

Si hay dudas o contribuciones para ampliar esta guia, favor de
enviarlos a

avenzor@fis.cinvestav.mx
Jorge Venzor

***** INSTALACION de CLASS y MontePython *****
***** EN MAC OSX (Probado en Mountain Lion 10.08, Sierra 10.12)

La siguiente guia es complementaria con la documentacion oficial de
CLASS y MontePython, publicamente disponible:

<http://class-code.net/>

<http://montepython.net/>

Github:

https://github.com/lesgourg/class_public

https://github.com/audren/montepython_public

0. PRE-REQUISITOS:

La Terminal de Mac puede usarse muy parecido a Linux, pero antes debe
prepararse al sistema operativo.

1. Instalar Xcode desde App Store (es gratis)

2. Acepte los acuerdos de licencia

~]\$ sudo xcodebuild -license

3. Instalar la "Linea de Comandos":

En Xcode -> Preferencias -> Downloads -> instalar la casilla
"Command Line Tools".

4. Instale XQuartz a traves de un "dmg", [http://](http://xquartz.macosforge.org/landing/)

xquartz.macosforge.org/landing/

5. Instale MacPorts a traves de un "dmg" [http://www.macports.org/](http://www.macports.org/install.php)
[install.php](http://www.macports.org/install.php)

6. Una vez instalado Macports, si no tiene instalado gcc-mp-4.7, se
instala como sigue:

\$ sudo port selfupdate

Instalar

\$ sudo port install gcc47

Seleccionar como predeterminado:

\$ port select --list gcc

\$ sudo port select --set gcc mp-gcc47

MacPorts funciona similar a los habituales repositorios de Linux.

Por ejemplo: sudo port install programa

Desde MacPorts, se recomienda instalar los compiladores del proyecto
GNU.

+ Compilador de C version > 4.2. Se recomienda gcc-mp-4.7

- + Python, version ≥ 2.5 , pero la version 3.0 no es soportada por MontePython. La version recomendada es python2.7
- + Numpy
- + Scipy
- + Cython
- + Pyfits
- + gnuplot (recomendado para graficar)

Para revisar que se instalaron correctamente puede usar

```
~]$ gcc-mp-4.7 -v
```

```
~]$ python2.7
```

```
>>> import numpy
```

```
>>> import scipy
```

```
>>> import cython
```

```
>>> import pyfits
```

```
>>> exit()
```

```
~]$ gnuplot --version
```

Si no hay errores puede proceder con la instalacion de CLASS.

ADVERTENCIA: Si no estan presentes Numpy, Scipy, Cython o Pyfits entonces antes de instalarlas desde MacPorts, ejecuten la siguiente linea

```
~]$ sudo port select --set python python27
```

En la pagina de MacPorts debe buscar la version de las bibliotecas que corresponden a la version 2.7 de Python.

Otra opción para tener todas las paqueterías de python necesarias, es hacer un ambiente virtual, por ejemplo instalando miniconda de <https://conda.io/miniconda.html>:

Una vez instalado conda, el ambiente virtual se crea facilmente

Crear ambiente virtual de python 2.7 con nombre=Ejemplo :

```
$conda create --name Ejemplo python=2.7
```

Instalar pip en el ambiente virtual :

```
$conda install -n Ejemplo pip
```

Activar el ambiente virtual

```
$source activate Ejemplo
```

Instalar modelos en ambiente virtual con pip:

```
(Ejemplo)$pip install pyfits
```

```
(Ejemplo)$pip install cython
```

```
(Ejemplo)$pip install scipy
```

Documentación completa en <https://conda.io/docs/user-guide/tasks/manage-environments.html>

ADVERTENCIA: Es recomendable que el directorio de CLASS y el de MontePython esten separados.

Por ejemplo:

```
~/codigos/class
```

```
~/codigos/montepython
```

1. INSTALACION DE CLASS

Ir a <http://class-code.net/> y descargar la version mas reciente de class (v2.6.1)

Despues de descomprimir, edite el archivo class/Makefile para que utilice el compilador gcc-mp-4.7

Ir a la carpeta desde terminal y teclear
class]\$ make

Si no hay errores entonces se pueden hacer las primeras pruebas.

class]\$./class lcdm.ini

Los archivos de salida tienen extension "dat" y estan en class/output/

Edite lcdm.ini para pedir diferentes espectros: tCl, mPk, etc.

Los scripts de gnuplot adjuntos a esta guia pueden ser copiados a output/ para ayudar a graficar.

Nota: Para evitar problemas de compatibilidad con Montepython es recomendable instalar Class con gcc

2. INSTALACION DE MONTEPYTHON

En la pagina web <http://montepython.net/> se puede descargar

montepython_v1.2.4.tar.bz2 (se refiere a version 1.2).

Despues de descomprimir, dentro del directorio de montepython/ edite el archivo default.conf.

Este archivo le dice a MontePython donde puede encontrar el directorio class/

default.conf debe contener dos lineas parecidas a las siguientes:

root = 'ruta/hacia/codigos'

path['cosmo'] = root+'/class'

Como una primera prueba, desde terminal puede escribir

montepython]\$ python code/MontePython.py --help

Si no hay error, seguir adelante.

—> Hagamos la primera corrida de prueba de MontePython.

Por simplicidad, probemos solo con los datos simulados de Planck.

En el archivo example.param asegurese de que contenga la linea exacta:
data.experiments=['fake_planck_bluebook']

Ahora desde terminal ir a la carpeta de montepython y hacer la primera corrida

montepython]\$ montepython/MontePython.py run -p example.param

-o output -N 5

Significado: "-p" le dice cual es el archivo de parametros, "-o" le dice cual es el archivo de salida,

"-N" le dice cuantos puntos analizar.

En una corrida normal N sera del orden de 5000 o 10000, pero aqui elegimos 5 solo para probar.

Si hay errores que mencionen a class, puede verificar que la ruta a class este correcta en default.conf
Checar que el modulo classy.so se encuentre en /ruta/hacia/lib/python2.7/site-packages/

—> Ahora puede probar con diferentes parametros y conjuntos de datos (excepto wmap y planck).
Despues de la primera corrida se genera un archivo log.param dentro de output/,
a partir de entonces ya no es necesario incluir "-p example.param -conf default.conf"
en las siguientes corridas con los mismos parametros.
Adjunto a esta guia se puede utilizar el script runMP.sh, que ayuda a ejecutar varias veces la siguiente linea:
python ./code/MontePython.py -o output -N 50
Por ejemplo, ejecute:
montepython]\$ sh runMP.sh 3 5

Advertencia: Recuerde proveer una carpeta vacia para cada nuevo conjunto de parametros a usar.

—> Analizar las cadenas graficar.
Si se acumularon varias cadenas y fueron informativas, la visualizacion del resultado es bastante comodo en MontePython.
Obtener: estimacion de convergencia, mejor ajuste de parametros, intervalos de confianza, matriz de covarianza, plots de probabilidad posterior marginalizada.
MontePython produce todos estos resultados simplemente con el comando:
montepython]\$ python code/MontePython.py -info output

3. INSTALACION DEL LIKELIHOOD DE PLANCK

Si ya se descargo el likelihood de Planck en el pasado (para CosmoMC por ejemplo),
en MontePython hay opciones para simplemente darle las rutas.
En lo que sigue, suponemos que la descarga y compilacion del likelihood de Planck se hace por primera vez.

Ir al sitio de Planck en la ESA, <https://pla.esac.esa.int/pla/#home>
Seguir los enlaces Planck Legacy Archive Interoperability System —
> LIKELIHOOD
Crear una carpeta llamada "planck"
Descargar todos los archivos tar.gz dentro de ~/planck
Descomprimir todos los tar.gz

——> Ajustando rutas

**** Archivo de configuracion**

En el archivo montepython/default.conf añadir una linea como
path['clik'] = 'ruta/hacia/planck/plc-2.0/'
path['data'] = 'ruta/hacia/montepython/data/'
path['cosmo'] = 'ruta/hacia/class/'

NOTA: Si ya se habia instalado antes el likelihood de Planck, solo hay que ajustar las rutas anteriores. Recuerde que para instalar los datos Planck tiene que cumplir todos los prerequisites (todas las paqueterías necesarias de python).

——> Hacer el 'wrapper' (parche) de Planck

Ir al directorio ~/planck/plc-2.0

En Terminal, ejecutar el comando de ayuda del waf y leer las opciones disponibles

```
plc-2.0]$ ./waf configure --help
```

En mi maquina yo utilice:

```
plc-2.0]$ python waf configure --lapack_mkl=/opt/intel/mkl --  
lapack_mkl_version=10.3  
plc-2.0]$ python waf install
```

**** Lapack**

El waf contiene varias opciones para instalar lapack.

```
--lapack_install      Instala lapack de manera local  
--lapack_islocal      Le indica al waf que ya se instalo
```

lapack con --lapack_install

```
--lapack_mkl=LAPACK_MKL Le indica al waf donde esta la mkl de  
intel (frecuentemente /opt/intel/mkl)
```

**** Compilador**

Se le puede especificar al waf cuales compiladores usar

Elegir entre gcc e icc

Elegir entre gfortran e ifort

**** Arquitectura**

```
--m32
```

```
--m64
```

Una vez elegidas las opciones anteriores, se puede configurar, por ejemplo:

```
plc-2.0]$ ./waf configure --install_all_deps
```

Si no hay errores, entonces

```
plc-2.0]$ ./waf install
```

```
montepython/MontePython.py run -p example.param -o output -N 5
```

——> Correr MontePython con Planck

Ya que se instalo correctamente, debemos cargar las rutas a la linea

de comandos de la Terminal

```
plc-2.0]$ source /ruta/hacia/planck/plc-2.0/bin/  
clik_profile.sh
```

Nos movemos en Terminal al home de montepython.

Para probar la instalacion utilizamos el archivo base.param

Podemos editarlo para probar cada una de los 4 likelihoods por separado.

```
montepython]$ montepython/MontePython.py run -p base.param -o  
test -N 5
```

NOTA: asegurese de borrar la carpeta de prueba test/ cada vez que se corra el comando

```
python montepython/MontePython.py run -p base.param -o output -N 5  
python montepython/MontePython.py run -p input/lcdm.param -o chains/  
planck/lcdm -c covmat/base.covmat -N 5
```

```
*****  
*****
```

```
-----  
-----
```

Si hay dudas o contribuciones para ampliar esta guia, favor de enviarlos a

avenzor@fis.cinvestav.mx
Jorge Venzor

```
-----  
-----
```