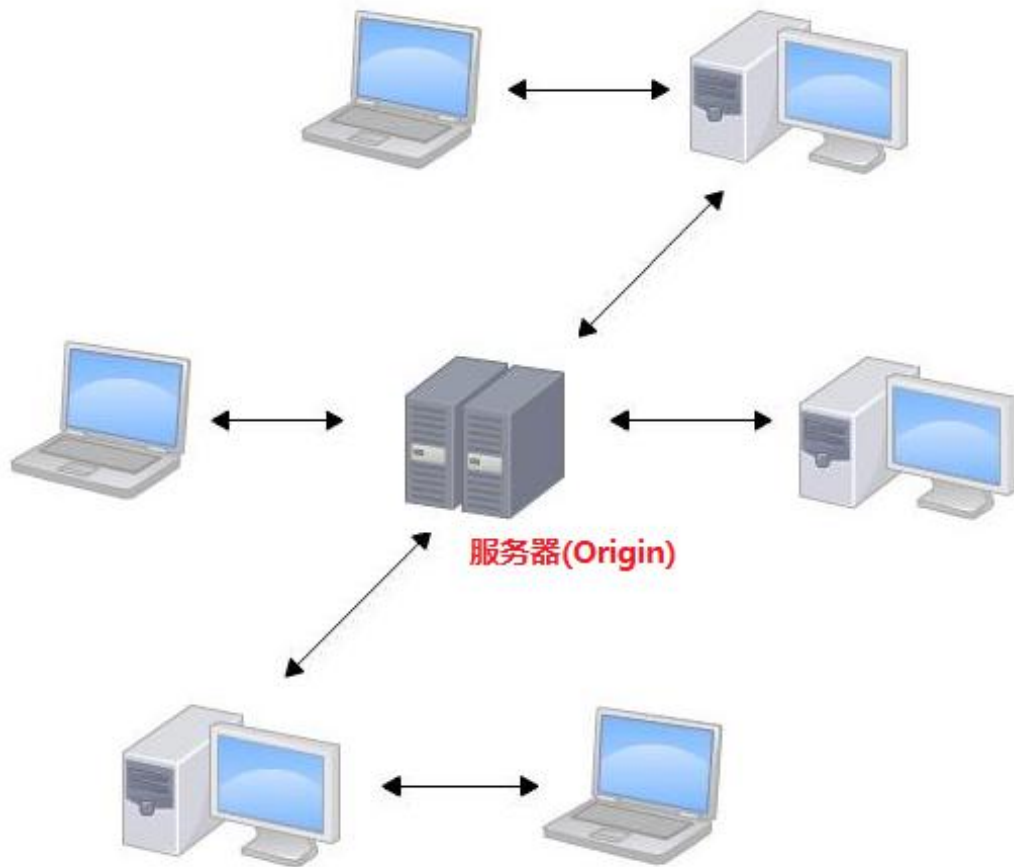


Git使用培训

2016-12-20



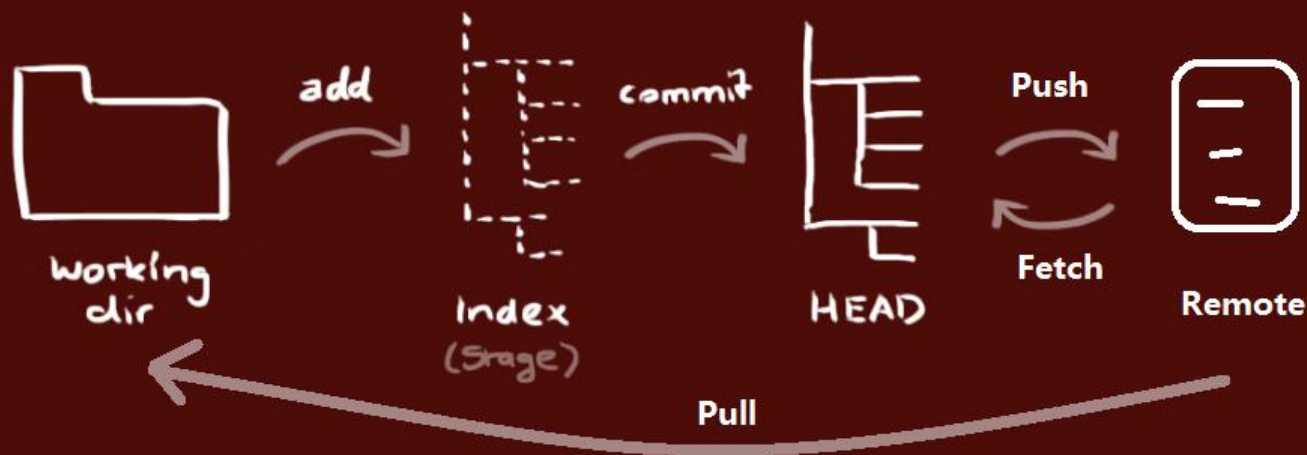
分布式版本控制系统



每个人的电脑上都是一个完整的版本库

工作流

你的本地仓库由 git 维护的三棵“树”组成。第一个是你的 **工作目录**，它持有实际文件；第二个是 **暂存区 (Index)**，它像个缓存区域，临时保存你的改动；最后是 **HEAD**，它指向你最后一次提交的结果。



git配置

自报家门：你的名字和Email地址

```
$ git config --global user.name "Your Name"  
$ git config --global user.email "email@example.com"
```

创建新仓库

创建新文件夹，打开，然后执行

```
git init
```

以创建新的 git 仓库。

检出仓库



执行如下命令以创建一个本地仓库的克隆版本：

```
git clone /path/to/repository
```

如果是远端服务器上的仓库，你的命令会是这个样子：

```
git clone username@host:/path/to/repository
```

Sanger: biocluster远程库

			
		Remotes	Advanced
Remote repository paths			
Name		Path	
origin		git@git.majorbio.com:sanger_bioinfo	

添加和提交

你可以提出更改（把它们添加到暂存区），使用如下命令：

```
git add <filename>
```

```
git add *
```

这是 git 基本工作流程的第一步；使用如下命令以实际提交改动：

```
git commit -m "代码提交信息"
```

现在，你的改动已经提交到了 **HEAD**，但是还没到你的远端仓库。

Git使用

git commit 提交

```
Present-tense summary under 50 characters
```

- * More information about commit (under 72 characters).
- * More information about commit (under 72 characters).

第一行是不超过50个字的提要，然后空一行，罗列出改动原因、主要变动、以及需要注意的问题。

推送改动

你的改动现在已经在本地仓库的 **HEAD** 中了。执行如下命令以将这些改动提交到远端仓库：

```
git push origin master
```

可以把 *master* 换成你想要推送的任何分支。

如果你还没有克隆现有仓库，并欲将你的仓库连接到某个远程服务器，你可以使用如下命令添加：

```
git remote add origin <server>
```

如此你就能够将你的改动推送到所添加的服务器上去了。

分支

分支是用来将特性开发绝缘开来的。在你创建仓库的时候，*master* 是“默认的”分支。在其他分支上进行开发，完成后再将它们合并到主分支上。



Git使用

创建一个叫做“feature_x”的分支，并切换过去：

```
git checkout -b feature_x
```

切换回主分支：

```
git checkout master
```

再把新建的分支删掉：

```
git branch -d feature_x
```

除非你将分支推送到远端仓库，不然该分支就是 *不为他人所见的*：

```
git push origin <branch>
```

更新与合并

要更新你的本地仓库至最新改动，执行：

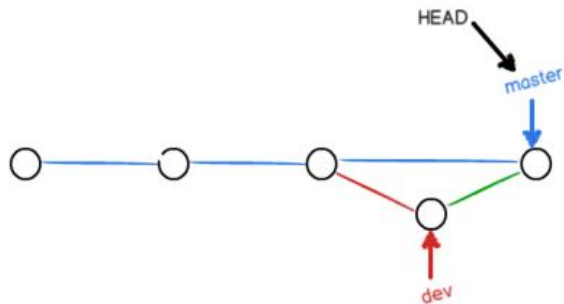
```
git pull
```

以在你的工作目录中 获取 (*fetch*) 并 合并 (*merge*) 远端的改动。

要合并其他分支到你的当前分支（例如 **master**），执行：

```
git merge <branch>
```

- 合并分支时，加上--no-ff参数就可以用普通模式合并，合并后的历史有分支，能看出来曾经做过合并，而fast forward合并就看起来曾经做过合并。



《==加上--no-ff之后的合并

解决冲突

```
<<<<<< HEAD
aaa
bbb
=====
master
aaa
>>>>>> master
```

在这两种情况下，**git** 都会尝试去自动合并改动。遗憾的是，这可能并非每次都成功，并可能出现**冲突**（*conflicts*）。这时候就需要你修改这些文件来手动合并这些**冲突**（*conflicts*）。改完之后，你需要执行如下命令以将它们标记为合并成功：

```
git add <filename>
```

在合并改动之前，你可以使用如下命令预览差异：

```
git diff <source_branch> <target_branch>
```


替换本地改动

假如你操作失误（当然，这最好永远不要发生），你可以使用如下命令替换掉本地改动：

```
git checkout -- <filename>
```

此命令会使用 **HEAD** 中的最新内容替换掉你的工作目录中的文件。已添加到暂存区的改动以及新文件都不会受到影响。

假如你想丢弃你在本地的所有改动与提交，可以到服务器上获取最新的版本历史，并将你本地主分支指向它：

```
git fetch origin
```

或 `git checkout .`

```
git reset --hard origin/master
```

时光机 - 撤销修改

场景1：当你改乱了工作区某个文件的内容，想直接丢弃工作区的修改时，用命令`git checkout -- file`

场景2：当你不但改乱了工作区某个文件的内容，还添加到了暂存区时，想丢弃修改，分两步，第一步用命令`git reset HEAD file`，就回到了场景1，第二步按场景1操作。

场景3：已经提交了不合适的修改到版本库时，想要撤销本次提交，参考[版本回退](#)一节，不过前提是没有推送到远程库。

Git使用

时光机 - 版本回退

`git reset --hard savepoint`

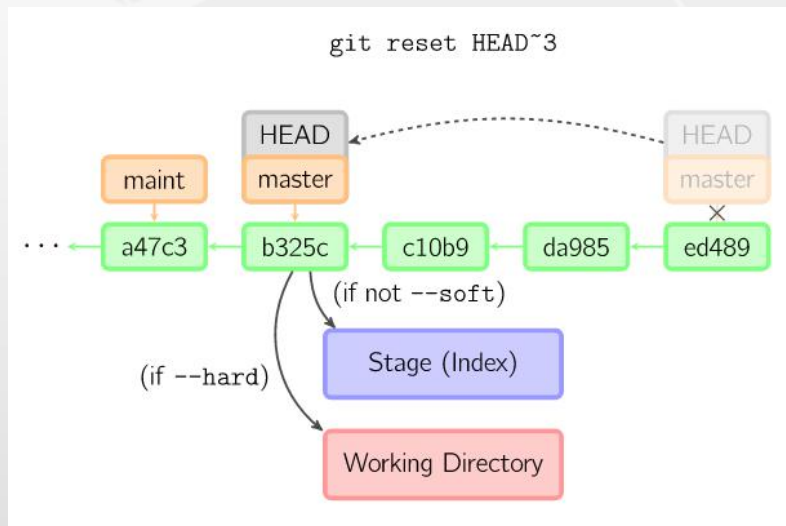
BRANCHES AS SAVEPOINTS

Creating a branch is like saving your game before you battle the boss

`git reset --hard 1234abcd`

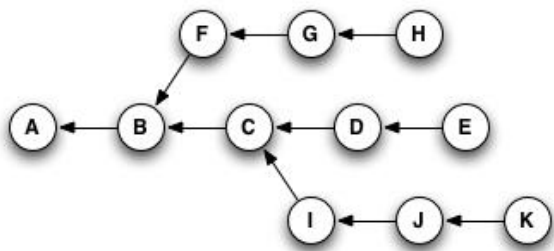
从最近的一个commit恢复

- 穿梭前，用`git log`可以查看提交历史，以便确定要回退到哪个版本。
- 要重返未来，用`git reflog`查看命令历史，以便确定要回到未来的哪个版本。

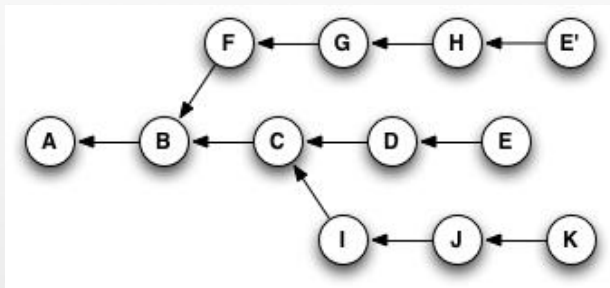


Git使用

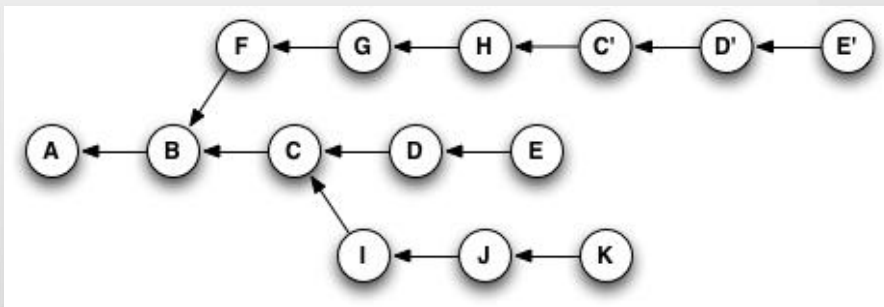
捡樱桃



git cherry-pick E

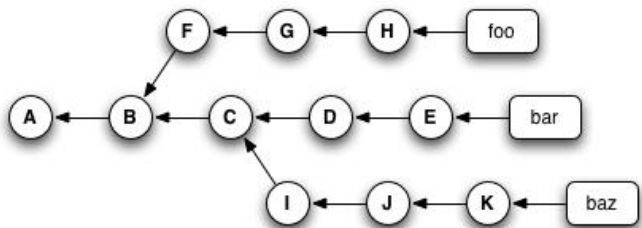


git cherry-pick C D E

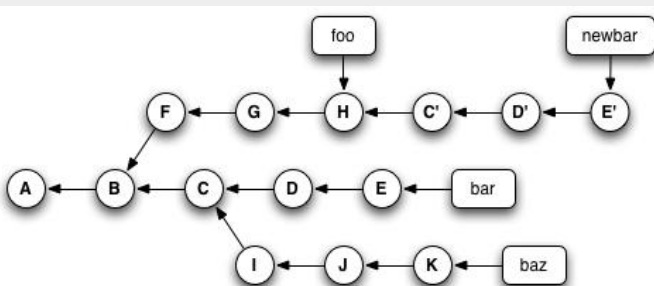


Git使用

改写历史

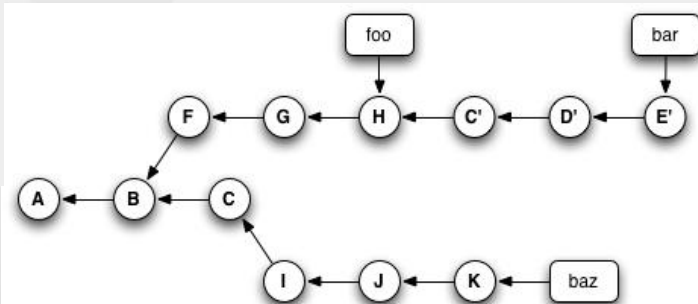


```
git checkout foo  
git checkout -b newbar  
git cherry-pick C D E
```



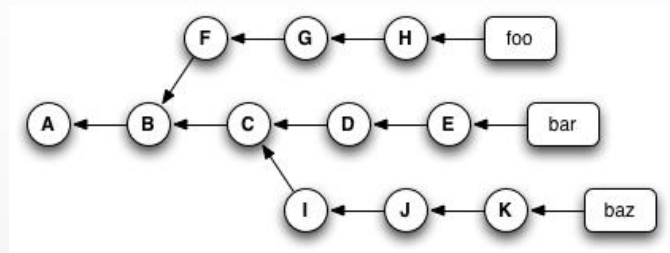
git rebase foo bar

```
git checkout bar  
git reset --hard newbar  
git branch -d newbar
```



Git使用

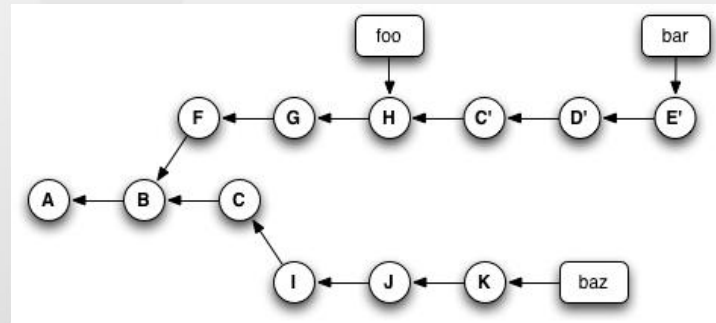
改写历史



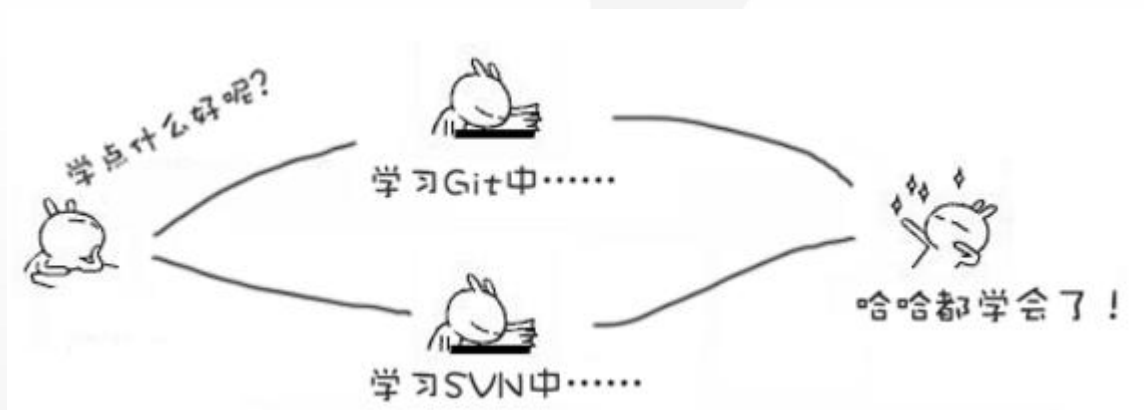
git rebase foo bar

后悔了....

git rebase --abort



Git分支



- ✓ 无论创建、切换和删除分支，**Git**在**1秒钟**之内就能完成！
- ✓ 无论你的版本库是**1个文件**还是**1万个文件**。

40 bytes

切换分支时储藏工作目录

需要切换分支，但当前工作目录下有没处理好的东西，不想提交也不想带到切换的分支里去，怎么办？

git stash 储藏你的工作

要查看现有的储藏，使用 **git stash list**

```
$ git stash list
stash@{0}: WIP on master: 049d078 added the index file
stash@{1}: WIP on master: c264051 Revert "added file_size"
stash@{2}: WIP on master: 21d80a5 added number to log
```

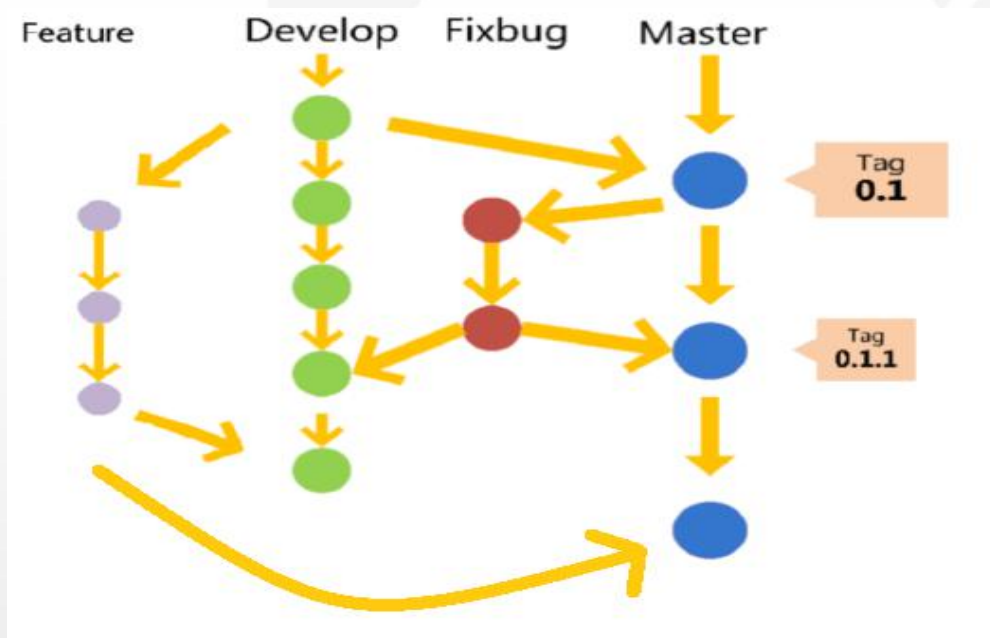
重新应用实施过的储藏，**git stash apply**

取出旧的储藏，**git stash apply stash@{2}**

重新应用被暂存的变更，**git stash apply --index**

正式版与开发版分支

GIT



Copy/Clone



服务器

devcluster

biocluster

web

tsanger

sanger

桑格代码提交更新

- 正式版：origin/master分支与正式服务器101上biocluster同步更新
 - 不允许在master分支上直接修改提交
 - 第一次克隆：`git clone biocluster:sanger_bioinfo`
- 测试分支：origin/dev用于102测试服务器上的开发测试
 - 获取开发分支（拉取并进入dev分支）：
`git branch -b dev origin/dev`
 - 切换到开发分支
`git checkout dev`
注意：通常情况下禁止将dev分支并入master分支

创建新功能分支或bug分支（newfeature/fixbug）：

- 开发新的pipe或tool或框架功能cluster：
 - `git checkout -b pipe-meta dev` # 从dev创建pipe-meta分支并切换到pipe-meta
 - `git checkout -b tool-fastatrim dev` # 单独开发一个tool
 - `git checkout -b cluster-xxx dev` # 开发框架新功能
 - `git checkout -b fixbug-0.1 master` # 由master创建一个bug分支
- 多人开发同一个功能时，将新功能分支推到远程新分支便于与他人同步共享：
 - `git pull` # 拉取更新
 - `git commit -m "提交更新信息"` # 提交更新，注意写清楚
 - `git push origin/newfeature newfeature` # 推到远程
- 开发完成一个新功能，将相关代码单独拷到102上对应地方进行测试，测试通过后，将新功能分支分别合并到origin/dev分支和origin/master
 - `git checkout dev` # 切换到dev或master分支
 - `git merge --no-ff newfeature` 将newfeature merge到dev或master
 - `git push origin/dev dev` # 推送到origin，拷到服务器进行测试
 - `git push origin/master` # 推到远程master，通知管理员更新集群biocluster

source tree

git-sanger-bioinfo

文件状态

工作副本

分支

标签

远程

origin

所有分支

☒ 显示远程分支

按日期排序

跳转到

图谱

描述

日期

作者

提交

master	Merge branch 'master' of xt:sanger_bioinfo	2016-05-20 15:14	xuting <arashi.xu	7e95985
origin/master	数据拆分 v0.20	2016-05-20 15:14	xuting <arashi.xu	79d551e
origin/dev	Merge branch 'master' of git.majorbio.com:sanger_bioinfo	2016-05-18 10:56	Yu Guo <jsb778@	6f24d2a
origin/HEAD	no message	2016-05-18 10:37	Yu Guo <jsb778@	3eb69c3
	1.进化树修剪bug修改, 2.pca和rda环境因子放大	2016-05-18 08:34	shenghe <sheng	0da557c
	taxassign bug fix	2016-05-10 14:26	Yu Guo <jsb778@	b2d901e
	Merge branch 'master' of git.majorbio.com:sanger_bioinfo	2016-05-06 16:15	shenghe <sheng	d2b28ee
	database部分bugs修改	2016-05-06 16:15	shenghe <sheng	56341fa
	Merge branch 'master' of git.majorbio.com:sanger_bioinfo	2016-05-06 16:14	qiuping <ping.qi	15667f8
	Merge branch 'master' of git.majorbio.com:sanger_bioinfo	2016-05-06 16:13	qiuping <ping.qi	26b1e5f
	alpha相关	2016-05-06 16:13	qindanhua <qdh	d1230dc
	差异模块最新代码	2016-05-06 16:13	qiuping <ping.qi	6b61eec

已依照文件状态排序

提交: 7e9598585f42e45d2e14c82b186a925ff4424b6b [7e95985]

父级: 79d551ea65, 6f24d2a253

作者: xuting <arashi.xu@majorbio.com>

日期: 2016年5月20日 15:14:55

标签: HEAD, -, master, origin/master, origin/dev, origin/HEAD

Merge branch 'master' of xt:sanger_bioinfo

bin/upload_result.py

src/mbio/api/database/anosim.py

src/mbio/api/database/beta_multi_analysis.py

src/mbio/api/database/distance.py

src/mbio/api/database/test/test...

文件状态

日志 / 历史

搜索

bin/upload_result.py

同该区块

```
79 79         os.dup2(so.fileno(), sys.stdout.fileno())
80 80         os.dup2(se.fileno(), sys.stderr.fileno())
81 81         self.log("开始监控需要上传的文件, pid: %s" % os.getpid())
82
83
84 +         tasks = self.get_upload_tasks()
85 +         try:
86 +             tasks = self.get_upload_tasks()
87 +         except Exception, e:
88 +             exstr = traceback.format_exc()
89 +             print exstr
90 +             self.log("查询上传任务出错: %s" % e)
91 +             self._db = Config().get_db()
92 +             tasks = self.get_upload_tasks()
93
94 if tasks:
95     for record in Asyncio
```

Git参考资料

- git - 简明指南
 - <http://rogerdudler.github.io/git-guide/index.zh.html>
- 常用 Git 命令清单
 - http://www.ruanyifeng.com/blog/2015/12/git-cheat-sheet.html?bsh_bid=5983510
 - <https://git-scm.com/book/>
- Git远程操作详解 http://www.ruanyifeng.com/blog/2014/06/git_remote.html
- <http://think-like-a-git.net/halp.html>