

Gestion de projets en bioinformatique

Pierre Poulain

pierre.poulain@univ-paris-diderot.fr

09/2011



À l'exception des illustrations et images dont les crédits sont indiqués à la fin du document et dont les droits appartiennent à leurs auteurs respectifs, le reste de ce cours est sous licence Creative Commons Paternité (CC-BY).

<http://creativecommons.org/licenses/by/2.0/fr/>

Menu

- 1 **Qu'est-ce qu'un projet ?**
- 2 Gestion de projet en bioinformatique
- 3 Outils pour le travail collaboratif
- 4 Comparaison de code source
- 5 Gestionnaires de versions
- 6 Documentation (automatique) de code

Définition

D'après Wikipédia : « Un projet est un **engagement** irréversible de résultat incertain, non reproductible a priori à l'identique, nécessitant le **concours** et l'**intégration** d'une grande diversité de **contribution**, et **répondant à un besoin** exprimé. »

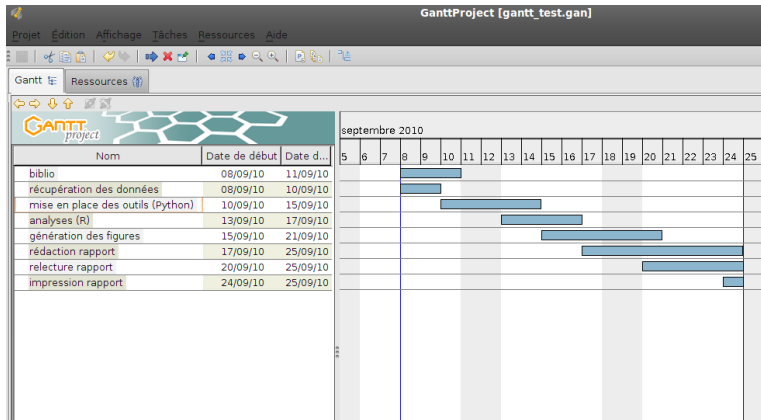
Nature d'un projet

Organisation d'un projet par la méthode **QQOQCCP**.

- **Q**uoi (Quelles actions ?)
- **Q**ui (Pour qui est le projet ? Qui est impliqué ?)
- **O**ù (Quel domaine est concerné ? Quel est le contexte ?)
- **Q**uand (Dans quel ordre fait-on les choses ?)
- **C**omment (Quels moyens ? Quelles méthodes)
- **C**ombien (Combien ça coûte ?)
- **P**ourquoi (Pourquoi ce projet ? Quels sont les objectifs ?)

Diagramme de Gantt

Modélisation de la planification de tâches nécessaires à la réalisation d'un projet (Henry L. Gantt, 1977).



GanttProject <http://www.ganttproject.biz/>

Menu

- 1 Qu'est-ce qu'un projet ?
- 2 Gestion de projet en bioinformatique**
- 3 Outils pour le travail collaboratif
- 4 Comparaison de code source
- 5 Gestionnaires de versions
- 6 Documentation (automatique) de code

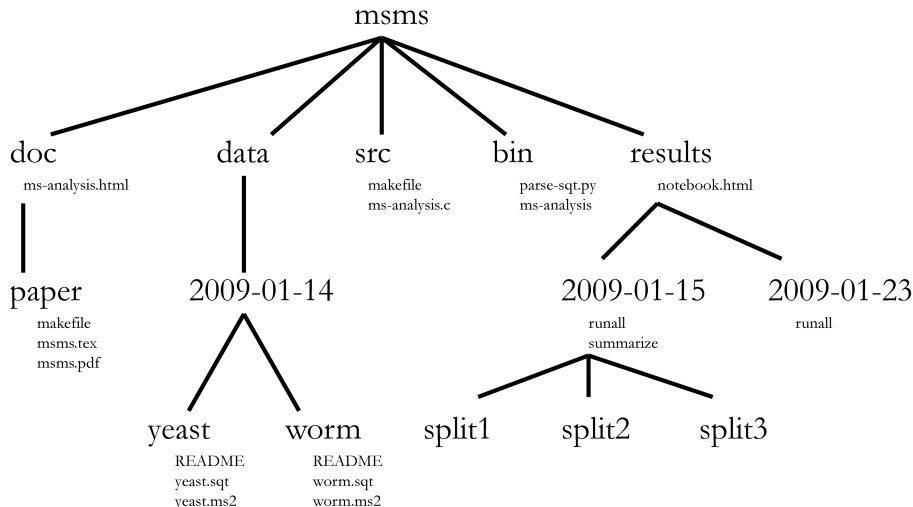
Gestion de projet en bioinformatique

A Quick Guide to Organizing Computational Biology Projects

W. S. Noble, *PLoS Computational Biology*, 5 : e1000424, 2009

1. structuration d'un projet en bioinformatique
2. reproductibilité des résultats
3. documentation

Organisation des fichiers et répertoires



Bien gérer un projet (bioinformatique)

séparer données primaires, résultats et analyses

script global pour régénérer simulations et analyses

gestion des erreurs

gestionnaires de versions

sauvegardes !

Bien gérer un projet (bioinformatique) 2

documentation générale – cahier de laboratoire

documentation ponctuelle (README)

documentation des scripts et programmes
(commentaires + docstring/doxygen)

Cahier de laboratoire

physique (papier) ou virtuel (notebook.html)

horodatage obligatoire

valeur légale

y consigner **toutes** les simulations et **tous** les résultats
(programmes utilisés et paramètres)

répertoires et noms de fichiers

~~perso~~ (laboratoire / entreprise)

→ refaire simulations et analyses (sans vous)

Menu

- 1 Qu'est-ce qu'un projet ?
- 2 Gestion de projet en bioinformatique
- 3 Outils pour le travail collaboratif**
- 4 Comparaison de code source
- 5 Gestionnaires de versions
- 6 Documentation (automatique) de code

Travail collaboratif

travailler à plusieurs sur un même projet

à des temps différents ou pas

à des endroits différents ou pas

outils pour discuter, échanger et comparer

Travail collaboratif

discuter : MSN, Gtalk, Facebook, Skype...

échanger : mail, clef USB, Dropbox...

se réunir : Doodle (<http://www.doodle.com/>)

EtherPad

édition collaborative simultanée en temps réel

ietherpad <http://ietherpad.com/>

SplinePad <http://pad.spline.de/>

PrimaryPad <http://primarypad.com/>

TypeWithMe <http://typewith.me/>

services similaires

Writeboard <http://writeboard.com/>

collabedit <http://collabedit.com/>

EtherPad

The screenshot displays the EtherPad web interface. At the top, a blue header bar contains the text "EtherPad". Below this, a navigation bar includes a "Public Pad" link, a "Pad Options" button, an "Import/Export" button, a "Saved revisions" button, and a "Time Slider" button. The main editing area features a toolbar with various text formatting options (bold, italic, underline, strikethrough, text color, background color, bulleted list, numbered list, indent, outdent, undo, redo, link, unlink, image) and a text input field. The code editor shows the following Python code:

```
1 # hello this is a sample code
2 # of the turtle Python module
3
4 # Pierre et Toto 02/08/2010
5
6 from turtle import *
7
8 angle = 5
9 distance = 15
10 pas = 0
11
12 while pas <= 50:
13     forward(distance)
14     left(angle)
15     compteur += 1
16     angle += 2
17
18 print "this is the end"
19
```

On the right side, there is a chat area with a text input field and a "Share this pad" button. Below the chat area, a "History" section shows a list of revisions:

August 2, 2010	
Pierre: Salut, je commence le projet "tortue"	10:16
Toto: OK, j'apporte quelques modifications	10:20
Pierre: Merci pour tes modif	10:24

Adresse « publique » du type

<http://ietherpad.com/N3rtMQIpK>

Menu

- 1 Qu'est-ce qu'un projet ?
- 2 Gestion de projet en bioinformatique
- 3 Outils pour le travail collaboratif
- 4 Comparaison de code source**
- 5 Gestionnaires de versions
- 6 Documentation (automatique) de code

xxdiff

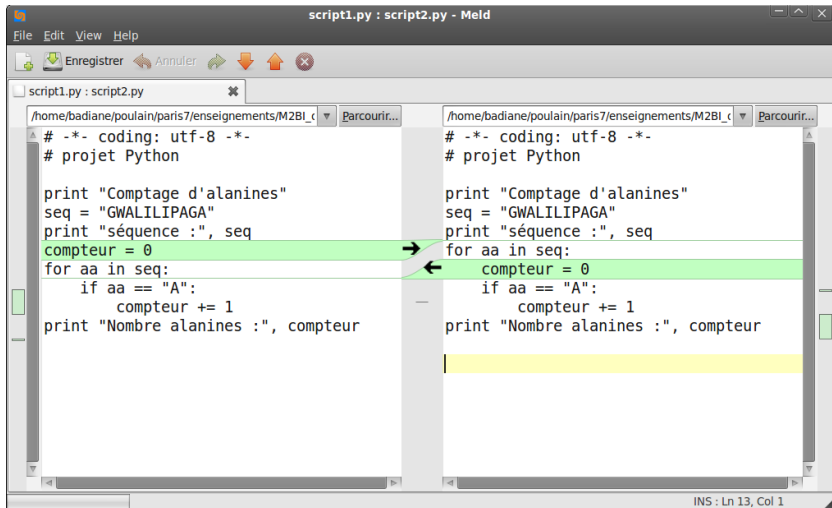
```
xxdiff script1.py script2.py
```

```
script1.py <-> script2.py
```

script1.py	7	script2.py	6	1
# -*- coding: utf-8 -*-		# -*- coding: utf-8 -*-		
# projet Python		# projet Python		
print "Comptage d'alanines"		print "Comptage d'alanines"		
seq = "GWALILIPAGA"		seq = "GWALILIPAGA"		
print "sÃ©quence :", seq		print "sÃ©quence :", seq		
compteur = 0		compteur = 0		
for aa in seq:		for aa in seq:		
if aa == "A":		if aa == "A":		
compteur += 1		compteur += 1		
print "Nombre alanines :", compteur		print "Nombre alanines :", compteur		

Meld

`meld script1.py script2.py`



Menu

- 1 Qu'est-ce qu'un projet ?
- 2 Gestion de projet en bioinformatique
- 3 Outils pour le travail collaboratif
- 4 Comparaison de code source
- 5 Gestionnaires de versions**
- 6 Documentation (automatique) de code

Pourquoi ?

```
login@host> cd test_projet_en_bazar/
```

```
login@host> ls
```

00README	script (8e copie).py	script_OLD.py
script (12e copie).py	script (autre copie).py	script.py
script_14042010.py	script (copie).py	script_ver1.py
script-22032010.py	script_LAST.py	script_ver2.py
script (7e copie).py	script_NEW.py	script_ver3_03052010.py

Oui mais !

```
login@host> ls -l
total 60
-rw-r--r-- 1 poulain dsimb 38 2010-07-22 10:47 00README
-rw-r--r-- 1 poulain dsimb 153 2010-07-22 10:45 script (12e copie).py
-rw-r--r-- 1 poulain dsimb 75 2010-04-14 09:17 script_14042010.py
-rw-r--r-- 1 poulain dsimb 51 2010-03-22 15:49 script-22032010.py
-rw-r--r-- 1 poulain dsimb 124 2010-07-22 10:40 script (7e copie).py
-rw-r--r-- 1 poulain dsimb 135 2010-07-22 11:12 script (8e copie).py
-rw-r--r-- 1 poulain dsimb 95 2010-07-22 11:31 script (autre copie).py
-rw-r--r-- 1 poulain dsimb 108 2010-07-22 12:05 script (copie).py
-rw-r--r-- 1 poulain dsimb 153 2010-06-27 10:35 script_LAST.py
-rw-r--r-- 1 poulain dsimb 176 2010-07-22 17:30 script_NEW.py
-rw-r--r-- 1 poulain dsimb 69 2010-03-29 11:56 script_OLD.py
-rw-r--r-- 1 poulain dsimb 27 2010-02-02 17:03 script.py
-rw-r--r-- 1 poulain dsimb 131 2010-07-22 18:05 script_ver1.py
-rw-r--r-- 1 poulain dsimb 167 2010-07-22 17:50 script_ver2.py
-rw-r--r-- 1 poulain dsimb 80 2010-05-03 14:30 script_ver3_03052010.py
```

La solution

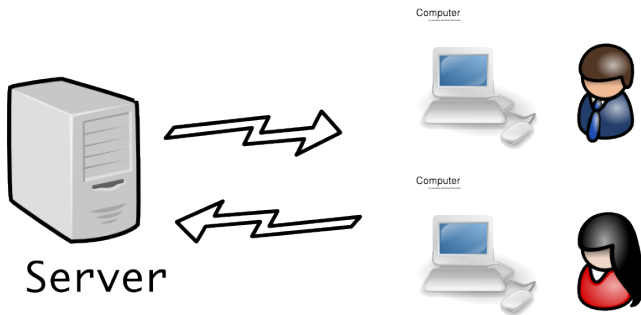
les gestionnaires de versions

archiver les différentes modifications apportées à des données textuelles

`.py .c .cpp .pl .java .html .xhtml .tex .txt`

Gestionnaires de versions

Version Control System (VCS)



Deux principes :

- *Centralized Version Control System (CVCS)*
- *Distributed Version Control System (DVCS)*

Gestionnaires de versions (2)

Centralized Version Control System (CVCS)


- *Concurrent Versions System (CVS)*
- *subversion (svn)*

Distributed Version Control System (DVCS)



Ranger le bazar avec Bazaar

<http://bazaar.canonical.com/>

- né en 2005
- supporté par Canonical (Ubuntu)
- robuste
- simple d'utilisation
- développé en Python
- adossé à la plateforme de développement  launchpad

<https://launchpad.net/>

Bazaar in five minutes <http://doc.bazaar.canonical.com/bzr.dev/en/mini-tutorial/index.html>

Initialisation de la branche

```
login@host> mkdir projet
```

```
login@host> cd projet
```

```
login@host> bzzr init
```

```
Created a standalone tree (format: 2a)
```

```
login@host> ls -a
```

```
./  ../  .bzzr/
```

```
# définition identité
```

```
login@host> bzzr whoami "Prénom Nom <adresse.mail@provider>"
```

```
# vérification
```

```
login@host> bzzr whoami
```

```
Prénom Nom <adresse.mail@provider>
```

Liste des commandes

```
login@host> bzz help  
Bazaar 2.1.1 -- a free distributed version-control tool  
http://bazaar-vcs.org/
```

Basic commands:

bzz init	makes this directory a versioned branch
bzz branch	make a copy of another branch
bzz add	make files or directories versioned
bzz ignore	ignore a file or pattern
bzz mv	move or rename a versioned file
bzz status	summarize changes in working copy
bzz diff	show detailed diffs
bzz merge	pull in changes from another branch
bzz commit	save some or all changes
bzz send	send changes via email
bzz log	show history of changes
bzz check	validate storage
bzz help init	more help on e.g. init command
bzz help commands	list all commands
bzz help topics	list all help topics

Ajout de fichier et statut

```
login@host> vi script.py
```

```
# -*- coding: utf-8 -*-  
# projet Python  
  
print "début du projet"  
seq = "GWALILIPAGA"  
print "séquence :", seq
```

script.py

```
login@host> ls -a
```

```
./  ../  .bzz/  script.py
```

```
login@host> bzz status
```

```
unknown:
```

```
    script.py
```

Ajout de fichier et statut (2)

```
login@host> bzip add script.py
adding script.py
```

```
login@host> bzip status
added:
    script.py
```

script.py est maintenant « versionné »

```
login@host> bzip commit -m "ajout de script.py"
Committing to: /home/login/chemin/du/projet/
added script.py
Committed revision 1.
```

commit = enregistrement version et journal (*log*) de la branche
`bzip commit` sans option `-m` lance un éditeur de texte

Suivi des modifications

script.py (avant)

```
# -*- coding: utf-8 -*-  
# projet Python  
  
print "début du projet"  
seq = "GWALILIPAGA"  
print "séquence :", seq
```

script.py (après)

```
# -*- coding: utf-8 -*-  
# projet Python  
  
print "Comptage d'alanines"  
seq = "GWALILIPAGA"  
print "séquence :", seq  
compteur = 0  
for aa in seq:  
    if aa == "A":  
        compteur += 1  
print compteur
```

```
login@host> bzip status
```

```
modified:
```

```
script.py
```

```
login@host> bzip commit -m "comptage des alanines"
```

```
Committing to: /home/login/chemin/du/projet/
```

```
modified script.py
```

```
Committed revision 2.
```


Suivi des modifications (2)

script.py (avant)

```
# -*- coding: utf-8 -*-
# projet Python

print "Comptage d'alanines"
seq = "GWALILIPAGA"
print "séquence :", seq
compteur = 0
for aa in seq:
    if aa == "A":
        compteur += 1
print compteur
```

script.py (après)

```
# -*- coding: utf-8 -*-
# projet Python

print "Comptage d'alanines"
seq = "GWALILIPAGA"
print "séquence :", seq
compteur = 0
for aa in seq:
    if aa == "A":
        compteur += 1
print "Nombre alanines :", compteur
```

```
login@host> bzip status
```

```
modified:
```

```
script.py
```

```
login@host> bzip commit -m "amélioration affichage résultat"
```

```
Committing to: /home/login/chemin/du/projet/
```

```
modified script.py
```

```
Committed revision 3.
```

Journal des modifications

```
login@host> bzz log
```

```
-----  
revno: 3  
committer: Prénom Nom <adresse.mail@provider>  
branch nick: projet  
timestamp: Sat 2010-09-11 18:26:32 +0200  
message:  
    amélioration affichage résultat  
-----
```

```
revno: 2  
committer: Prénom Nom <adresse.mail@provider>  
branch nick: projet  
timestamp: Sat 2010-09-11 18:15:47 +0200  
message:  
    comptage des alanines  
-----
```

```
revno: 1  
committer: Prénom Nom <adresse.mail@provider>  
branch nick: projet  
timestamp: Sat 2010-09-11 18:00:45 +0200  
message:  
    ajout de script.py
```

Rappel des modifications

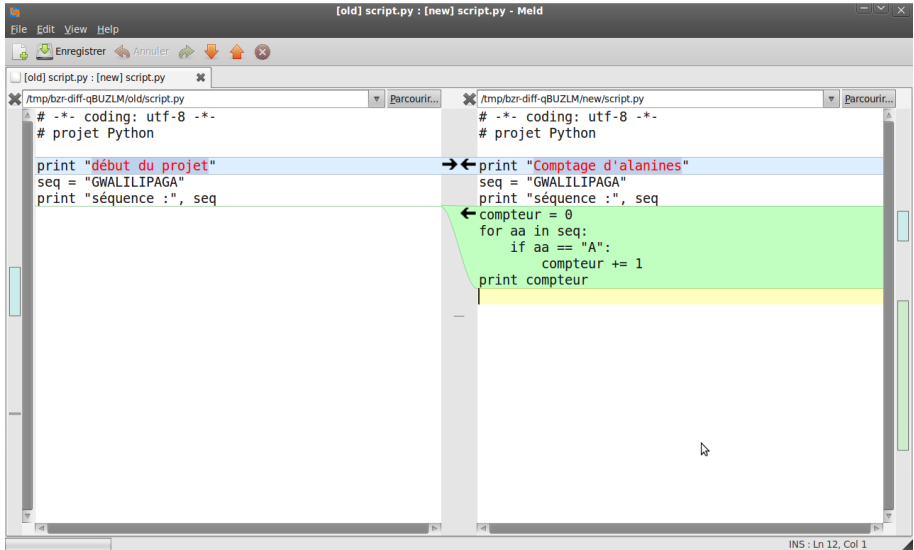
```
login@host> bzip diff -r1..2
== modified file 'script.py'
--- script.py      2010-09-11 16:00:45 +0000
+++ script.py      2010-09-11 16:15:47 +0000
@@ -1,7 +1,12 @@
# -*- coding: utf-8 -*-
# projet Python

-print "début du projet"
+print "Comptage d'alanines"
+  seq = "GWALILIPAGA"
+  print "séquence :", seq
+compteur = 0
+for aa in seq:
+    if aa == "A":
+        compteur += 1
+print compteur

bzip diff -r..2 / bzip diff -r2..
```

Rappel des modifications (bonus)

```
login@host> bzip diff -r1..2 --using meld
```



The screenshot shows the Meld diff tool interface. The title bar reads "[old] script.py : [new] script.py - Meld". The menu bar includes File, Edit, View, and Help. Below the menu bar are buttons for "Enregistrer" (Save), "Annuler" (Cancel), and navigation arrows. The main area is split into two panes. The left pane, titled "/tmp/bzip-diff-qBUZLM/old/script.py", shows the original code:

```
# -*- coding: utf-8 -*-  
# projet Python  
  
print "début du projet"  
seq = "GWALILIPAGA"  
print "séquence :", seq
```

 The right pane, titled "/tmp/bzip-diff-qBUZLM/new/script.py", shows the modified code:

```
# -*- coding: utf-8 -*-  
# projet Python  
  
print "Comptage d'alanines"  
seq = "GWALILIPAGA"  
print "séquence :", seq  
compteur = 0  
for aa in seq:  
    if aa == "A":  
        compteur += 1  
print compteur
```

 A blue highlight is on the first line of the new code. A green highlight covers the loop section. A yellow highlight is at the bottom. Arrows indicate line correspondences between the two panes. The status bar at the bottom right shows "INS : Ln 12, Col 1".

Annulation de modifications

script.py (avant)

```
# -*- coding: utf-8 -*-
# projet Python

print "Comptage d'alanines"
seq = "GWALILIPAGA"
print "séquence :", seq
compteur = 0
for aa in seq:
    if aa == "A":
        compteur += 1
print "Nombre alanines :", compteur
```

script.py (après)

```
# -*- coding: utf-8 -*-
# projet Python

print "Comptage d'alanines"
seq = "GWALILIPAGA"
print "séquence :", seq
for aa in seq:
    compteur = 0
    if aa == "A":
        compteur += 1
print "Nombre alanines :", compteur
```

```
login@host> python script.py
Comptage d'alanines
séquence : GWALILIPAGA
Nombre alanines : 1
```

oups !

Annulation de modifications (2)

```
login@host> bzip2 revert  
M script.py
```

```
login@host> cat script.py  
# -*- coding: utf-8 -*-  
# projet Python
```

```
print "Comptage d'alanines"  
seq = "GWALILIPAGA"  
print "séquence :", seq  
compteur = 0  
for aa in seq:  
    if aa == "A":  
        compteur += 1  
print "Nombre alanines :", compteur
```

ouf ! Retour à l'état du dernier *commit*.

Annulation de modifications (3)

script.py (avant)

```
# -*- coding: utf-8 -*-  
# projet Python  
  
print "Comptage d'alanines"  
seq = "GWALILIPAGA"  
print "séquence :", seq  
compteur = 0  
for aa in seq:  
    if aa == "A":  
        compteur += 1  
print "Nombre alanines :", compteur
```

script.py (après)

```
# -*- coding: utf-8 -*-  
# projet Python  
  
print "Comptage d'alanines"  
seq = "GWALILIPAGA"  
print "séquence :", seq  
for aa in seq:  
    compteur = 0  
    if aa == "A":  
        compteur = 1  
print "Nombre alanines :", compteur
```

```
login@host> bzip2 commit -m "meilleur compteur"  
Committing to: /home/login/chemin/du/projet/  
modified script.py  
Committed revision 4.
```

```
login@host> python script.py  
Comptage d'alanines  
séquence : GWALILIPAGA  
Nombre alanines : 1
```

re-oups !

Annulation de modifications (4)

```
login@host> bzip2 revert script.py -r3
M script.py
```

```
login@host> cat script.py
# -*- coding: utf-8 -*-
# projet Python
```

```
print "Comptage d'alanines"
seq = "GWALILIPAGA"
print "séquence :", seq
compteur = 0
for aa in seq:
    if aa == "A":
        compteur += 1
print "Nombre alanines :", compteur
```

```
login@host> bzip2 status
modified:
  script.py
login@host> bzip2 commit -m "script.py - retour version 3"
Committing to: /home/login/chemin/du/projet/
modified script.py
Committed revision 5.
```

re-ouf !

Annulation de modifications (5)

Si *commit* par erreur :

```
bzr uncommit
```

Pour enlever un fichier du gestionnaire de versions :

- en le gardant sur le disque

```
bzr remove --keep script.py
```

- en le détruisant aussi sur le disque

```
bzr remove --force script.py
```

Copie de branche

```
login@host> cd ..
```

```
login@host> bzip branch projet/ projet2  
Branched 5 revision(s).
```

```
login@host> cd projet2
```

```
login@host> ls  
script.py
```

Travail sur la nouvelle branche (projet2)

script.py (avant)

```
# -*- coding: utf-8 -*-  
# projet Python  
  
print "Comptage d'alanines"  
seq = "GWALILIPAGA"  
print "séquence :", seq  
compteur = 0  
for aa in seq:  
    if aa == "A":  
        compteur += 1  
print "Nombre alanines :", compteur
```

script.py (après)

```
# -*- coding: utf-8 -*-  
# projet Python  
  
print "Comptage d'alanines"  
seq = "GWALILIPAGA"  
print "séquence :", seq  
compteur = seq.count("A")  
print "Nombre alanines :", compteur
```

```
login@host> bzr commit -m "count() au lieu de for"  
Committing to: /home/login/chemin/du/projet2/  
modified script.py  
Committed revision 6.
```

Historique de la branche projet2

```
login@host> bzz log
```

```
-----  
revno: 6  
committer: Prénom Nom <adresse.mail@provider>  
branch nick: projet2  
timestamp: Mon 2010-09-13 20:51:32 +0200  
message:  
    count() au lieu de for  
-----
```

```
revno: 5  
committer: Prénom Nom <adresse.mail@provider>  
branch nick: projet  
timestamp: Sun 2010-09-12 21:10:08 +0200  
message:  
    script.py - retour version 3  
-----
```

```
revno: 4  
committer: Prénom Nom <adresse.mail@provider>  
branch nick: projet  
timestamp: Sun 2010-09-12 18:44:15 +0200  
message:  
    meilleur compteur  
-----
```

Fusion de branches

```
login@host> cd ../projet
```

```
login@host> bzip merge ../projet2/
```

```
M script.py
```

```
All changes applied successfully.
```

```
----- script.py -----
```

```
# -*- coding: utf-8 -*-
```

```
# projet Python
```

```
print "Comptage d'alanines"
```

```
seq = "GWALILIPAGA"
```

```
print "séquence :", seq
```

```
compteur = seq.count("A")
```

```
print "Nombre alanines :", compteur
```

```
login@host> bzip commit -m "ajout amélioration de projet2"
```

```
Committing to: /home/login/chemin/du/projet/
```

```
modified script.py
```

```
Committed revision 6.
```

Gestion des conflits

```
_____ script.py (projet) _____  
# -*- coding: utf-8 -*-  
# projet Python  
  
print "Comptage d'alanines"  
seq = "GWALILIPAGA"  
print "séquence :", seq  
compteur = seq.count("A")  
print "Nb ala :", compteur
```

```
_____ script.py (projet2) _____  
# -*- coding: utf-8 -*-  
# projet Python  
  
print "Comptage d'alanines"  
seq = "GWALILIPAGA"  
print "séquence :", seq  
compteur = seq.count("A")  
print "alanines :", compteur
```

```
login@host> bzip merge ../projet2  
M script.py  
Text conflict in script.py  
1 conflicts encountered.
```

```
login@host> ls  
script.py script.py.BASE script.py.OTHER script.py.THIS
```

Gestion des conflits (2)

script.py (conflits)

```
# -*- coding: utf-8 -*-
# projet Python

print "Comptage d'alanines"
seq = "GWALILIPAGA"
print "séquence :", seq
compteur = seq.count("A")
<<<<<< TREE
print "Nb ala :", compteur
=====
print "alanines :", compteur
>>>>>> MERGE-SOURCE
```

script.py.BASE (ancêtre commun)

```
# -*- coding: utf-8 -*-
# projet Python

print "Comptage d'alanines"
seq = "GWALILIPAGA"
print "séquence :", seq
compteur = seq.count("A")
print "Nombre alanines :", compteur
```

script.py.OTHER (branche projet2)

```
# -*- coding: utf-8 -*-
# projet Python

print "Comptage d'alanines"
seq = "GWALILIPAGA"
print "séquence :", seq
compteur = seq.count("A")
print "alanines :", compteur
```

script.py.THIS (branche projet)

```
# -*- coding: utf-8 -*-
# projet Python

print "Comptage d'alanines"
seq = "GWALILIPAGA"
print "séquence :", seq
compteur = seq.count("A")
print "Nb ala :", compteur
```

Gestion des conflits (3)

On garde la version de projet :

```
login@host> cp script.py.THIS script.py
```

```
login@host> bzip resolve  
All conflicts resolved.
```

```
login@host> ls  
script.py
```

```
----- script.py -----  
# -*- coding: utf-8 -*-  
# projet Python  
  
print "Comptage d'alanines"  
seq = "GWALILIPAGA"  
print "séquence :", seq  
compteur = seq.count("A")  
print "Nb ala :", compteur
```

```
login@host> bzip commit -m "merge with projet2"  
Committing to: /home/login/chemin/du/projet/  
Committed revision 8.
```


Conclusion

commit-ez souvent !

explorez les (très très très nombreuses)
capacités de Bazaar

Menu

- 1 Qu'est-ce qu'un projet ?
- 2 Gestion de projet en bioinformatique
- 3 Outils pour le travail collaboratif
- 4 Comparaison de code source
- 5 Gestionnaires de versions
- 6 Documentation (automatique) de code**

Principe

Générer de la documentation

1. automatiquement

2. à partir du code source

pydoc – docstring

```
testmod.py
# -*- coding: utf-8 -*-
'''
docstring pour le module testmod.py
testmod est un module de test
les docstrings sont de simples chaînes de caractères
'''

class TestClasse():
    """docstring pour la classe TestClasse"""

    def testmethode(self):
        "docstring pour la méthode testmethode"

def testfonction():
    "docstring pour la fonction testfonction"
```

pydoc – docstring 2

```
login@host> python
Python 2.6.5 (r265:79063, Apr 16 2010, 13:57:41)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import testmod
>>> help(testmod)
```

Help on module testmod:

NAME

testmod

FILE

/chemin/du/module/testmod.py

DESCRIPTION

docstring pour le module testmod.py
testmod est un module de test
les docstrings sont de simples chaînes de caractères

CLASSES

TestClasse

```
class TestClasse
|   docstring pour la classe TestClasse
|
|   Methods defined here:
|
|   testmethode(self)
|       docstring pour la méthode testmethode
```

FUNCTIONS

testfonction()
 docstring pour la fonction testfonction

Doxygen

code source → documentation (variables, fonctions, classes)

C, C++, Java, Python, PHP...

HTML, LaTeX, RTF

Doxygen 2

paramètres via Doxyfile (complexe)

DoxyWizard

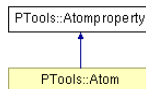
```
login@host> doxygen  
...  
login@host> firefox html/index.html
```


[Page principale](#)[Espaces de nommage](#)[Classes](#)[Fichiers](#)[Répertoires](#)[Liste des classes](#)[Index des classes](#)[Hiérarchie des classes](#)[Membres de classe](#)**PTools::Atomproperty**

Référence de la classe PTools::Atomproperty

```
#include <atom.h>
```

Grappe d'héritage de PTools::Atomproperty:



[Liste de tous les membres](#)

Fonctions membres publiques

	Atomproperty () default constructor
std::string	GetType () const return atom type (CA, CB, O, N...)
void	SetType (std::string newtype) define atom type (CA, CB, O, N...)
std::string	GetResidType () const return residue type (LEU, ARG...)

Doxygen 4

Coord3D PTools::Rigidbody::FindCenter () const

return geometric center of all atoms

Définition à la ligne **101** du fichier **rigidbody.cpp**.

Références **GetCoords()**, et **Size()**.

Référencé par **PTools::BaseAttractForceField::AddLigand()**, **CenterToOrigin()**, **Radius()**, **RadiusGyration()**, et **PTools::M**.

```
00102 {  
00103     Coord3D center(0.0,0.0,0.0);  
00104     for (uint i=0; i< this->Size() ; i++)  
00105     {  
00106         center = center + GetCoords(i);  
00107     }  
00108     return ( (1.0/(dbl)this->Size())*center);  
00109 }
```

Atomproperty const& PTools::Rigidbody::GetAtomProperty (uint pos) const [inline]

return atom properties

Définition à la ligne **90** du fichier **rigidbody.h**.

Références **mAtomProp**.

Référencé par **PTools::extractExtra()**.

```
00091 {  
00092     return mAtomProp[pos];  
00093 }
```

Conclusion générale

organisez-vous

temps, arborescence

documentez

cahier de laboratoire, code

investissez

bzr, outils

sauvegardez !

Références

A Quick Guide to Organizing Computational Biology Projects
W. S. Noble, PLoS Computational Biology, 5 : e1000424, 2009

<http://www.ploscompbiol.org/article/info%3Adoi%2F10.1371%2Fjournal.pcbi.1000424>

Présentation bazaar

<http://www.slideshare.net/giordano/bazaar-dvcs-for-human-beings-3121638>

Bazaar in five minutes

<http://doc.bazaar.canonical.com/bzr.dev/en/mini-tutorial/index.html>

Quick ref card

<http://doc.bazaar.canonical.com/latest/en/quick-reference/index.html>

bzr explorer

<http://doc.bazaar.canonical.com/explorer/en/visual-tour-gnome.html>

Doxygen

<http://www.stack.nl/~dimitri/doxygen/>