Source Code Management with git

Genecats Meeting
9 June 2010

Ann Zweig & Galt Barber

Why switch?

- Source tree too large for CVS
- We need new features:
  - Faster checkout
  - File & directory renaming
  - Easier branching
  - Easier merging

Why git?

- We considered:
  - svn
  - mercurial
  - bazzar
  - git
- svn is very CVS-like
- The other three offer branch-based development, but git is the fastest
- We will use branches to manage release of trackDb and static docs

The Logistics

- Friday, June 11 @ 5pm
  - freeze CVS repository
- June 12-13:
  - Galt will convert code base to git
- Monday, June 14
  - git-only access to code

References

- Genomewiki page
  - http://genomewiki.cse.ucsc.edu/index.php/Getting_Started_With_Git
- Other web resources
  - http://git-scm.com
  - http://wiki.freegeek.org/index.php/Git_for_dummies
- Reference books
  - *Version Control with Git* - Jon Loeliger (O'Reilly)
  - *Pragmatic Version Control Using Git* - Travis Swicegood (Pragmatic Bookshelf)

Mirror Sites

- Encouraged to switch to git
- Providing a CVS-server to our git repository for a few months
- Providing read-only public access via
  - Git-daemon

```
git clone \
git://genome-source.cse.ucsc.edu/kent.git kent
```

  - http for those with strict firewalls

```
git clone \
http://genome-source.cse.ucsc.edu/kent.git kent
```

- Chin will build new mirror using git code

Questions?

- Questions about the logistics?

How to set up your own repository

- Clone from the central repository

```
ssh hgwdev
cd $HOME
mv kent kent-cvs
git clone
   $USER@hgwdev.cse.ucsc.edu:
   /data/git/kent.git kent
```

- This URL will work anywhere on the planet

How it's different from CVS

- Each user has own copy of the repository so all the history is available
- Work and commit to your local repo
- Stage related changes in a single commit
- Easier branching and merging
- Push to central repository
- Often need to first pull from the central repo to fetch and merge changes from others

Example:
a new conversion utility

```
cd $HOME/kent/src/hg/utils
mkdir fastaToMoney
cd fastaToMoney
#Create .h, .c, makefile with vi
git diff
git add fastaToMoney.h
git add fastaToMoney.c
git add makefile
git status; git pull; git push
```

Working on a new feature with a local branch

```
cd $HOME/kent
git branch #show which is active
git checkout -b blueMusic
# edit several files
git add inc/common.h
git add lib/common.c
git add lib/blueMusic.c
git commit -m 'good start on
blueMusic'
```

Working on a new feature with a local branch (cont)

```
# For branches that take a while to finish
# continue merging in others changes
# and testing as needed.
git checkout master
git pull # get other peoples changes
git checkout blueMusic
git diff master
git merge master
# repeat: work, test, merge
```

Working on a new feature with a local branch (cont)

```
# Ready to push final changes
git checkout master
git diff blueMusic
git merge blueMusic
git pull
git push
# When we are SURE we do not need it
git -d blueMusic
```

Cool Git Facts

- Content-addressable storage
- SHA1-hash ids are globally unique
  fc6c45e91d08a5f31899db872c07bc88a53299a0
- Git repo is a DAG of commits
- Git stores files, not diffs
- Git gc/pack compresses history
- A merge creates a commit with two or more parents
- Besides central-repo model, git supports hierarchical, peer-to-peer, etc.

Thanks to the git team

- Galt Barber
- Mark Diekhans
- Pauline Fujita
- Donna Karolchick
- Brian Raney
- Vanessa Swing
- Ann Zweig