Python Plots & Plots with GetDist

MANUEL TRASHORRAS, IFT/UAM-CSIC

SCHOOL OF COSMOLOGY TOOLS, MARCH 2017

Contents

1. CosmoMC outputs:

   chains, input parameters, parameter names and ranges.

2. The GetDist package.

   How to compile & analyze MCMC chains.

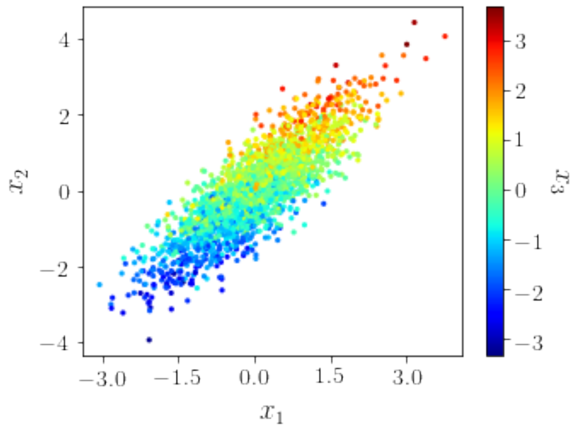
3. GetDist outputs:

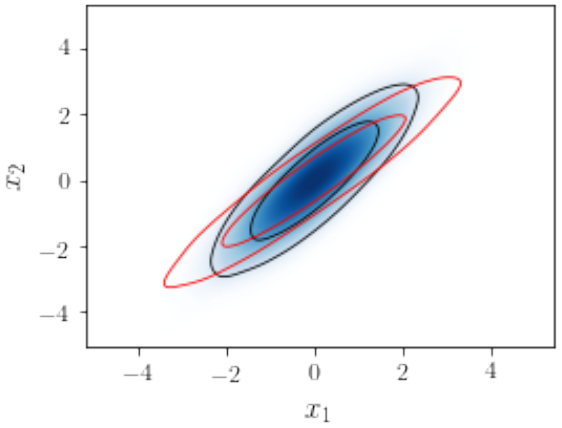
   convergence, likelihood, marginalization and covariance/correlation.

4. The GetDist GUI.

   How to compile & make interactive plots.

5. Useful links.

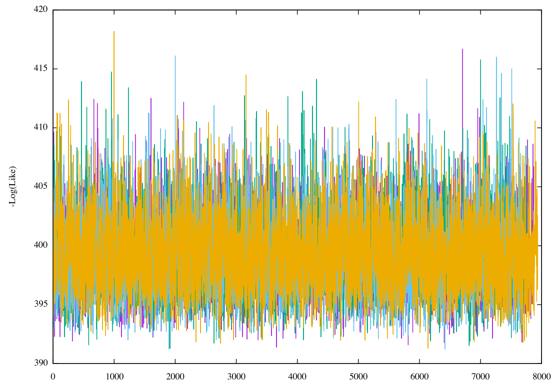


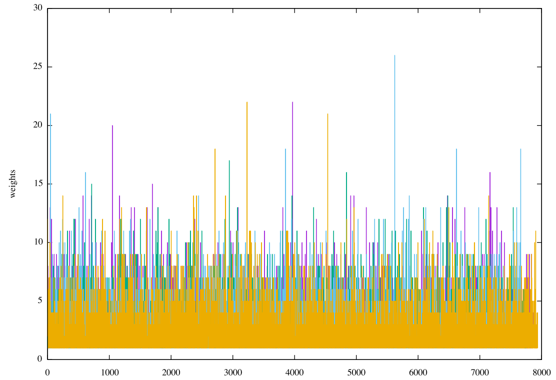


CosmoMC outputs

CosmoMC output files:

- root.inputparams → CosmoMC execution parameters.
- root.likelihoods → Likelihoods taking part in the MCMC.
- root.paramnames → Names of the chain parameters.
- root.ranges → Min. & max. values of the parameters.
- root_1.txt, root_2.txt… → Chain file with weight, likelihood…





Of these, only go into GetDist/GetDist GUI...

{ root.ranges, root.paramnames, root_1.txt, root_2.txt… } Let’s see them closer

CosmoMC outputs

root.ranges

```
omegabh2         0.5000000E-02   0.1000000E+00
omegach2         0.1000000E-02   0.9900000E+00
theta            0.5000000E+00   0.1000000E+02
tau              0.1000000E-01   0.8000000E+00
omegak           0.0000000E+00   0.0000000E+00
mnu              0.6000000E-01   0.6000000E-01
meffsterile      0.0000000E+00   0.0000000E+00
w               -0.1000000E+01  -0.1000000E+01
wa               0.0000000E+00   0.0000000E+00
nnu              0.3046000E+01   0.3046000E+01
yhe              0.2400000E+00   0.2400000E+00
alpha1           0.0000000E+00   0.0000000E+00
deltazrei        0.5000000E+00   0.5000000E+00
Alens            0.1000000E+01   0.1000000E+01
Alensf          -0.1000000E+01  -0.1000000E+01
fdm              0.0000000E+00   0.0000000E+00
logA             0.2000000E+01   0.4000000E+01
ns               0.8000000E+00   0.1200000E+01
nrun             0.0000000E+00   0.0000000E+00
nrunrun          0.0000000E+00   0.0000000E+00
r                0.0000000E+00   0.0000000E+00
nt               0.0000000E+00   0.0000000E+00
ntrun            0.0000000E+00   0.0000000E+00
Aphiphi          0.1000000E+01   0.1000000E+01
calPlanck        0.9000000E+00   0.1100000E+01
acib217          0.0000000E+00   0.2000000E+03
ncib            -0.1300000E+01  -0.1300000E+01
xi               0.0000000E+00   0.1000000E+01
asz143           0.0000000E+00   0.1000000E+02
```

Gives min. (1st col.) and max. (2nd) col) value of the prior. Hard limit.

1st column: variable names within the code.

2nd column: LaTeX names for plots format.

* stands for derived params, not MCMCs.

root.paramnames

```
omegabh2        \Omega_b h^2
omegach2        \Omega_c h^2
theta   100\theta_{MC}
tau     \tau
logA    {\rm{ln}}(10^{10} A_s)
ns      n_s
calPlanck       y_{\rm cal}
acib217 A^{CIB}_{217}
xi      \xi^{tSZ-CIB}
asz143  A^{tSZ}_{143}
aps100  A^{PS}_{100}
aps143  A^{PS}_{143}
aps143217       A^{PS}_{143\times 217}
aps217  A^{PS}_{217}
aksz    A^{kSZ}
kgal100 A^{{\rm dust}TT}_{100}
kgal143 A^{{\rm dust}TT}_{143}
kgal143217      A^{{\rm dust}TT}_{143\times 217}
kgal217 A^{{\rm dust}TT}_{217}
cal0    c_{100}
cal2    c_{217}
H0*     H_0
omegal* \Omega_\Lambda
omegam* \Omega_m
omegamh2*       \Omega_m h^2
omeganuh2*      \Omega_\nu h^2
omegamh3*       \Omega_m h^3
sigma8* \sigma_8
s8omegamp5*     \sigma_8 \Omega_m^{0.5}
```

CosmoMC outputs

root_1.txt

6.000000E+00	4.020593E+02	2.237905E-02	1.210001E-01	1.040673E+00
7.961286E-02	3.095802E+00	9.689967E-01	1.000905E+00	5.072992E+01
9.992040E-01	3.612232E+00	2.627977E+02	3.357852E+01	3.989548E+01
1.090350E+02	6.971365E+00	1.173381E+01	9.813909E+00	1.323354E+01
7.928245E+01	9.996154E-01	9.943139E-01	6.693115E+01	6.785014E-01
3.214986E-01	1.440243E-01	6.451439E-04	9.639711E-02	8.366901E-01
4.744104E-01	6.300274E-01	1.022705E+00	2.510974E+00	1.011810E+01
2.210495E+00	1.885119E+00	1.228440E+03	5.695538E+03	2.539407E+03
8.190871E+02	2.324751E+02	9.689967E-01	2.453968E-01	2.467233E-01
2.589648E+00	1.380795E+01	1.089994E+03	1.441675E+02	1.040856E+00
1.385086E+01	1.060047E+03	1.468169E+02	1.411623E-01	1.606773E-01
3.426328E+03	1.045747E-02	8.088600E-01	4.470520E-01	7.099794E-02
9.281875E+01	1.395937E+03	6.785478E-01	4.889963E-01	6.199804E-01
2.472889E+01	2.928717E+01	1.041134E+02	1.359998E+01	7.821099E+02
8.408603E+00	7.957099E+02			
1.000000E+00	4.051888E+02	2.200543E-02	1.235358E-01	1.040069E+00
3.637712E-02	3.013929E+00	9.567040E-01	9.996694E-01	7.415064E+01
4.920187E-01	6.157384E+00	3.232672E+02	4.418528E+01	3.463432E+01
7.945601E+01	4.202721E+00	7.431789E+00	8.378352E+00	1.865833E+01
7.722513E+01	9.991016E-01	9.974961E-01	6.555908E+01	6.598728E-01
3.401272E-01	1.461864E-01	6.451439E-04	9.583848E-02	8.081931E-01
4.713416E-01	6.171993E-01	9.981566E-01	2.463189E+00	5.836487E+00
2.036726E+00	1.893808E+00	1.241035E+03	5.693506E+03	2.535620E+03
8.129329E+02	2.288998E+02	9.567040E-01	2.452225E-01	2.465484E-01
2.660936E+00	1.387338E+01	1.090697E+03	1.438041E+02	1.040291E+00
1.382344E+01	1.059322E+03	1.465729E+02	1.411301E-01	1.610327E-01
3.478011E+03	1.061516E-02	7.984244E-01	4.418493E-01	7.007005E-02
9.217150E+01	1.415249E+03	6.831381E-01	4.765368E-01	5.946455E-01
3.156666E+01	3.240660E+01	1.049658E+02	1.475256E+01	7.915088E+02

1st point in the Markov chain.
weight = 6.0 *-LogLike = 4.0E+02* …

In general we are interested in the 1D (histograms) and 2D (contours) marginalized posterior probability densities of this parameters.

2nd point in the Markov chain.
weight = 1.0 *-LogLike = 4.1E+02* …

Plotting & analyzing MCMCs with GetDist

GETDIST COMPILATION

Stand alone version: A. Lewis, March 2017

GitHub: https://github.com/cmbant/getdist

1. tar -xvf getdist-master.tar.gz
2. cd getdist-master
3. python setup.py install
4. python GetDist.py

CosmoMC version: A. Lewis, March 2017

GitHub: https://github.com/cmbant/CosmoMC

1. tar –xvf cosmomc-master
2. cd cosmomc-master
3. make getdist
4. ./getdist distparams.ini chains/root

- Caution! GetDist only supported by versions Python 2.7 or newer.

  In Hydra = Python version 2.7

- Caution! You need to add CosmoMC's python path to your environment variables.

  export PYTHONPATH=COSMOMCPATH/python:$PYTHONPATH

- There is a ton of scripts to plot the chains in the /COSMOMCPATH/python directory…

  … that can be used as a black box.

Plotting & analyzing MCMCs with GetDist

distparams.ini

./getdist distparams.ini chains/root

- Caution! The GetDist output folder has to be created beforehand or no output is created.
- Caution! Some parameters have hard boundaries that have to be taken care of:

  #Need to give limits if prior cuts off distribution where not very small

  limits[r02]= 0 N

- For 1D plots:

  #Parameters to use. If not specified use all parameters which have labels.

  #plot_params = PARAM1 PARAM2 PARAM3...

- For 2D plots:

  plot_2D_param = 0

  #if both zero it will plot most correlated variables

  plot_2D_num = 0

  plot1 = XPARAM YPARAM

  plot2 = XPARAM YPARAM

Plotting & analyzing MCMCs with GetDist

./getdist distparams.ini

```
[mtrashorras@hydra0 cosmomc-master]$ ./getdist distparams.ini
 producing files in directory ./chains/plikHM_TT_lowTEB_lensing/dist/
 reading chains/plikHM_TT_lowTEB_lensing/base_plikHM_TT_lowTEB_lensing_1.txt
 reading chains/plikHM_TT_lowTEB_lensing/base_plikHM_TT_lowTEB_lensing_2.txt
 reading chains/plikHM_TT_lowTEB_lensing/base_plikHM_TT_lowTEB_lensing_3.txt
 reading chains/plikHM_TT_lowTEB_lensing/base_plikHM_TT_lowTEB_lensing_4.txt
 Number of chains used =             4
 var(mean)/mean(var), remaining chains, worst e-value: R-1 =      0.00644
 RL: Thin for Markov:         20
 RL: Thin for indep samples:  22
 RL: Estimated burn in steps: 126 (55 rows)
 mean input multiplicity =    2.27828971618135
 Random seeds: 31201, 17315 rand_inst:   0
 using        24417  rows, processing           2  parameters
 Approx indep samples:         2529
 Best fit sample -log(Like) =    5638.37700000000
 Ln(mean 1/like) =    5652.96527696196
 mean(-Ln(like)) =    5646.14296077585
 -Ln(mean like)  =    5643.28888428720
 Producing            1  2D plots
 producing            1 2D colored scatter plots
[mtrashorras@hydra0 cosmomc-master]$
```

Plotting & analyzing MCMCs with GetDist

distparams.ini

./getdist distparams.ini (chains/root)

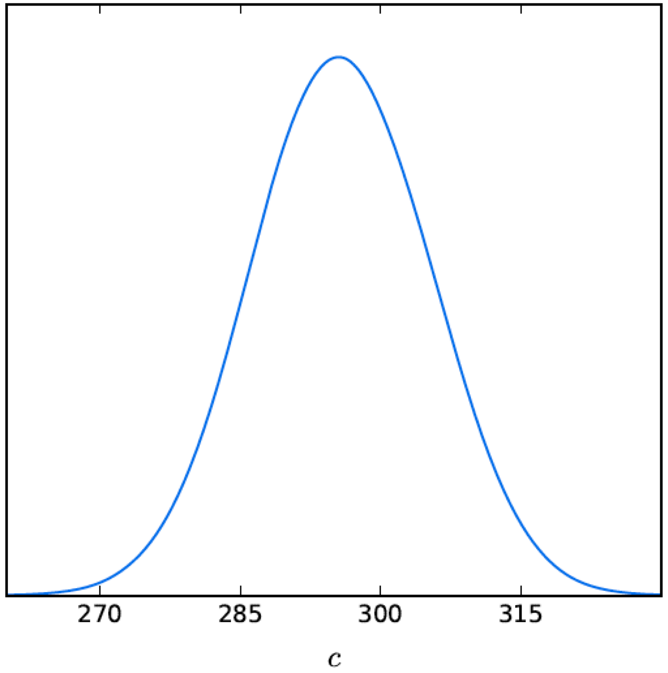
- Caution! The GetDist output folder has to be created beforehand or no output is created.
- Caution! Some parameters have hard boundaries that have to be taken care of:

```
#Need to give limits if prior cuts off distribution where not very small
limits[r02]= 0 N
```

1. Set input & output folders:
   - file_root = chains/root
   - out_root =
   - out_dir = ./chains
   - plot_data_dir = ./chains
2. Set which chains to read:
   - chain_num = 4
   - first_chain = 1
   - exclude_chain = 3
3. Disregard burn-in if using raw chains:
   - ignore_rows = 0.3

Plotting & analyzing MCMCs with GetDist

distparams.ini



- For 1D plots:

  #Parameters to use. If not specified use all parameters which have labels.

  plot_params = PARAM1 PARAM2 PARAM3…

- For 2D plots:

  plot_2D_param = 0

  #if both zero it will plot most correlated variables

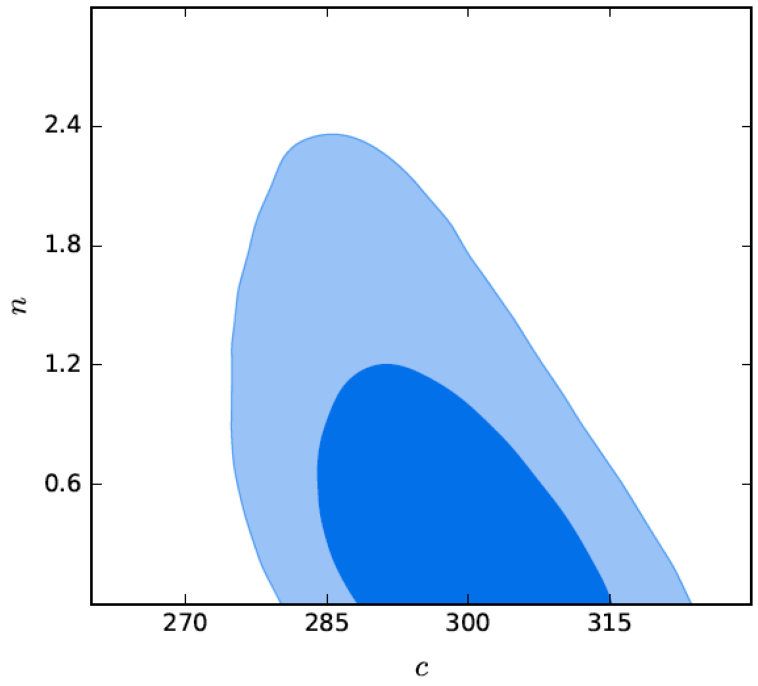
  plot_2D_num = 0

  plot1 = XPARAM YPARAM

  plot2 = XPARAM YPARAM

Plotting & analyzing MCMCs with GetDist

distparams.ini



- **For 1D plots:**

  #Parameters to use. If not specified use all parameters which have labels.

  #plot_params = PARAM1 PARAM2 PARAM3...

- **For 2D plots:**

  plot_2D_param = 0

  #if both zero it will plot most correlated variables

  plot_2D_num = 0

  plot1 = XPARAM YPARAM

  plot2 = XPARAM YPARAM

Plotting & analyzing MCMCs with GetDist

distparams.ini

./getdist distparams.ini chains/root

- Caution! The GetDist output folder has to be created beforehand or no output is created.
- Caution! Some parameters have hard boundaries that have to be taken care of:

#Need to give limits if prior cuts off distribution where not very small

limits[r02]= 0 N

- For 3D plots:

#number of sample plots, colored by third parameter

#if last parameter is 0 or -1 colored by the parameter most correlated

#with one of the eigenvector directions (e.g. parallel or orthogonal to degeneracy)

num_3D_plots = 1

3D_plot1 = XPARAM YPARAM ZPARAM

- For triangle 2D plots:

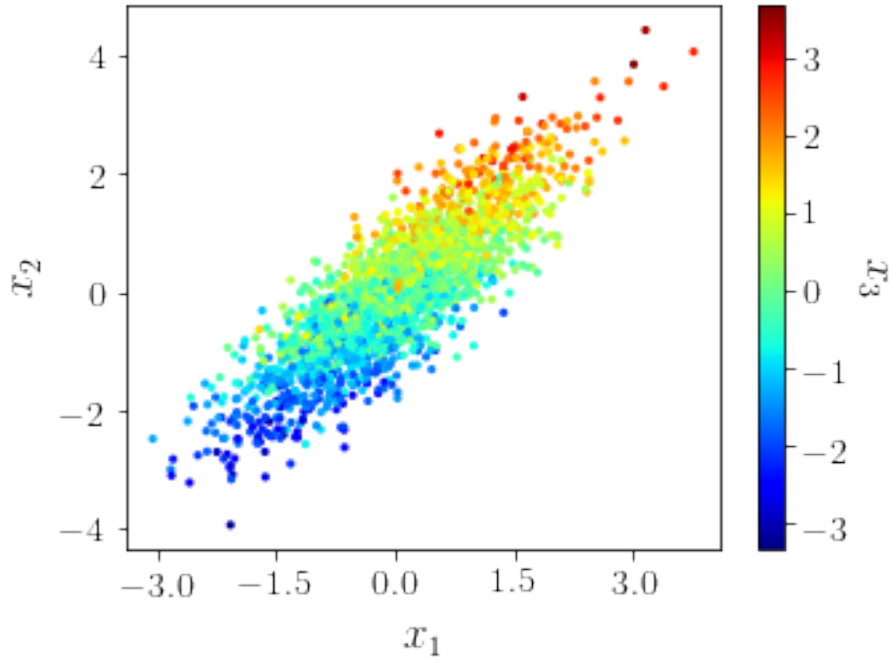
#Output 2D plots for param combos with 1D marginalized plots along the diagonal

triangleplot = T

triangle_params = PARAM1 PARAM2 PARAM3…

Plotting & analyzing MCMCs with GetDist

distparams.ini



- For 3D plots:

  #number of sample plots, colored by third parameter

  #if last parameter is 0 or -1 colored by the parameter most correlated

  #with one of the eigenvector directions (e.g. parallel or orthogonal to degeneracy)

  num_3D_plots = 1

  3D_plot1 = XPARAM YPARAM ZPARAM

- For triangle 2D plots:

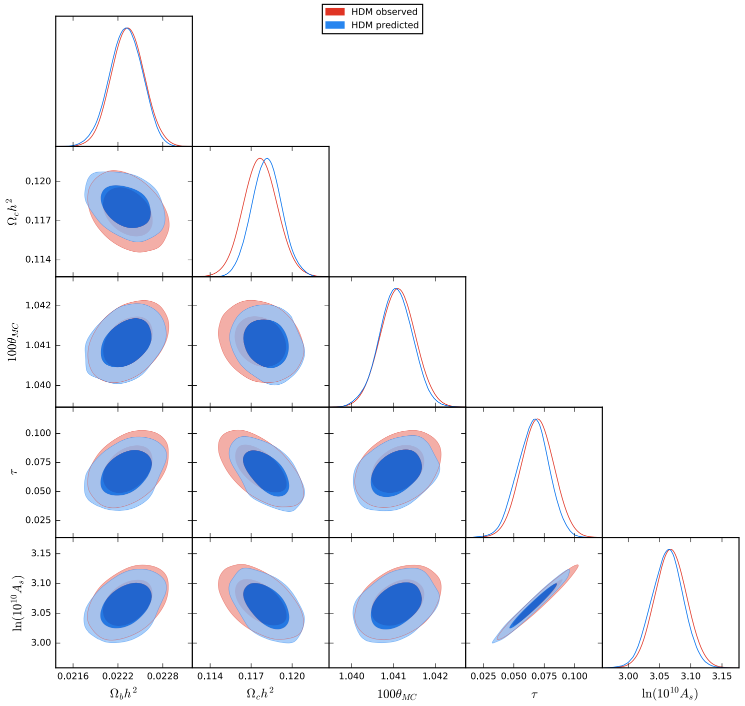
  #Output 2D plots for param combos with 1D marginalized plots along the diagonal

  triangleplot = T

  triangle_params = PARAM1 PARAM2 PARAM3…

Plotting & analyzing MCMCs with GetDist

distparams.ini



- For 3D plots:

  #number of sample plots, colored by third parameter

  #if last parameter is 0 or -1 colored by the parameter most correlated

  #with one of the eigenvector directions (e.g. parallel or orthogonal to degeneracy)

  num_3D_plots = 1

  3D_plot1 = XPARAM YPARAM ZPARAM

- For triangle 2D plots:

  #Output 2D plots for param combos with 1D marginalized plots along the diagonal

  triangleplot = T

  triangle_params = PARAM1 PARAM2 PARAM3…

GetDist outputs

root.converge → Information about chain R convergence for each parameter.

root.likestats (action=2) → Best-fit values of each parameters at the point of maximum likelihood.
This is NOT the mean of the value after marginalization.

root.margestats (action=4) → μ, σ, lower and upper limits at the 68%, 95% & 99% C.L. of each parameter.
It takes into account whether the distribution has one tail (like r)…
… or two tails (like $\Omega_b h^2$, $\Omega_c h^2$, τ, $100\theta_{MC}$, n_s, A_s…),
… or uneven (plus/minus) upper and lower error bands.
This are the distribution values you put on your paper tables.

root.covmat → Covariance matric Cov_{XY} of the MCMC chain: $Cov_{XY} = E[(X-\mu_X)(Y-\mu_Y)]$

root.corrmat → Correlation matrix $Corr_{XY}$ of the MCMC chain: $Corr_{XY} = E[(X-\mu_X)(Y-\mu_Y)]/(\sigma_X\sigma_Y)$
Using this can accellerate A LOT the convergence of your chain.

GetDist outputs: Conv. & Like. stats.

root.converge

```
Variance test convergence stats using remaining chains
param var(chain mean)/mean(chain var)

 1      0.00188  \Omega_b h^2
 2      0.00032  \Omega_c h^2
 3      0.00078  100\theta_{MC}
 4      0.00130  \tau
 5      0.00138  {\rm{ln}}(10^{10} A_s)
 6      0.00020  n_s
 7      0.00055  y_{\rm cal}
 8      0.00043  A^{CIB}_{217}
 9      0.00028  \xi^{tSZ-CIB}
10      0.00019  A^{tSZ}_{143}
11      0.00031  A^{PS}_{100}
12      0.00017  A^{PS}_{143}
13      0.00021  A^{PS}_{143\times 217}
14      0.00037  A^{PS}_{217}
15      0.00061  A^{kSZ}
16      0.00054  A^{{\rm dust}TT}_{100}
17      0.00070  A^{{\rm dust}TT}_{143}
18      0.00011  A^{{\rm dust}TT}_{143\times 217}
19      0.00047  A^{{\rm dust}TT}_{217}
20      0.00010  c_{100}
21      0.00013  c_{217}
22      0.00036  H_0
23      0.00034  \Omega_\Lambda
24      0.00034  \Omega_m
25      0.00034  \Omega_m h^2
27      0.00169  \Omega_m h^3
```

root.likestats

```
Best fit sample -log(Like) =    391.208700000000
Ln(mean 1/like) =    408.142147850205
mean(-Ln(like)) =    399.192801115134
-Ln(mean like)  =    396.137287781809

param  bestfit        lower1         upper1         lower2         upper2
    1  0.2254609E-01  0.2156323E-01  0.2336626E-01  0.2139829E-01  0.2341123E-01   \Omega_b h^2
    2  0.1175199E+00  0.1097975E+00  0.1267073E+00  0.1087987E+00  0.1276735E+00   \Omega_c h^2
    3  0.1041411E+01  0.1039323E+01  0.1043063E+01  0.1039075E+01  0.1043122E+01   100\theta_{MC}
    4  0.1389374E+00  0.1006788E-01  0.2216860E+00  0.1000718E-01  0.2216860E+00   \tau
    5  0.3206859E+01  0.2955915E+01  0.3358265E+01  0.2948083E+01  0.3358946E+01   {\rm{ln}}(10^{10} A_s)
    6  0.9740042E+00  0.9465927E+00  0.9967470E+00  0.9443846E+00  0.1001550E+01   n_s
    7  0.1000669E+01  0.9920820E+00  0.1008768E+01  0.9918367E+00  0.1008859E+01   y_{\rm cal}
    8  0.6017307E+02  0.4051734E+02  0.8401751E+02  0.3747470E+02  0.8964108E+02   A^{CIB}_{217}
    9  0.6689172E+00  0.8915621E-04  0.9999306E+00  0.8915621E-04  0.9999306E+00   \xi^{tSZ-CIB}
   10  0.4205002E+01  0.8551188E-01  0.9989486E+01  0.3169491E-01  0.9998132E+01   A^{tSZ}_{143}
   11  0.2537030E+03  0.1529453E+03  0.3397544E+03  0.1463604E+03  0.3571739E+03   A^{PS}_{100}
   12  0.4234664E+02  0.1260847E+02  0.6966265E+02  0.7209575E+01  0.6966265E+02   A^{PS}_{143}
   13  0.3965156E+02  0.9750135E+01  0.6871381E+02  0.4757443E+01  0.6871381E+02   A^{PS}_{143\times 217}
   14  0.9955250E+02  0.6041211E+02  0.1355931E+03  0.5220979E+02  0.1384577E+03   A^{PS}_{217}
   15  0.3807139E+01  0.1304440E-03  0.9982156E+01  0.1304440E-03  0.9994407E+01   A^{kSZ}
   16  0.8351118E+01  0.1470876E+01  0.1410769E+02  0.5086565E+00  0.1421403E+02   A^{{\rm dust}TT}_{100}
   17  0.7560569E+01  0.3028060E+01  0.1513238E+02  0.2440929E+01  0.1552554E+02   A^{{\rm dust}TT}_{143}
   18  0.1321709E+02  0.4205105E+01  0.3113902E+02  0.1658623E+01  0.3228868E+02   A^{{\rm dust}TT}_{143\times 217}
   19  0.7812315E+02  0.5605151E+02  0.1077274E+03  0.5345433E+02  0.1125470E+03   A^{{\rm dust}TT}_{217}
   20  0.9979888E+00  0.9953912E+00  0.1000244E+01  0.9945802E+00  0.1000757E+01   c_{100}
   21  0.9950807E+00  0.9909866E+00  0.1000810E+01  0.9904039E+00  0.1001499E+01   c_{217}
   22  0.6854248E+02  0.6419189E+02  0.7195068E+02  0.6367920E+02  0.7259521E+02   H_0
   23  0.7004918E+00  0.6394513E+00  0.7418866E+00  0.6307364E+00  0.7484496E+00   \Omega_\Lambda
```

GetDist outputs: Conv. & Like. stats.

root.converge

Information about chain R convergence for each parameter.

root.likestats

Best-fit values of each parameters at the point of maximum likelihood. This is NOT the mean of the value after marginalization.

GetDist outputs: Marginalized distr.

root.margestats

Marginalized limits: 0.68; 0.95; 0.99

parameter	mean	sddev	lower1	upper1	limit1	lower2	upper2	limit2	lower3	upper3	limit3	
omegabh2	0.2240105E-01	0.2720869E-03	0.2213491E-01	0.2267298E-01	two	0.2187295E-01	0.2293902E-01	two	0.2170909E-01	0.2311312E-01	two	\Omega_b h^2
omegach2	0.1180795E+00	0.2601772E-02	0.1154620E+00	0.1207285E+00	two	0.1130735E+00	0.1231817E+00	two	0.1116684E+00	0.1247566E+00	two	\Omega_c h^2
theta	0.1041093E+01	0.5252627E-03	0.1040571E+01	0.1041617E+01	two	0.1040062E+01	0.1042126E+01	two	0.1039734E+01	0.1042452E+01	two	100\theta_{MC}
tau	0.1122441E+00	0.3308596E-01	0.8022497E-01	0.1484359E+00	two	0.4454194E-01	0.1739681E+00	two	0.2469220E-01	0.1894897E+00	two	\tau
logA	0.3154177E+01	0.6216351E-01	0.3094375E+01	0.3222096E+01	two	0.3026148E+01	0.3270048E+01	two	0.2989510E+01	0.3298291E+01	two	{\rm{ln}}(10^{10} A_s)
ns	0.9711264E+00	0.7949215E-02	0.9630789E+00	0.9791998E+00	two	0.9560376E+00	0.9867325E+00	two	0.9510960E+00	0.9912493E+00	two	n_s
calPlanck	0.1000240E+01	0.2458639E-02	0.9978083E+00	0.1002675E+01	two	0.9953802E+00	0.1005107E+01	two	0.9939051E+00	0.1006521E+01	two	y_{\rm cal}
acib217	0.6232246E+02	0.6733444E+01	0.5558833E+02	0.6901537E+02	two	0.4907236E+02	0.7557394E+02	two	0.4498581E+02	0.7950621E+02	two	A^{CIB}_{217}
xi	0.5313423E+00	0.2832136E+00	0.0000000E+00	0.1000000E+01	none	0.0000000E+00	0.1000000E+01	none	0.0000000E+00	0.1000000E+01	none	\xi^{tSZ-CIB}
asz143	0.5427851E+01	0.1893317E+01	0.3561682E+01	0.7552006E+01	two	0.1637007E+01	0.9010552E+01	two	0.0000000E+00	0.1000000E+02	none	A^{tSZ}_{143}
aps100	0.2517272E+03	0.2818557E+02	0.2234962E+03	0.2794511E+03	two	0.1960807E+03	0.3074713E+03	two	0.1784630E+03	0.3247337E+03	two	A^{PS}_{100}
aps143	0.4084459E+02	0.8177795E+01	0.3268743E+02	0.4901528E+02	two	0.2453979E+02	0.5658535E+02	two	0.1987311E+02	0.6113381E+02	two	A^{PS}_{143}
aps143217	0.3865720E+02	0.9809581E+01	0.2859053E+02	0.4888123E+02	two	0.2004803E+02	0.5736936E+02	two	0.1499268E+02	0.6230958E+02	two	A^{PS}_{143\times 217}
aps217	0.9798726E+02	0.1101387E+02	0.8690086E+02	0.1090775E+03	two	0.7642706E+02	0.1188460E+03	two	0.6917582E+02	0.1260989E+03	two	A^{PS}_{217}
aksz	0.3026377E+01	0.2307365E+01	0.0000000E+00	0.3837883E+01	>	0.0000000E+00	0.7571021E+01	>	0.0000000E+00	0.1000000E+02	none	A^{kSZ}
kgal100	0.7396573E+01	0.1890509E+01	0.5509501E+01	0.9289582E+01	two	0.3704003E+01	0.1112353E+02	two	0.2441027E+01	0.1214036E+02	two	A^{{\rm dust}TT}_{100}
kgal143	0.8954715E+01	0.1856776E+01	0.7110114E+01	0.1077449E+02	two	0.5298666E+01	0.1259713E+02	two	0.4214609E+01	0.1384710E+02	two	A^{{\rm dust}TT}_{143}
kgal143217	0.1691740E+02	0.4170524E+01	0.1275497E+02	0.2108686E+02	two	0.8633815E+01	0.2498006E+02	two	0.6236259E+01	0.2757912E+02	two	A^{{\rm dust}TT}_{143\times 217}
kgal217	0.8186303E+02	0.7436888E+01	0.7442150E+02	0.8929146E+02	two	0.6717990E+02	0.9629155E+02	two	0.6254587E+02	0.1007803E+03	two	A^{{\rm dust}TT}_{217}
cal0	0.9978826E+00	0.7817532E-03	0.9971161E+00	0.9986690E+00	two	0.9963363E+00	0.9994061E+00	two	0.9958315E+00	0.9998717E+00	two	c_{100}
cal2	0.9957631E+00	0.1461775E-02	0.9943244E+00	0.9972108E+00	two	0.9928894E+00	0.9986552E+00	two	0.9919926E+00	0.9995745E+00	two	c_{217}
H0*	0.6812991E+02	0.1211243E+01	0.6690890E+02	0.6935398E+02	two	0.6577937E+02	0.7051601E+02	two	0.6512496E+02	0.7125740E+02	two	H_0
omegal*	0.6954928E+00	0.1593819E-01	0.6801620E+00	0.7129643E+00	two	0.6632804E+00	0.7252183E+00	two	0.6532571E+00	0.7332156E+00	two	\Omega_\Lambda
omegam*	0.3045072E+00	0.1593819E-01	0.2870357E+00	0.3198380E+00	two	0.2747889E+00	0.3367197E+00	two	0.2667916E+00	0.3467447E+00	two	\Omega_m
omegamh2*	0.1411257E+00	0.2421213E-02	0.1386906E+00	0.1435618E+00	two	0.1364763E+00	0.1458851E+00	two	0.1351923E+00	0.1473413E+00	two	\Omega_m h^2

GetDist outputs: Marginalized distr.

root.margestats

μ, σ, lower and upper limits at the 68%, 95% & 99% C.L. of each parameter.

It takes into account whether the distribution has one tail (like r) or two tails (like $\Omega_b h^2$, $\Omega_c h^2$, τ, $100\theta_{MC}$, n_s, A_s and uneven (plus/minus) error bands.

GetDist outputs: Covariance Matrices

root.covmat

```
# omegabh2 omegach2 theta tau logA ns calPlanck acib217 xi asz143 aps100 aps143 aps143217 aps217 aksz kgal100 kgal143 kgal143217 kgal217 cal0 cal2
    0.7403128E-07   -0.4904856E-06    0.8092008E-07    0.5746298E-05    0.1046478E-04    0.1545109E-05    0.2233939E-07   -0.3924507E-03
0.9924280E-05    0.8743173E-04   -0.1579764E-02   -0.8710876E-03   -0.1780580E-03    0.6997603E-04   -0.1213103E-03   -0.1243924E-04
-0.1438233E-04   -0.3858449E-04    0.3016098E-04    0.2974383E-08   -0.3432493E-07
   -0.4904856E-06    0.6769215E-05   -0.7830788E-06   -0.5789203E-04   -0.9961561E-04   -0.1847922E-04   -0.9909124E-07    0.2536625E-02
-0.6428382E-04   -0.5112298E-03    0.1058943E-01    0.5767582E-02    0.1403701E-02   -0.3365252E-03    0.7156340E-03    0.4988922E-04
0.1092689E-03    0.3574208E-03    0.7068472E-04   -0.2082193E-07    0.2802215E-06
    0.8092008E-07   -0.7830788E-06    0.2759009E-06    0.8013429E-05    0.1443024E-04    0.2479319E-05    0.4602097E-07   -0.3676843E-03
0.8162940E-05    0.6235312E-04   -0.1446224E-02   -0.8087629E-03   -0.2359120E-03    0.4553454E-04   -0.9699484E-04   -0.3999631E-05
-0.1509502E-04   -0.4070976E-04    0.4398018E-04    0.2600451E-08   -0.2870789E-07
    0.5746298E-05   -0.5789203E-04    0.8013429E-05    0.1094681E-02    0.2043952E-02    0.1950492E-03   -0.1692286E-05   -0.4789840E-01
0.1211651E-02    0.8941231E-02   -0.1941932E+00   -0.8997870E-01   -0.1027853E-01    0.2021908E-01   -0.1541530E-01   -0.2294695E-02
-0.2447339E-02   -0.7572012E-02   -0.8556180E-03   -0.1439203E-06   -0.5308757E-05
    0.1046478E-04   -0.9961561E-04    0.1443024E-04    0.2043952E-02    0.3864302E-02    0.3436868E-03    0.8618189E-05   -0.9028255E-01
0.2303450E-02    0.1676398E-01   -0.3578979E+00   -0.1655686E+00   -0.1494953E-01    0.4554277E-01   -0.2983471E-01   -0.4745292E-02
-0.5121210E-02   -0.1457946E-01    0.4262537E-03   -0.7323391E-07   -0.1006075E-04
    0.1545109E-05   -0.1847922E-04    0.2479319E-05    0.1950492E-03    0.3436868E-03    0.6319002E-04    0.1323770E-06   -0.1292790E-01
0.3327060E-03    0.2503872E-02   -0.5781278E-01   -0.2343101E-01   -0.9576717E-03    0.7291539E-02   -0.4472613E-02    0.1430250E-03
-0.4512391E-03   -0.1817738E-02    0.2005840E-03   -0.1835668E-06   -0.1287372E-05
    0.2233939E-07   -0.9909124E-07    0.4602097E-07   -0.1692286E-05    0.8618189E-05    0.1323770E-06    0.6044907E-05    0.6099607E-03
-0.5085330E-05    0.2590295E-04    0.3481454E-02    0.1609045E-03    0.1535810E-03    0.1238527E-02    0.9124751E-05   -0.1021976E-05
0.2594028E-04    0.2378471E-04    0.8576764E-03   -0.3239925E-08   -0.2934137E-07
   -0.3924507E-03    0.2536625E-02   -0.3676843E-03   -0.4789840E-01   -0.9028255E-01   -0.1292790E-01    0.6099607E-03    0.4533927E+02
-0.5545483E+00   -0.2171337E+01    0.5201484E+02    0.4308102E+01   -0.2121300E+02   -0.6211410E+02    0.1063413E+01    0.6300170E-01
0.1298894E+00   -0.1682619E+01   -0.1827893E+02   -0.2634071E-03    0.4612065E-02
```

GetDist outputs: Covariance Matrices

root.covmat

Covariance matrix Cov_{XY} of the MCMC chain:

$$Cov_{XY} = E[(X-\mu_X)(Y-\mu_Y)],$$

related to the correlation matrix $Corr_{XY}$ of the MCMC chain by

$$Cov_{XY} = Corr_{XY}/(\sigma_X\sigma_Y).$$

And to the idea of the Figures of Merit in a 2D contour plot:

$$FoM_{XY} = 1/Area_{XY} = |M(Cov)_{XY}|^{1/2} = (\sigma_x{}^2\sigma_Y{}^2 - Cov_{xy}{}^2)^{1/2}$$

The GetDist GUI

GETDIST GUI COMPILATION & EXECUTION

Stand alone version: A. Lewis, March 2017
GitHub: https://github.com/cmbant/getdist

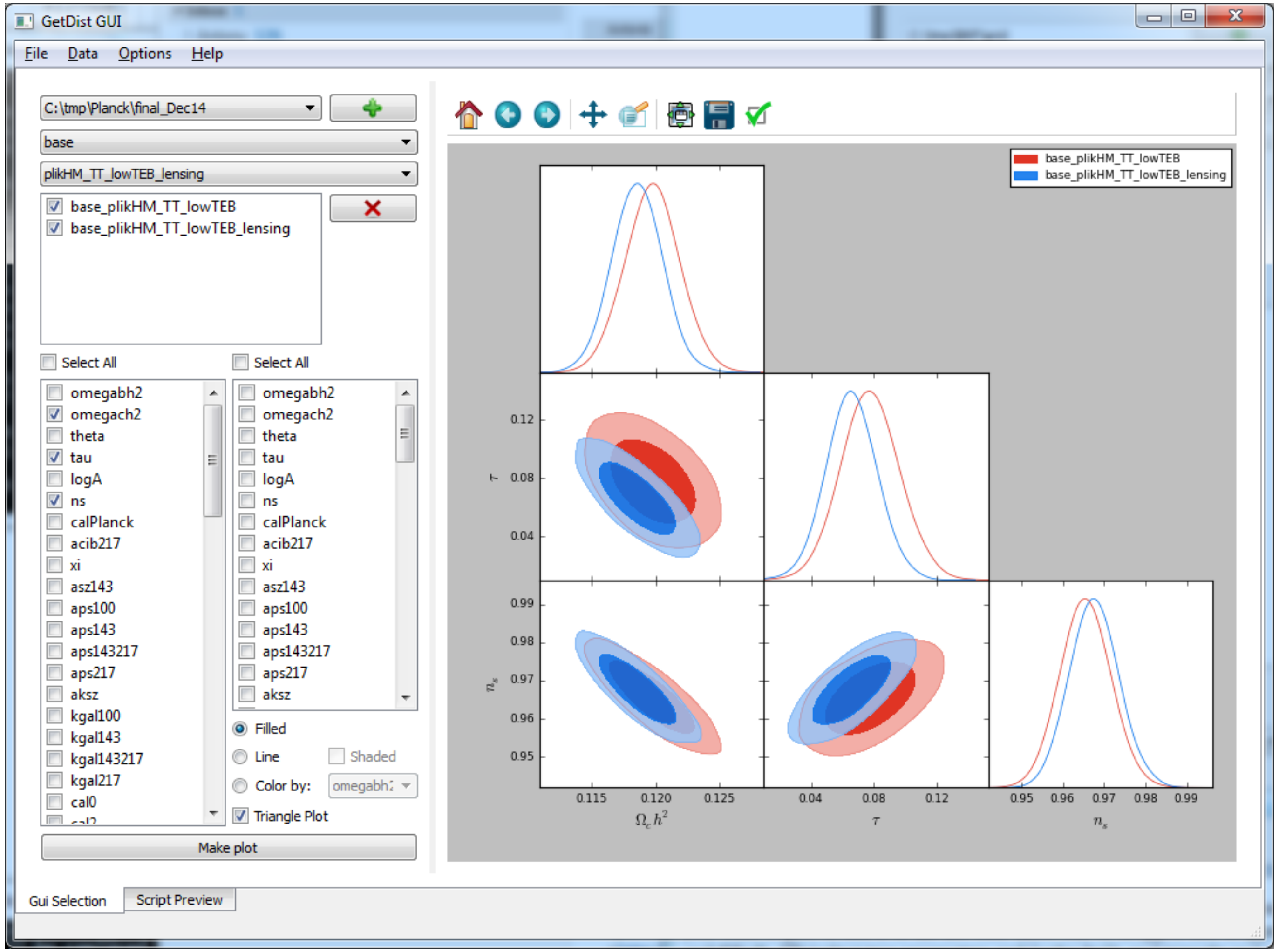
1. Get MacPorts/PiP/Brew to install GetDist
   port install python27
   port select --set python python27 → NN = 27
   port install py-matplotlib
   port install py-scipy
   port install py-pyside
   port install texlive-latex-extra
   port install texlive-fonts-recommended
   port install dvipng
2. Also for the GetDist GUI you also need
   port install pyNN-pyside
   port install pyNN-pyqt4
3. Add to hour ~/.bashr file the lineS
   export LC_ALL=en_US.UTF-8
   export LANG=en_US.UTF-8
4. Compile and launch the GetDist GUI
   1. cd getdist-master
   2. python setup.py install
   3. python GetDistGUI.py

The GetDist GUI

GUI – Interactive mode

Advantages: Extremly easy & fast to use.

Disadvantages: Not as customizable as Python.

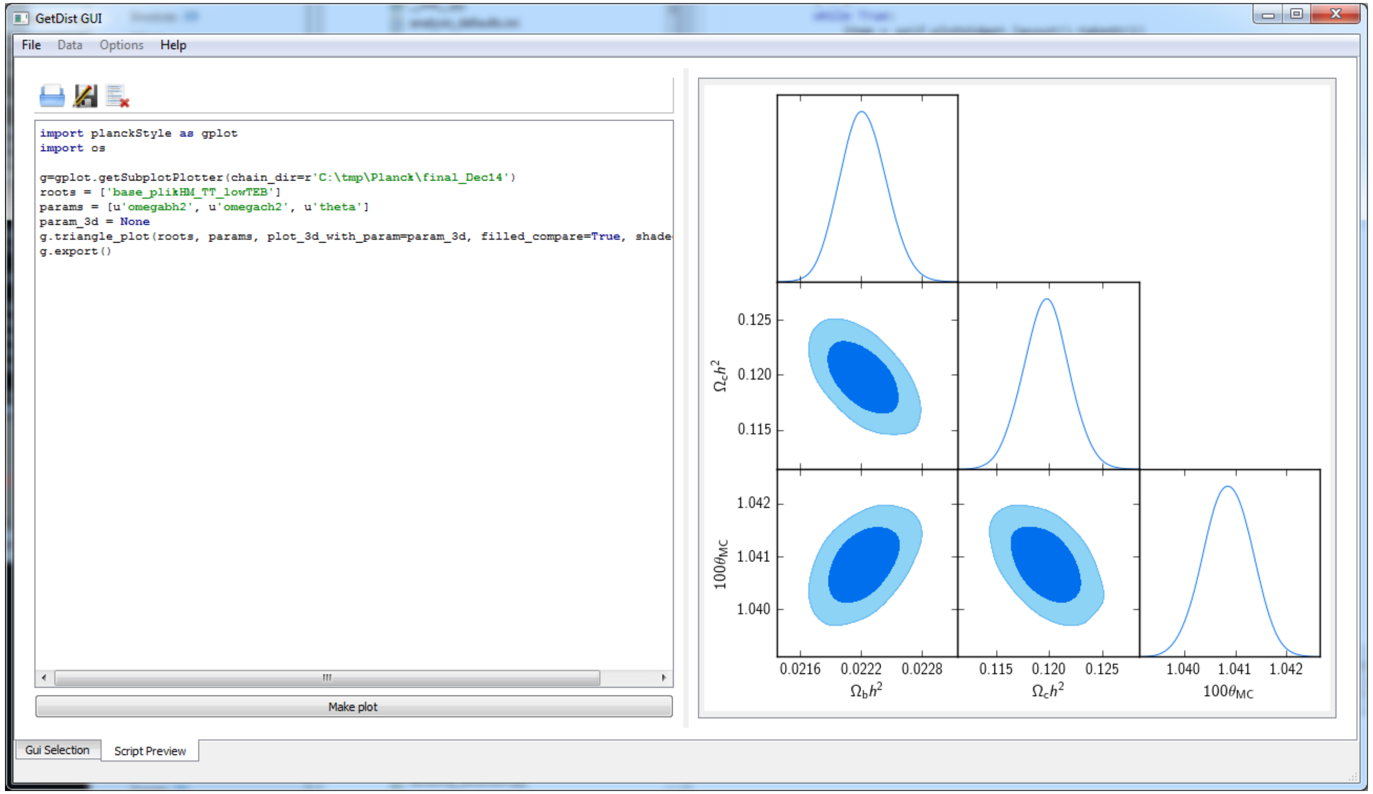


The GetDist GUI

GUI – Python mode

Advantages: Almost customizable as Python.

Disadvantages: Not many, really. You can modify the script as much as you like.



Useful links

Introduction to MCMCs: http://www.uco.es/~ajcuesta/cosmomc-ugr---day-1.pdf

Introduction to CosmoMC: http://www.uco.es/~ajcuesta/cosmomc-ugr---day-2.pdf

Introduction to GetDist: http://www.uco.es/~ajcuesta/cosmomc-ugr---day-3.pdf

CosmoCoffee wiki: http://cosmocoffee.info/

CosmoMC Readme: http://cosmologist.info/cosmomc/readme.html

Planck Readme: http://cosmologist.info/cosmomc/readme_planck.html

Python Readme: http://cosmologist.info/cosmomc/readme_python.html

GetDist GUI Readme : http://cosmologist.info/cosmomc/readme_gui.html