

## What is Git?

- Designed for non-linear development (easy branching and merging)
- Distributed development: local copy with the entire project history
- Optimized performance: speed and repository size

## Why should we move to Git?

- It is as easy to use as CVS/SVN for the standard user but offers much more for advanced users
- We should take the migration opportunity to CLEAN the main repository
  - only migrate "active" packages
  - organize package in a subtree structure (basic/analysis/detector/computing...)
  - separate software from other material (paper/doc...)
- The Git workflow perfectly fits the new Virgo development workflow proposed by Loic (virgoApp / virgoStagging / \$HOME)
- There is no tag in SVN

```
svn checkout https://svn.ego-gw.it/svn/advsw/myfavoritepackage/trunk myfavoritepackage
```



```
git clone https://ego-gw.it/git/myfavoritepackage.git
```

This command gives you an entire history of your project. After that, every operation you do is performed offline on your local disk (commits, diff, log...)

A centralized workflow is no longer required with Git. However we might keep this philosophy in Virgo.

## **Branching and merging**

- Fast and easy.
- You can easily create a new branch quickly, do a few commits on that branch and then either merge it into your mainline work or throw it away. You don't have to mess up the mainline
- Merging is trivial
- Create a separate branch to develop your analysis

# Git: basic commands

## Clone your package:

```
git clone https://ego-gw.it/git/myfavoritepackage.git
```

## Add/remove files

```
git add somefile.txt
git mv somefile.old somefile.new
git rm somefile.txt
```

## Commit

```
git commit -a
git commit somefile_1.txt somefile_2.txt
```

## Tags

```
git tag -a v0r0p0          # create a new tag
git checkout v0r0p0        # switch to a tag
```

## Branching

```
git branch NewBranch      # create a new branch
git checkout NewBranch    # switch to new branch
git branch -a             # list all branches
git merge NewBranch       # merge NewBranch to current branch
```

## Main repository

```
git pull                  # update your local repository
git push origin           # push your changes to the main repository
git push origin tag v0r0p0 # push a tag
```

## Commands you already know:

```
git log
git diff
git status
```

```
# clone your package (only the first time)
git clone https://ego-gw.it/git/myfavoritepackage.git

# make your changes

# commit your changes
git commit -a -m "my comments"

# push your changes to the master repository
git push origin
```