Deploying bioinformatics tools with CloudBioLinux

Brad Chapman
Bioinformatics Core,
Harvard School of Public Health
https://github.com/chapmanb

30 June 2014
Outline

- Overview of CloudBioLinux
- Create custom set of tools
- Install from custom flavor
- Add a new tool
What is CloudBioLinux?

Infrastructure for installing biological software
- deb/rpm packages
- Bio-Linux
- Linuxbrew with homebrew-science
- Python, Ruby, R package management
- Conda + Binstar [https://conda.binstar.org/](https://conda.binstar.org/)
- Custom installation scripts
Key Features

- Community
- Curation
Community

https://github.com/chapmanb/cloudbiolinux
History

Integration of multiple efforts

- JCVI Cloud Bio-Linux
- Bioperl Max
- Infochimps machetEC2
- Bio-Linux
- DebianMed
Overcome bare-metal problem with AWS images

- Ubuntu
- Single AMI with biological tools
- Automated build infrastructure
- Bring in developer community
- Ready to use for researchers
Biological data

- Genomes, organized and indexed
- Associated data files: dbSNP, reference transcripts
- S3 bucket
- Tools with organized data
- GEMINI: https://github.com/arq5x/gemini
Local installation

- Multiple platforms: Ubuntu, RedHat/CentOS, Debian, ScientificLinux
- Isolated installations: no sudo, non-VM environments
- Rapid turnaround for fixes
Flavors: customized installations

- Target specific use case
- Sub-collection of packages from full distribution
- Example:
  cloudbiolinux/contrib/flavor/biopython

Pjotr Prins
Hidden infrastructure

bcbio-nextgen

- CloudBioLinux drives fully automated installation
- Reproducible build scripts for docker migration

https://github.com/chapmanb/bcbio-nextgen
Containers

http://docker.io/
https://github.com/chapmanb/bcbio-nextgen-vm
Galaxy toolshed integration vision

- CloudBioLinux flavor to install tools
- Install in isolated Docker container
- Galaxy support for Docker

Manifest

- Full manifest of installed software
- Prioritize biological software
- YAML format for parsing and downstream queries
Outline

- Overview of CloudBioLinux
- Create custom set of tools
- Install from custom flavor
- Add a new tool
CloudBioLinux architecture

- YAML configuration
- Flavors
- Fabric scripts
- Documentation
What is a flavor?

- Subset of full CloudBioLinux packages
- Defined set of packages for a task
## YAML configuration: directory

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>packages-debian.yaml</td>
<td>Remap debian packages not available from latest ubuntu list. Thanks t…</td>
</tr>
<tr>
<td>packages-homebrew.yaml</td>
<td>Update to latest version of snpEff, GEMINI and VEP. Prefer brew insta…</td>
</tr>
<tr>
<td>packages-nix.yaml</td>
<td>Disable default use of Nix Packages</td>
</tr>
<tr>
<td>packages-scientificlinux.yaml</td>
<td>Fill in some missing gaps between Ubuntu and CentOS/SL (base R). Add …</td>
</tr>
<tr>
<td>packages-yum.yaml</td>
<td>Ensure ruby installed for bootstrapping homebrew on bare machines dur…</td>
</tr>
<tr>
<td>packages.yaml</td>
<td>Avoid estscan install which is problematic on Ubuntu 14.04. Fixes #163</td>
</tr>
<tr>
<td>perl-libs.yaml</td>
<td>Mega patch of fixes to get installation working cleanly on localhost …</td>
</tr>
<tr>
<td>puppet_classes.yaml</td>
<td>Initial work LWR integration.</td>
</tr>
<tr>
<td>python-libs.yaml</td>
<td>fixed issues with libraries missing from pypi by adding --allow-unver…</td>
</tr>
<tr>
<td>r-libs.yaml</td>
<td>updated yaml; sacCer2 =&gt; sacCer3</td>
</tr>
<tr>
<td>ruby-libs.yaml</td>
<td>Mega patch of fixes to get installation working cleanly on localhost …</td>
</tr>
</tbody>
</table>
YAML configuration: example

```yaml
# Packages available in the Homebrew and Linuxbrew package manager
---
bio_nextgen:
  alignment:
    - bwa
    - bowtie2
    - novoalign
    - rna-star
  utilities:
    - bamtools
    - bedtools
    - cramtools
    - libmaus
    - biobambam
    - fastqc
    - fastx_toolkit
    - qualimap
    - sambamba
    - staden_io_lib
  analysis:
    #-
    cufflinks
    - samtools
    - htslib
    - bcftools
    #-
    tophat
  variant:
```
## Example flavor

**branch:** master

**cloudbiolinux / contrib / flavor / ngs_pipeline_minimal / +**

**Update to latest version of snpEff, GEMINI and VEP. Prefer brew insta...**

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>custom.yml</td>
<td>Remove pbgzip which fails to compile on Mac and is not currently used...</td>
</tr>
<tr>
<td>main.yml</td>
<td>Add bioconductor libraries to bcbio-nextgen installation flavor to st...</td>
</tr>
<tr>
<td>packages-homebrew.yml</td>
<td>Update to latest version of snpEff, GEMINI and VEP. Prefer brew insta...</td>
</tr>
<tr>
<td>r-libs.yml</td>
<td>Add additional R libraries for cn.mops: snow and rtracklayer. Update ...</td>
</tr>
</tbody>
</table>
---
# Flavor containing with minimal instructions to install tools for
# running next-generation sequencing pipelines.
packages:
  - minimal
  - libraries
    - python
    - java
    - r
    - ruby
    - bio_nextgen
libraries:
  - r-libs
Edit set of brew installed packages

```bash
# Packages available in the Homebrew and Linuxbrew package manager

---

bio_nextgen:
  alignment:
    - bwa

#   - bowtie2 (2.2.0 doesn't work with Tophat, so use it
    - novoalign
    - rna-star

utilities:
  - bamtools
  - bedtools
  - cramtools
  - libmaus
  - biobambam
  - fastqc
  - qualimap
  - sambamba
  - samblaster
  - seqtk==HEAD
  - speedseq
  - staden_io_lib
```
# Global installation directory for packages and standard programs
system_install = /usr/local

# Local install directory for versioned software that will not
# be included in the path by default
local_install = /usr/local/share

# Shell to be used by CBL scripts during runtime
shell_config = ~/.bashrc
shell = /bin/bash -i -c

# Global setting for using sudo; allows installation of custom packages
# by non-privileged users.
# *Note*: ``system_install`` needs to point to a user-writeable directory if
# ``use_sudo`` is set to ``False``
use_sudo = True

# -- Details about reference data installation

# Path where biological reference data files should be retrieved to
data_files = /mnt/biodata
Outline

- Overview of CloudBioLinux
- Create custom set of tools
- Install from custom flavor
- Add a new tool
Setup: get CloudBioLinux and Fabric

Retrieve source and fabric for execution

```
$ git clone https://github.com/chapmanb/cloudbiolinux.git
$ pip install fabric
```

Short demonstration flavor

contrib/flavor/demo
Install

Single command

$ cd cloudbiolinux
$ fab -H localhost install_biolinux:flavor=demo
$ tree -d -L 2 ~/tmp/cbl_demo/
/home/chapmanb/tmp/cbl_demo
|-- bin
| `-- Cellar
|     |-- bedtools
|     |    |-- bwa
|     |    `-- gatk-framework
|     `-- samtools
|-- include
| `-- bam -> ../Cellar/samtools/0.1.19/include/bam
|-- lib
     `-- pkgconfig
|-- Library
     |-- Aliases
    |-- Contributions
     |    `- ENV
     |-- Formula
     |-- Homebrew
     |-- LinkedKegs
     `-- Taps
|-- opt
     |-- bedtools -> ../Cellar/bedtools/2.19.1
     |    `-- bwa -> ../Cellar/bwa/0.7.9a
     |    `-- gatk-framework -> ../Cellar/gatk-framework/3.1-1
     `-- samtools -> ../Cellar/samtools/0.1.19
`-- share
     |-- doc
     |    `-- java -> ../Cellar/gatk-framework/3.1-1/share/java
     |-- man
     `-- samtools -> ../Cellar/samtools/0.1.19/share/samtools

Isolated install directory
Update paths to include automatically

export PATH=~~/tmp/cbl_demo/bin:$PATH
export LD_LIBRARY_PATH=~/tmp/cbl_demo/lib:$LD_LIBRARY_PATH
export PERL5LIB=~/tmp/cbl_demo/lib/perl5:
   ~/tmp/cbl_demo/lib/perl5/site_perl:${PERL5LIB}
Run

$ bedtools
$ samtools
$ gatk-framework
Outline

- Overview of CloudBioLinux
- Create custom set of tools
- Install from custom flavor
- Add a new tool
Tool add options

- Add to any existing packaging community
  - DebianMed
  - Bio-Linux
  - Homebrew

- Custom python code

- Will show example with brew recipe
Homebrew/Linuxbrew

Linuxbrew

A fork of Homebrew for Linux

Install Linuxbrew (tl;dr)

Paste at a Terminal prompt:

```
ruby -e "$(wget -O- https://raw.githubusercontent.com/Homebrew/homebrew/go/install)"
```

See Dependencies and Installation below for more details.

Features

- Can install software to a home directory and so does not require sudo
- Install software not packaged by the native distribution
- Install up-to-date versions of software when the native distribution is old
- Use the same package manager to manage both your Mac and Linux machines

https://github.com/Homebrew/homebrew
https://github.com/Homebrew/linuxbrew
homebrew-science

https://github.com/Homebrew/homebrew-science
**Homebrew repository for CloudBioLinux:** incubator for formulas to end up in homebrew-science — Edit

- **67 commits**
- **1 branch**
- **0 releases**
- **3 contributors**

### Branch: master - homebrew-cbl / +

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Author</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>README.md</td>
<td>Add builds for freebayes associated tools from GitHub revisions: vcf...</td>
<td>chapmanb</td>
<td>7 months ago</td>
</tr>
<tr>
<td>blobbam.rb</td>
<td>VEP: ensure plugin directory included in library. blobbam: ensure w...</td>
<td>chapmanb</td>
<td>22 days ago</td>
</tr>
<tr>
<td>cramtools.rb</td>
<td>Update cramtools jar sha1 to match latest upstream update with exclud...</td>
<td>chapmanb</td>
<td>27 days ago</td>
</tr>
<tr>
<td>treebayes.rb</td>
<td>Ensure FreeBayes version set prior to build, for systems with differe...</td>
<td>chapmanb</td>
<td>2 months ago</td>
</tr>
<tr>
<td>gatk-framework.rb</td>
<td>Update gatk-framework to latest version with fix for quoted shell arg...</td>
<td>chapmanb</td>
<td>a month ago</td>
</tr>
<tr>
<td>glia.rb</td>
<td>Revert glia back to pre-gssw version to avoid compile errors. Fixes #4...</td>
<td>chapmanb</td>
<td>4 months ago</td>
</tr>
<tr>
<td>hall-lab-sv-tools.rb</td>
<td>Provide recipe for structural variation scripts from the Hall lab</td>
<td>chapmanb</td>
<td>a month ago</td>
</tr>
<tr>
<td>libmaus.rb</td>
<td>Ensure pkg-config installed as part of libmaus install so biobambam c...</td>
<td>chapmanb</td>
<td>22 days ago</td>
</tr>
<tr>
<td>platypus-variant.rb</td>
<td>Avoid naming conflict for platypus with OSX build tool</td>
<td>chapmanb</td>
<td>4 months ago</td>
</tr>
<tr>
<td>ma-star.rb</td>
<td>Update ma-star to fixed z4 version.</td>
<td>chapmanb</td>
<td>2 months ago</td>
</tr>
</tbody>
</table>

For more details, visit: [https://github.com/chapmanb/homebrew-cbl](https://github.com/chapmanb/homebrew-cbl)
Simple recipe

```ruby
require 'formula'

class Vt < Formula
  homepage 'https://github.com/atks/vt'
  version '2014-04-23'
  url 'https://github.com/atks/vt.git', :revision => '22894f949a'

  def install
    system 'make'
    bin.install 'vt'
  end

  test do
    system 'vt'
  end
end
```

https://github.com/chapmanb/homebrew-cbl/blob/master/vt.rb
require 'formula'

class Vep < Formula
  homepage 'http://ensembl.org/info/docs/variation/vep/index.html'
  version '75_2014-05-12'
  url 'https://github.com/Ensembl/ensembl-tools/archive/771dafa196c357145be7016c91e1155ae7c621f2.zip'
  sha1 '141a0c639c442bf062d15846fab534de58dec3e4'

  resource "plugins" do
    url "https://github.com/ensembl-variation/VEP_plugins/archive/2c123faff2deef07ee694984fc44e19c48975af4.zip"
    sha1 '0569239ed8255d277db634d838d6ec51b98481a8'
  end

  resource 'loftee' do
    url 'https://github.com/konradjk/loftee/archive/545cf9ac5f25b6a6872984dd1a3197a7e7caf086.zip'
    sha1 'a51c6196964526becfbc7afa9838d0495d7996e9'
  end

  def install
    # VEP
    inreplace 'scripts/variant_effect_predictor/variant_effect_predictor.pl' do |s|
      s.gsub! 'use lib $Bin;', "use lib '$(prefix)/lib';\nuse lib '${prefix}/lib/Plugins';\n"
      s.gsub! "my $default_dir = join '/', (ENV['HOME'], '.vep')", "my $default_dir = '${prefix}/lib';"
    end

    inreplace 'scripts/variant_effect_predictor/INSTALL.pl' do |s|
      s.gsub! "$DEST_DIR ||= '.'", "$DEST_DIR ||= '${prefix}/lib';"
      s.gsub! "$CACHE_DIR ||= ENV['HOME'] ? ENV['HOME'].'/.vep' : 'cache';", "SCACHE_DIR ||= '#{share}/data';"
    end

    inreplace 'scripts/variant_effect_predictor/convert_cache.pl' do |s|
      s.gsub! 'use strict;', 'use strict;\nuse lib '${prefix}/lib';"'
    end

    prefix.install Dir["scripts/variant_effect_predictor/", '.pl']
    bin.install_symlink prefix / 'variant_effect_predictor.pl'
    bin.install_symlink prefix / 'filter_vep.pl'
    bin.install_symlink prefix / 'INSTALL.pl' => 'vep_install.pl'

https://github.com/chapmannb/homebrew-cbl/blob/master/vep.rb
Recap

- Overview of CloudBioLinux
- Create custom set of tools
- Install from custom flavor
- Add a new tool