Descriptions of Common Scripts and Workflow

Unix shell scripts carry out a list of instructions and can automate any process depending on your scripting abilities and sophistication. For those unfamiliar with shell scripts here is a very basic definition of four different types that represent a complete workflow working with an Amazon instance or server.

- Toolkit Script – Install Software and Script Libraries
  - Unless you purchase an image with your toolkit installed, Amazon cloud instances are typically an empty OS which must then be populated with data and software. Storing large files for long-term projects in Amazon storage space is handy, but starting up instances and loading them with tools can be accomplished easily with a single script. This very simple type of script automates the keyboard instructions you would enter to download, unzip, install, and copy every commonly used program.
- Populate Data
  - Files hosted remotely through FTP sites like those maintained by NCBI and flybase.org can be transferred by a script run from within the instance.
  - Files hosted locally can be sent to the instance with a script run locally. Avoided sending large sequence files unless on a university network because broadband services don't give you much upload capacity and large files may take a long time. Remember with Amazon Web Services you pay by the hour. A better solution is to upload large files once and store them within an Amazon storage volume. Small scripts and files under 10M are small enough to be sent locally on any connection. Such files include the shell and python scripts you run on your instance, small software tools that perform limited functions.
- Assemble, Align, Parse, and Otherwise Manipulate Data
  - This script contains the activities that require large amounts of memory or processing power necessitating the use of servers or cloud instances. An example script that I use will generate and BLAST a large number of assemblies of various k-mer lengths, create a .csv file of resulting names, query sequences, and expected values which are then compared to a candidate list. There are about six different operations which are then repeated a specified number of times, which is very easy to script using a basic loop.
- Retrieve Generated Data
  - Run locally this script will download data produced on your instance. This can be annoying if you have a number of files in different directories. Creating a script that will pull individual files from numerous directories and give them unique, descriptive names is a handy solution. The data produced during the processing step uses a naming convention also used by the data retrieval script, I can easily pull all of the relevant data or transmit it to a storage volume with the same descriptive identifiers.

Another advantage of automating this complete workflow I can create an overabundance of data and test parameters which I wouldn't ordinarily have time to mess with to see if they have a positive effect. Instead of being tied to a computer terminal all day I can generate and parse a massive amount of data and examine the output statistics log generated by the script to see if there are any interesting hits. By including scripts that remove redundancy and append new sequences it is possible to generate a growing, organized list of ESTs.

Resources

SEQanswers Wiki and Forum

The wiki describes the function of many useful tools and links to the files. There are also great how-to articles which describe some basic workflows and provide reviews for various software packages. The forum is a great resource for advice, help, and scripts from some very talented members.

BioPython Tutorial and Cookbook

Python is a very useful language for parsing sequence data. BioPython has many scripts useful for bioinformatics as built-in functions allowing for easy sequence conversion, generating histograms and statistics, etc. It also has an installation guide.

Python Documentation - docs.python.org/index.html

Contains tutorials and a database of every aspect of Python and sample usages. A bit technical for a quick-reference, but very complete.

LinuxCommand - linuxcommand.org/writing_shell_scripts.php

Beginners guide to shell scripts. How to create workflows and pipelines that will automate otherwise time-consuming tasks and using powerful tools to process large quantities of data quickly.

Software Carpentry – software-carpentry.org

Beginners guides to various programming languages and shell scripts.

Stack Overflow – stackoverflow.com

A question and answer site for any programming language. Posting a question will usually result in a quick reply or link. Very useful for cleaning up scripts and problem solving. Many beginner questions have already been asked and answered with useful snippets of code or scripts.

SeqHack – seqhack.blogspot.com

Scripts related to my bioinformatic projects will be archived at this website.