

Josué De Santiago Sanabria



README

- De <https://cosmologist.info/cosmomc/readme.htm>:
 - CosmoMC es una herramienta de Markov-Chain Monte-Carlo (MCMC) escrita en Fortran 2008 para explorar el espacio de parámetros cosmológico, junto con código en Fortran y python para analizar muestras tipo Monte-Carlo así como importance sampling (mas una suite de scripts para construir redes de ejecuciones, graficar y presentar resultados).

README

- De <https://cosmologist.info/cosmomc/readme.htm>:
 - CosmoMC es una herramienta de **Markov-Chain Monte-Carlo** (MCMC) escrita en Fortran 2008 para explorar el espacio de parámetros cosmológico, junto con código en Fortran y python para analizar muestras tipo Monte-Carlo así como importance sampling (mas una suite de scripts para construir redes de ejecuciones, graficar y presentar resultados).

Monte Carlo

- Problema 1: Generar muestras $\{x^{(r)}\}_{r=1}^R$ con una cierta distribución de probabilidad $P(x)$
- Problema 2: Estimar valores de expectación de funciones bajo esta distribución

$$\Phi = \langle \phi(x) \rangle \equiv \int d^N x P(x) \phi(x)$$

- Solución 2: Simplemente calcula el estimador

$$\hat{\Phi} \equiv \frac{1}{R} \sum \phi(x^{(r)})$$

- Solución 1: No tan simple

¿Muestreo uniforme?

$$Z_R = \sum_{r=1}^R P^*(\mathbf{x}^{(r)}),$$

$$\Phi = \int d^N \mathbf{x} \phi(\mathbf{x}) P(\mathbf{x}) \text{ by}$$

$$\hat{\Phi} = \sum_{r=1}^R \phi(\mathbf{x}^{(r)}) \frac{P^*(\mathbf{x}^{(r)})}{Z_R}$$

- En cosmología \mathbf{x} es un vector de 6 componentes
- El cálculo toma 0.6s por cada punto
- 10 puntos por componente:
10⁶ evaluaciones=7 días
- 75% de los puntos están en el borde del espacio de parámetros

Metropolis Hastings Method

- Step 1: Pick a point to start the walk $(h_0, \Omega_{\Lambda 0})$
- Step 2: Pick a second point in the neigh

$$h_1 = h_0 + \Delta \cdot \text{GaussRand}(0,1)$$

- Step 3: Compute the likelihood of the new point and compare it to the previous point

$$\text{Ratio} = \log \left(\frac{\mathcal{L}(D|\Theta_1)}{\mathcal{L}(D|\Theta_0)} \right) = \chi^2(\Theta_0) - \chi^2(\Theta_1)$$

- if $\text{Ratio} > 0$ then accept the new point as the next step
- If $\text{Ratio} < 0$ then
 - if $\exp(\text{Ratio}) > \text{Rand}(0,1)$ then accept the point
 - if $\exp(\text{Ratio}) < \text{Rand}(0,1)$ then keep the old point as the new point

Metropolis Hastings Method

**Repeat this a
thousand times!**

- Step 1: Pick a point to start the walk (h_0, Ω_{Λ_0})
- Step 2: Pick a second point in the neigh

$$h_1 = h_0 + \Delta \cdot \text{GaussRand}(0,1)$$

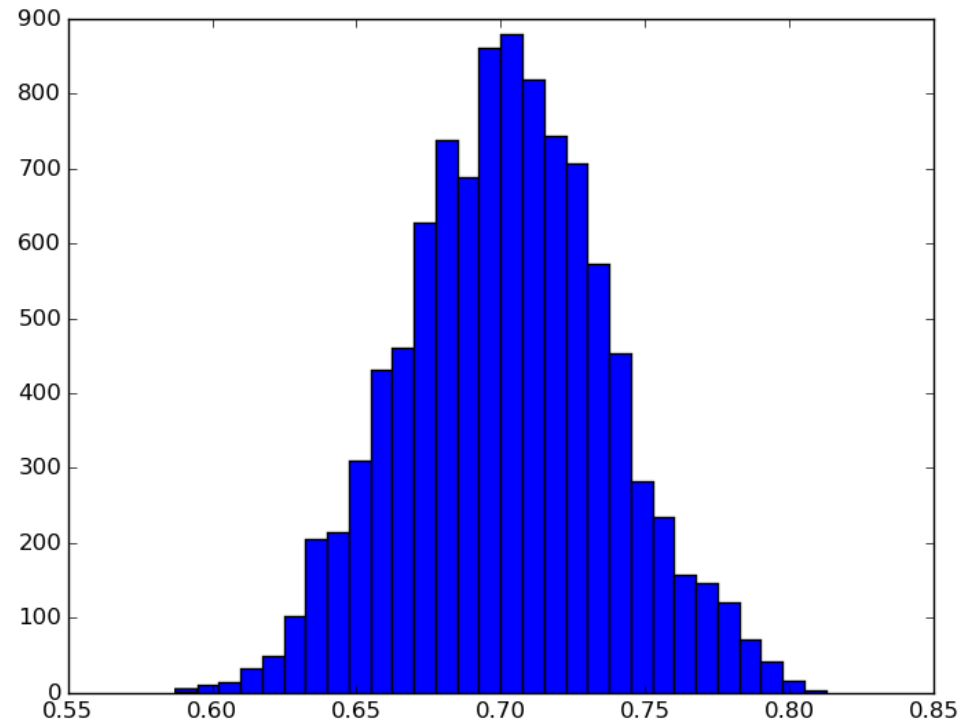
- Step 3: Compute the likelihood of the new point and compare it to the previous point

$$\text{Ratio} = \log \left(\frac{\mathcal{L}(D|\Theta_1)}{\mathcal{L}(D|\Theta_0)} \right) = \chi^2(\Theta_0) - \chi^2(\Theta_1)$$

- if Ratio > 0 then accept the new point as the next step
- If Ratio < 0 then
 - if $\exp(\text{Ratio}) > \text{Rand}(0,1)$ then accept the point
 - if $\exp(\text{Ratio}) < \text{Rand}(0,1)$ then keep the old point as the new point

Plotting the results

- Make a histogram for each parameter



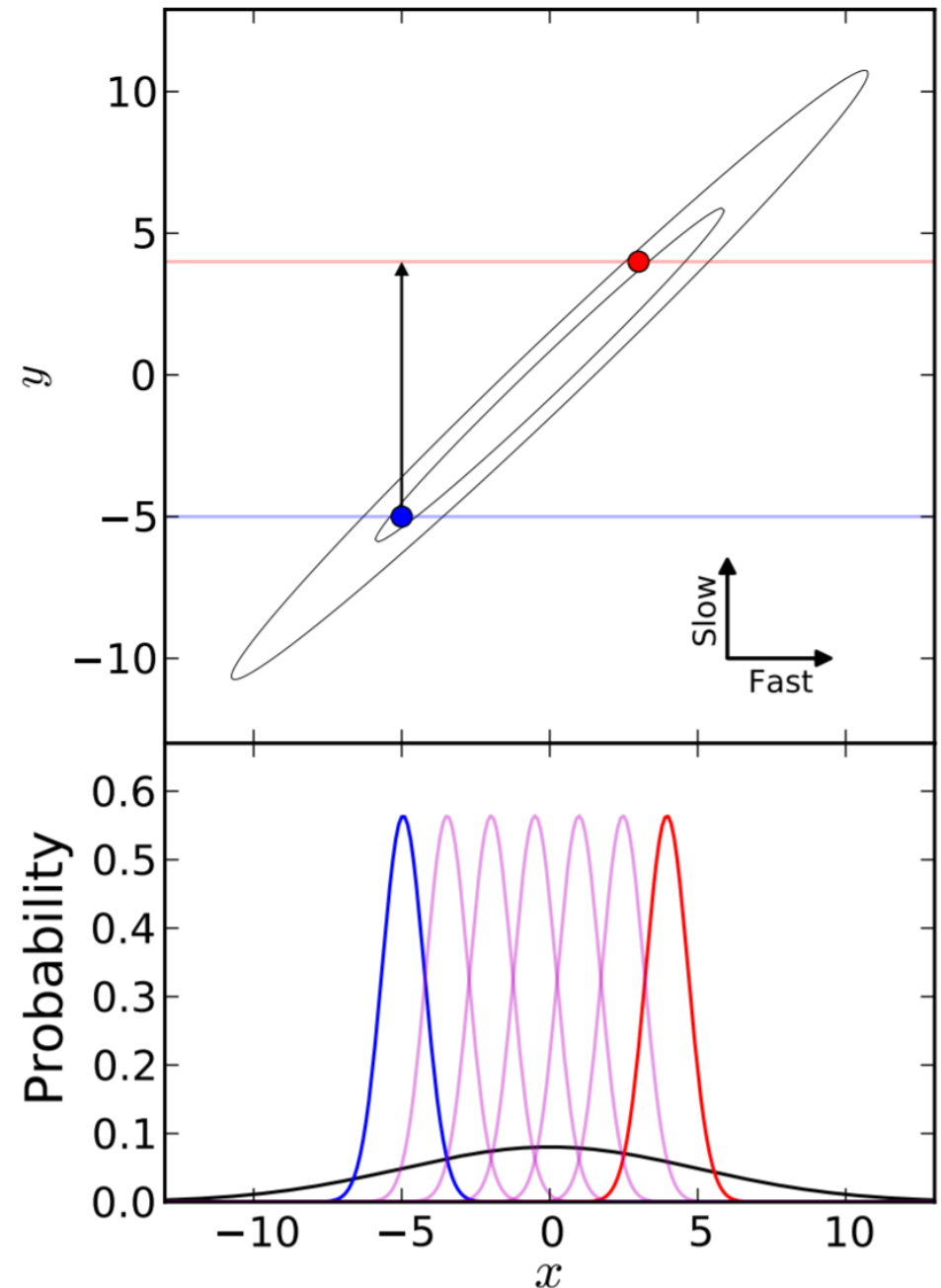
Fast slow sampling

- A partir de y proponer y'
- Interpolar la probabilidad

$$\ln P_i(x) \equiv \frac{(n-i) \ln P(x|y) + i \ln P(x|y')}{n}$$

- Dar uno o dos pasos de Metrópolis en x por cada i
- Aceptar el nuevo punto con probabilidad

$$\min \left[1, \exp \left(\frac{1}{n} \sum_{i=0}^{n-1} [\ln P(x_i, y) - \ln P(x_i, y')] \right) \right]$$

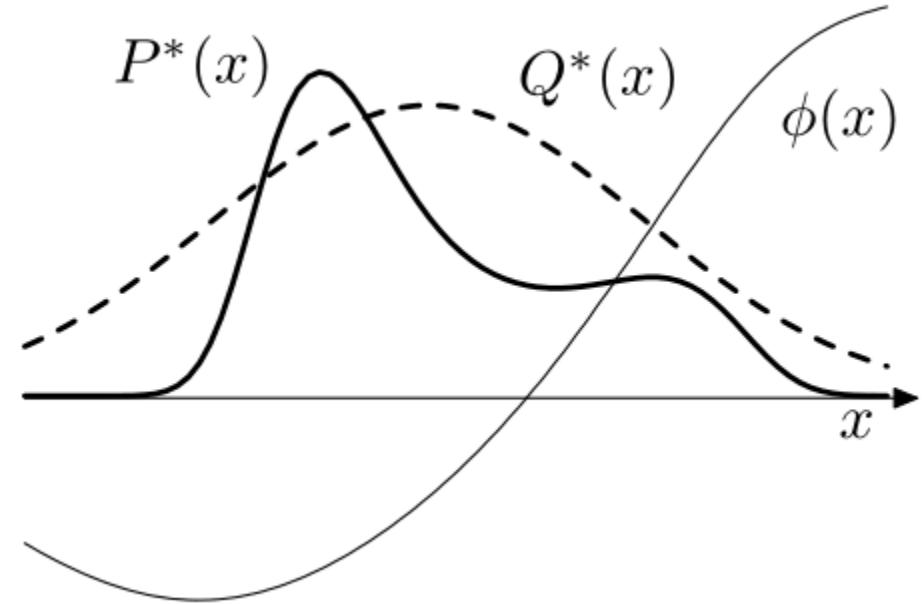


Importance sampling

- Si la probabilidad de tu muestreo Q es distinta que la probabilidad que quieres usar P

$$w_r \equiv \frac{P^*(x^{(r)})}{Q^*(x^{(r)})}$$

$$\hat{\Phi} \equiv \frac{\sum_r w_r \phi(x^{(r)})}{\sum_r w_r}$$



Bajar y compilar

- Requerimientos
 - ifort>14 o gfortran>6
- Alternativamente

```
docker run -v /local/code/source:/virtual_path -i -t cmbant/cosmobox /bin/bash
```

 - Para Windows -v /c/Users:/c/Users
- Correr

```
git clone https://github.com/cmbant/CosmoMC
cd CosmoMC
make
./cosmomc test.ini
```
- Más información <https://cosmologist.info/CosmoBox/>

Archivo de configuración (test.ini)

- Los parámetros se señalan como
 - `param[paramname] = center min max start_width propose_width`
 - Ver `batch3/params_CMB_defaults.ini`
- Para agregar nuevos likelihoods
 - `DEFAULT(batch3/DR12_final_consensus.ini)`
 - ¡Ojo! No todos los likelihoods son compatibles entre sí
 - WIGGLEZ y CMASS DR11
 - Diferentes redshift space distortions
- Nombre de los resultados
 - `root_dir = chains/`
`file_root=test`
- `action= 0` runs chains, `1` importance samples, `2` minimizes
- `checkpoint=True`

Archivo de configuración (common.ini)

- num_threads = 0
- MPI_Converge_Stop = 0.01
- MPI_LearnPropose = T
- dragging_steps = 3
- use_fast_slow = T

Likelihoods

- `data/DR12/sdss_DR12Consensus_final.dataset`

`#BAO+FS consensus results, Alam et al. 2016`

`#https://arxiv.org/abs/1607.03155`

`name = DR12final`

`num_bao=9`

`bao_measurements_file = sdss_DR12Consensus_final.dat`

`bao_measurements_file_has_error = F`

`#rs_rescale = 1/147.78`

`rs_rescale = .6766815537e-2`

`bao_cov_file = final_consensus_covtot_dM_Hz_fsig.txt`

Agregar nuevos parámetros

- Seguir las instrucciones de mgcosmomc
 - <http://aliojjati.github.io/MGCAMB/mgcosmomc-may11-instructions.html>

Outputs

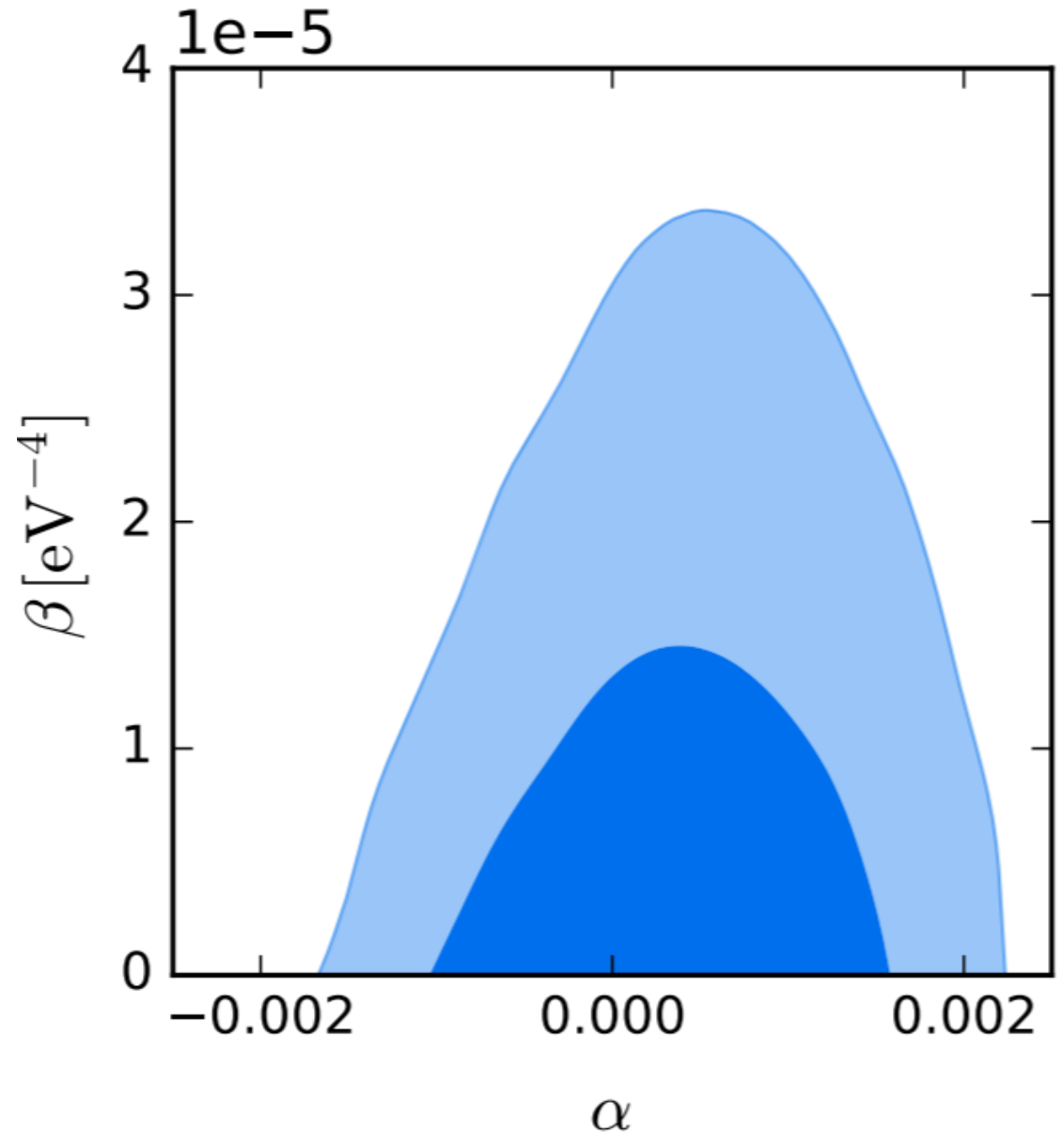
- .likelihoods
- .ranges
- _1.log
- .inputparams
- .paramnames
- _1.data
- _1.txt
- _1.chk

Getdist

- python/GetDist.py
- python/GetDistGUI.py
- Python scripts

```
import planckStyle
g = planckStyle.getSinglePlotter(chain_dir = './PLA')
roots = ['base_nnu_mnu_plikHM_TTTEEE_lowl_lowE',
'base_nnu_mnu_plikHM_TTTEEE_lowl_lowE_post_lensing',
'base_nnu_mnu_plikHM_TTTEEE_lowl_lowE_lensing_BAO']
g.plot_2d(roots, 'mnu', 'nnu', filled=True)
g.add_legend(['Planck', 'Planck+lensing',
'Planck+lensing+BAO'], legend_loc='upper right');
g.export('mnu_nnu.png')
```

Ver: <http://getdist.readthedocs.io/en/latest/>
También: CosmoMC/python/docs/plot_gallery.ipynb



Getdist

- python/GetDist.py
- python/GetDistGUI.py
- Python scripts

```
import planckStyle
g = planckStyle.getSinglePlotter(chain_dir = './PLA')
roots = ['base_nnu_mnu_plikHM_TTTEEE_lowl_lowE',
'base_nnu_mnu_plikHM_TTTEEE_lowl_lowE_post_lensing',
'base_nnu_mnu_plikHM_TTTEEE_lowl_lowE_lensing_BAO']
g.plot_2d(roots, 'mnu', 'nnu', filled=True)
g.add_legend(['Planck', 'Planck+lensing',
'Planck+lensing+BAO'], legend_loc='upper right');
g.export('mnu_nnu.png')
```

Ver: <http://getdist.readthedocs.io/en/latest/>

También: CosmoMC/python/docs/plot_gallery.ipynb

- Archivos de salida
 - file_root.margestats
 - file_root.likestats
 - file_root.converge
 - file_root.PCA
 - file_root.corr
 - file_root.covmat

Referencias

- MacKay, Information Theory, Inference and Learning Algorithms
- <https://cosmologist.info/cosmomc/readme.html>
- https://cosmologist.info/cosmomc/readme_planck.html
- <http://getdist.readthedocs.io/en/latest/>