

# Make Incremental Changes

Version Control with git and GitHub

# "FINAL".doc



FINAL.doc!



FINAL\_rev.2.doc



FINAL\_rev.6.COMMENTS.doc



FINAL\_rev.8.comments5.  
CORRECTIONS.doc



FINAL\_rev.18.comments7.  
corrections9.MORE.30.doc



FINAL\_rev.22.comments49.  
corrections.10.#@\$%WHYDID  
ICOMETOGRADSCHOOL????.doc

JORGE CHAN © 2012

WWW.PHDCOMICS.COM

<http://www.phdcomics.com/comics/archive.php?comid=1531>

# Methods for tracking versions

- ▶ Don't keep track
- ▶ Save numbered files
- ▶ Formal version control

# Suppose it stops working...

- ▶ Don't keep track
  - good luck!
- ▶ Save numbered files
  - Manually compare files (with `diff`)
- ▶ Formal version control
  - Easy to study changes back in time
  - Easy to jump back and test



**Karen Cranston**

@kcranstn



Following

@mtholder motivating git: You mostly collaborate with yourself, and me-from-two-months-ago never responds to email.

@swcarpentry

[http://bit.ly/kcranstn\\_tweet](http://bit.ly/kcranstn_tweet)

# Why use formal version control?

- ▶ Backup files that are changing
- ▶ Store a history of those changes
- ▶ Explore that history
- ▶ No worries about breaking things that work
- ▶ Merge changes from multiple people

# Example repository



kbroman / **PyBroman**

Unwatch ▾ 1

Various bits of code, while I learn the language — Edit

19 commits

1 branch

0 releases

1 contributor



branch: master ▾

**PyBroman** / +



Add \*.pyc to .gitignore



kbroman authored on Apr 23

latest commit 3b88e8177b



ParseGenotypeData

Add \*.pyc to .gitignore

4 months ago



License.txt

Initial commit

9 months ago



ReadMe.md

[Initial commit](#)

9 months ago



features.py

features.py: regex

9 months ago



**ReadMe.md**

## PyBroman







Various bits of Python code, while I learn the language.

Licensed under the [MIT license](#). ([More information](#))












# Example history

branch: **master** ▾
















Commits on Apr 23, 2014

	<b>Add *.pyc to .gitignore</b> kbroman authored on Apr 23	 3b88e81	
	<b>Improve convert?.py by not making copy of full data for individuals i...</b> ... kbroman authored on Apr 23	 fed56b6	

Commits on Apr 21, 2014

	<b>ParseGenotypeData/convert?.py: replace "" in function descriptions wi...</b> ... kbroman authored on Apr 21	 7a67efc	
	<b>Simplify ParseGentypeData/convert?.py re.sub() -&gt; string.replace()</b> kbroman authored on Apr 21	 6c7d0ad	
	<b>convert.py -&gt; convert3.py; add descriptions to all python functions</b> kbroman authored on Apr 21	 6d6ce3e	
	<b>ParseGenotypeData/ReadMe: fix typo</b> kbroman authored on Apr 21	 4a9052c	

Commits on Apr 19, 2014

	<b>Fix link in ParseGenotypeData/ReadMe.md</b> kbroman authored on Apr 19	 a7c42b1	
	<b>python 2 version of ParseGenotypeData/convert.py</b> kbroman authored on Apr 19	 aaac5a0	
	<b>convert.py: use "with open()" as f:" when opening files</b> kbroman authored on Apr 19	 4422f52	
	<b>convert.py needs python3</b> kbroman authored on Apr 19	 d0a71bb	
	<b>Add links to files in ParseGenotypeData/ReadMe.md</b> kbroman authored on Apr 19	 1c87015	



# Example commit

Improve convert?.py by not making copy of full data for individuals i...

[Browse code](#)

\_n family

master



kbroman authored on Apr 23

1 parent 7a67efc commit fed56b664cb2ac6f1beb9559f16109d13e6dda844

Showing 2 changed files with 10 additions and 8 deletions.

[Show diff stats](#)

9 ParseGenotypeData/convert2.py

[View](#)

```
@@ -61,8 +61,8 @@ def get_families (people):
61 61     return set([people[key].family for key in people])
62 62
63 63     def get_family_members (people, family):
64 - "Return a vector of subjects within a family."
65 - return [people[key] for key in people if people[key].family == family]
64 + "Return a vector of famids for subjects within a family."
65 + return [key for key in people if people[key].family == family]
66 66
67 67     def writeln (file, line, end="\n"):
68 68         "Write a single line to a file."
@@ -81,10 +81,11 @@ def write_genfile (filename, people, markers):
81 81
82 82         for family in families:
83 83             writeln(file, family)
84 - members = sorted(get_family_members(people, family), key=lambda person: int(person.id))
84 + members = sorted(get_family_members(people, family), key=lambda famid: int(people[famid].id))
85 85             writeln(file, len(members))
86 86
87 - for person in members:
87 + for famid in members:
88 +     person = people[famid]
88 89             writeln(file, "%s %s %s %s" % (person.id, person.mom, person.dad, person.sex))
89 90
90 91         for marker in markers:
```