom

- "Overall... pragmatism is [an] important factor in computational biology. Just do enough to get the results you need and only invest more time when it's necessary." - Michael Barton
- "The most important thing for me has been keeping a README file at the top of each project directory, where I write down not just how to run the scripts, but why I wrote them in the first place – coming back to a project after a several-month lull, it's remarkable difficult to figure out what all the half-finished results mean without detailed notes." - Eric Talevich

https://www.biostars.org/p/79/

Other Words of Wisdom

- "I generally use a simple shell script if I have multiple commands or scripts to run. I also try to make a notes.txt file to remind myself of what I did. Doesn't take long and comes in handy." - Madelaine Gogol
- "Something I've found useful... is to set up a framework that allows me to run specific functions in a module from the command line. This allows all the similar scripts (usually data aggregation, plotting, and analysis) to be put into a single module and each step ran from the command line. If I want to make a pipeline I just use a simple shell script." - Sequencegeek

https://www.biostars.org/p/79/

Ch- Ch- Check these out!

- GNU/Make and Bioinformatics: http://www.slideshare.net/giovanni/makefiles-bioinfo
- Ruffus: https://code.google.com/archive/p/ruffus/
- snakemake: https://bitbucket.org/johanneskoester/snakemake/wiki/Home
- Galaxy: https://galaxyproject.org/

Questions?

END OF CS 175 LECTURES!