How to run Cosmomc

Pascal M. Vaudrevange

31.07.2009

1 CosmoMC

CosmoMC/Camb (`http://cosmocoffee.info`)

- Camb (`http://camb.info`)
  - Boltzman solver
  - evolves primordial perturbation power spectra defined in
    `powertilt.f90:ScalarPower()`
    `powertilt.f90:TensorPower()`
    to today's C_ℓ
  - stand-alone program and library, OpenMP
- CosmoMC (`http://cosmologist.info/cosmomc`)
  - Parameter estimation package for cosmological parameters
  - uses Camb + MPI
  - needs WMAP5yr likelihood software + cfitsio

Debugging CosmoMC

```
pmv9@master:~/cosmomc> mpirun ./cosmomc params.ini
forrtl: severe (59): list-directed I/O syntax error, unit -5, file Internal List-Directed Read
Image              PC                Routine            Line        Source
cosmomc            00000000005F2B2A  Unknown               Unknown  Unknown
cosmomc            00000000005F1D2A  Unknown               Unknown  Unknown
cosmomc            00000000005A34D6  Unknown               Unknown  Unknown
cosmomc            00000000005572A2  Unknown               Unknown  Unknown
cosmomc            00000000005568AE  Unknown               Unknown  Unknown
cosmomc            000000000057203B  Unknown               Unknown  Unknown
cosmomc            0000000000571069  Unknown               Unknown  Unknown
cosmomc            00000000004B1AE9  Unknown               Unknown  Unknown
cosmomc            0000000000409A22  Unknown               Unknown  Unknown
libc.so.6          0000003DD9E1C3FB  Unknown               Unknown  Unknown
cosmomc            000000000040996A  Unknown               Unknown  Unknown
pmv9@master:~/cosmomc>
```

Debugged

Write your code for GCC!

- Switching compilers to “intel-compilers-8.1-64” did the trick
- adjust path in `likelihood_v3/WMAP_5yr_options.F90`

```
! ============================================================================
MODULE WMAP_OPTIONS

! This module contains the options in the likelihood code
!
! ============================================================================

!-----------------------------------------------
! location of input data
! -----------------------------------------------

  character(len=*), parameter :: WMAP_data_dir = '/home/pmv9/likelihood_v3/data/'
```

CosmoMC ini-file

```
#Sample parameters for cosmomc in default parameterization

#Root name for files produced
file_root = chains/test

#action = 0:  MCMC, action=1: postprocess .data file, action=2: find best fit point only
action = 0

#Maximum number of chain steps
samples = 200000

#Feedback level ( 2=lots,1=chatty,0=none)
feedback = 1

#Temperature at which to Monte-Carlo
temperature = 1

#filenames for CMB datasets and SZ templates (added to C_l times parameter(13))
#Note you may need to change lmax in cmbtypes.f90 to use small scales (e.g. lmax=2100)
cmb_numdatasets = 1
cmb_dataset1 = WMAP
cmb_dataset_SZ1 = data/WMAP_SZ_VBand.dat
cmb_dataset_SZ_scale1 = 1

cmb_dataset2 = data/acbar2007_l2000.newdat
cmb_dataset_SZ2 = data/WMAP_SZ_VBand.dat
cmb_dataset_SZ_scale2 = 0.28

cmb_dataset3 = data/CBIpol_2.0_final.newdat
cmb_dataset4 = data/B03_NA_21July05.newdat

#filenames for matter power spectrum datasets, incl twodf
mpk_numdatasets = 1
mpk_dataset1 = data/sdss_lrgDR4.dataset
#mpk_dataset1 = data/2df_2005.dataset

#if true, use HALOFIT for non-linear corrections (astro-ph/0207664).
#note lyman-alpha (lya) code assumes linear spectrum
nonlinear_pk = F
```

CosmoMC ini-file

```
use_CMB = T
use_HST = F
use_mpk = F
use_clusters = F
use_BBN = F
use_Age_Tophat_Prior = T
use_SN = F
use_lya = F
use_min_zre = 0

#Force computation of sigma_8 even if use_mpk = F
get_sigma8 = F

#1: Simple Metropolis, 2: slice sampling, 3: slice sampling fast parameters, 4: directional gridding
sampling_method = 1

#if sampling_method =4, iterations per gridded direction
directional_grid_steps = 20

#use fast-slow parameter distinctions to speed up
#(note for basic models WMAP3 code is only ~3x as fast as CAMB)
use_fast_slow = F

#Can use covariance matrix for proposal density, otherwise use settings below
#Covariance matrix can be produced using "getdist" program.
propose_matrix = params_CMB.covmat

#If propose_matrix is blank (first run), can try to use numerical Hessian to
#estimate a good propose matrix. As a byproduct you also get an approx best fit point
estimate_propose_matrix = F

#Tolerance on log likelihood to use when estimating best fit point
delta_loglike = 2

#Scale of proposal relative to covariance; 2.4 is recommended by astro-ph/0405462 for Gaussians
#If propose_matrix is much broader than the new distribution, make proportionately smaller
#Generally make smaller if your acceptance rate is too low
propose_scale = 2.4

#Increase to oversample fast parameters more, e.g. if space is odd shape
oversample_fast = 1
```

CosmoMC ini-file

```
#if non-zero number of steps between sample info dumped to file file_root.data
indep_sample = 0

#number of samples to disgard at start; usually set to zero and remove later
burn_in = 0

#If zero set automatically
num_threads = 0

#MPI mode multi-chain options (recommended)
#MPI_Converge_Stop is a (variance of chain means)/(mean of variances) parameter that can be used to stop the cha
#Set to a negative number not to use this feature. Does not guarantee good accuracy of confidence limits.
MPI_Converge_Stop = 0.03

#Do initial period of slice sampling; may be good idea if
#cov matrix or widths are likely to be very poor estimates
MPI_StartSliceSampling  = F

#Can optionally also check for convergence of confidence limits (after MPI_Converge_Stop reached)
#Can be good idea as small value of MPI_Converge_Stop does not (necessarily) imply good exploration of tails
MPI_Check_Limit_Converge = F

#if MPI_Check_Limit_Converge = T, give tail fraction to check (checks both tails):
MPI_Limit_Converge = 0.025
#permitted quantile chain variance in units of the standard deviation (small values v slow):
MPI_Limit_Converge_Err = 0.2
#which parameter's tails to check. If zero, check all parameters:
MPI_Limit_Param = 0

#if MPI_LearnPropose = T, the proposal density is continally updated from the covariance of samples so far (sinc
MPI_LearnPropose = T
#can set a value of converge at which to stop updating covariance (so that it becomes rigorously Markovian)
#e.g. MPI_R_StopProposeUpdate = 0.4 will stop updating when (variance of chain means)/(mean of variances) < 0.4
MPI_R_StopProposeUpdate = 0

#If have covmat, R to reach before updating proposal density (increase if covmat likely to be poor)
#Only used if not varying new parameters that are fixed in covmat
MPI_Max_R_ProposeUpdate = 2
#As above, but used if varying new parameters that were fixed in covmat
MPI_Max_R_ProposeUpdateNew = 30

#if blank this is set from system clock
rand_seed =
```

CosmoMC ini-file

```
#If true, generate checkpoint files and terminated runs can be restarted using exactly the same command
#and chains continued from where they stopped
#With checkpoint=T note you must delete all chains/file_root.* files if you want new chains with an old file_roo
checkpoint = F


#CAMB parameters
#If we are including tensors
compute_tensors = F
#Initial power spectrum amplitude point (Mpc^{-1})
pivot_k = 0.05
#If using tensors, enforce n_T = -A_T/(8A_s)
inflation_consistency = F

#Whether the CMB should be lensed (slows a lot unless also computing matter power)
CMB_lensing = T
#increase accuracy_level to run CAMB on higher accuracy
#(default is about 0.3%, accuracy_level=2 around 0.1% at high l)
accuracy_level = 1

#If action = 1
redo_likelihoods = T
redo_theory = F
redo_cls = F
redo_pk = F
redo_skip = 0
redo_outroot =
redo_thin = 1
redo_add = F
redo_from_text = F
#If large difference in log likelihoods may need to offset to give sensible weights
#for exp(difference in likelihoods)
redo_likeoffset =  0
```

CosmoMC ini-file

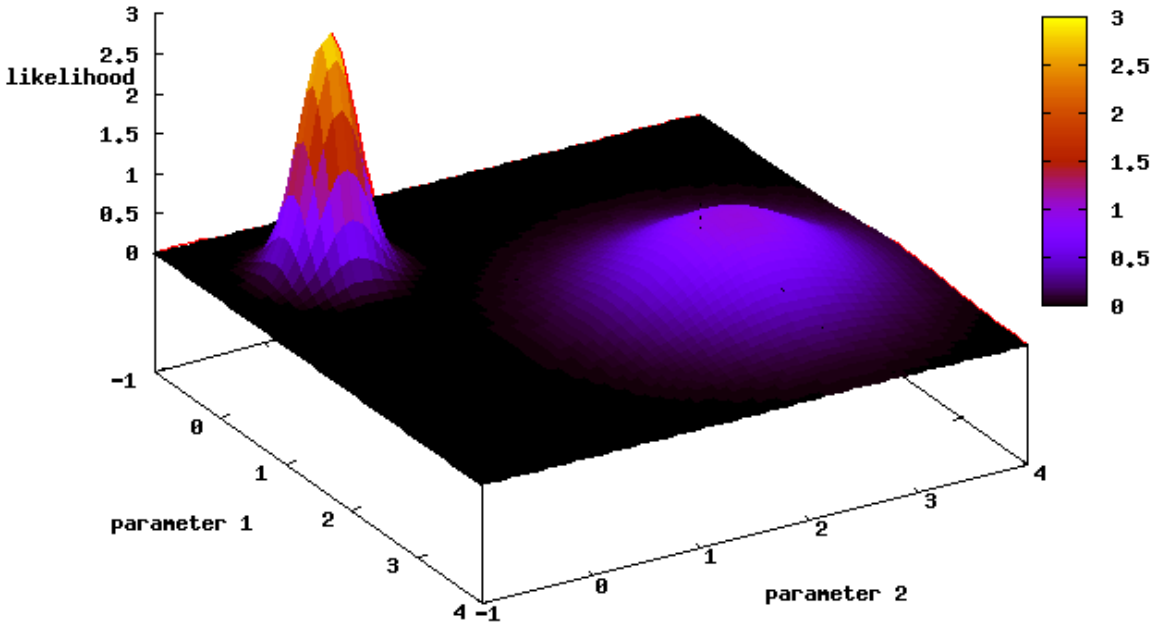
```
#parameter start center, min, max, start width, st. dev. estimate
#ombh2
param1 = 0.0223 0.005 0.1 0.001 0.001
#omdmh2
param2 = 0.105 0.01 0.99 0.01 0.01
#theta
param3 = 1.04 0.5 10 0.002 0.002
#tau
param4 = 0.09 0.01 0.8 0.03 0.03

#omk
param5 = 0 0 0 0 0
#nufrac
param6 = 0 0 0 0 0
#w
param7 = -1 -1 -1 0 0

#n_s
param8 = 0.95 0.5 1.5 0.02 0.01
#n_t
param9 = 0 0 0 0 0
#n_run
param10 = 0 0 0 0 0

#log[10^10 A_s]
param11 = 3 2.7 4 0.01 0.01
#amp_ratio
param12 = 0 0 0 0 0
#SZ amplitude, as in WMAP analysis
param13 = 1 0 2 0.4 0.4
```

CosmoMC ini-file: param*=




```
#parameter start center, min, max, start width, st. dev. estimate
#ombh2
param1 = 0.0223 0.005 0.1 0.001 0.001
```

- `start`: center of Gaussian
- `min, max`: strict bounds
- `start width`: width of Gaussian
- `st.dev.estimate`: estimate of posterior width

- mock likelihood surface
- high-dimensional random walk in parameter space
- several “chains” with different start points
- generate “chains” from posterior

Analysing the output

- output files in `cosmomc/chains/`

```
pascal@octonion:~/projects/cosmomc_june08/chains> ls
simulated_cls_1.log   simulated_cls2_4.log  simulated_test_1.log  test_1.log
simulated_cls_1.txt   simulated_cls2_4.txt  simulated_test_1.txt  test_1.txt
simulated_cls2_1.log  simulated_cls_2.log   simulated_test_2.log  test_2.log
simulated_cls2_1.txt  simulated_cls_2.txt   simulated_test_2.txt  test_2.txt
simulated_cls2_2.log  simulated_cls_3.log   simulated_test_3.log  test_3.log
simulated_cls2_2.txt  simulated_cls_3.txt   simulated_test_3.txt  test_3.txt
simulated_cls2_3.log  simulated_cls_4.log   simulated_test_4.log  test_4.log
simulated_cls2_3.txt  simulated_cls_4.txt   simulated_test_4.txt  test_4.txt
pascal@octonion:~/projects/cosmomc_june08/chains>
```

- `.txt` files contain the chains
- `.log` files contain STDOUT from each chain

GetDist: distparams.ini

```
#Params for "getdist" - for processing .txt chain information

#if zero, columnnum calculated automatically as total number of columns
columnnum = 22
file_root = chains/test
out_root =
out_dir =
plot_data_dir = plot_data/

#If generated chain at higher temperature can cool for computing results
cool = 1

#If 0 assume 1 and no chain filename prefixes
chain_num = 4
first_chain =
exclude_chain =

#width of Gaussian smoothing - Should check plots are robust to changes in
#this parameter. Narrow diagonal distributions need larger number
#Can also check plots by comparing with setting smoothing=F below
num_bins = 20

#For disgarding burn-in if using raw chains
#if < 1 interpreted as a fraction of the total number of rows (0.3 ignores first 30% of lines)
ignore_rows = 0.5

#if T produced B&W printer friendly output
B&W = F
#version of MatLab you are using
matlab_version = 7

#Switches; auto_label labels all parameters by their number
no_plots = F
no_tests = F
auto_label = F
#samples_are_chains = F can be useful for other samples when first two columns not present
samples_are_chains = T

#Include these in 1D plots for comparison - must have same parameters
compare_num = 0
compare1 = basic6_cmb

plot_meanlikes = T
shade_meanlikes = T
```

Getdist distparams.ini

```
# if non-zero, output _thin file, thinned by thin_factor
thin_factor = 0
#Do probabilistic importance sampling to single samples
make_single_samples = F
single_thin = 4

#Do simple importance sampling
adjust_priors = F
#Map parameters to other derived quantities
map_params = F

#Use a Gaussian smoothing with width of the bin size
#Otherwise use top hat bins
smoothing = T

num_contours = 2
contour1 = 0.68
contour2 = 0.95

do_minimal_1d_intervals = F

#if we only want 2D plots agains a particular variable
plot_2D_param = 0

#if above zero, instead plot just these combinations:
#if both zero it will plot most correlated variables
plot_2D_num = 0
plot1 = 8 1
plot2 =

#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 = 1 2 3

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

#Number of parameters to get covariance matrix for
#If you are going to use the output as a proposal density make sure
#you have map_params = F, and the dimension equal to the number of MCMC parameters
cov_matrix_dimension = 13
```

Getdist distparams.ini

```
#e.g. colormap('jet')
matlab_colscheme =

#Parameters to use. If zero use all parameters which have lables.
plotparams_num = 0
plotparams = 1 2 3 11

#labels for the parameters
lab1 = \Omega_b h^2
lab2 = \Omega_c h^2
lab3 = \theta
lab4 = \tau
#Omk
lab5 =
#nufrac
lab6 =
#w
lab7 =

lab8 = n_s
#n_t
lab9 =
lab10 = n_{run}
#markerx adds vertical line to MatLab 1D plot
#marker10 = 0

lab11 = log[10^{10} A_s]
#amp ratio
lab12 = r
lab13 =  A_{SZ}

lab14 = \Omega_\Lambda
lab15 = Age/GYr
lab16 = \Omega_m
lab17 = \sigma_8
lab18 = z_{re}
#r_{10}
lab19 =
lab20 = H_0
```

Getdist distparams.ini

```
#Need to give limits if prior cuts off distribution where not very small
limits4 = 0.01 N
limits6 = 0 N
limits7 = -1 N
limits12 = 0 N
limits13 = 0 2

#all_limits sets all limitsxx for all variables to the same; can be useful for bins
all_limits =

#compute two-tail marginalized limits irrespective of limits settings above
#(otherwise limits are two-tail only for those parameters without limits)
force_twotail = F

#PCA - analysis output in file file_root.PCA
#number of parameter to do PCA for
PCA_num = 0
PCA_normparam = 13
#The parameters to use
PCA_params = 3 13 16
#L for log(x), M for log(-x), N for no log
PCA_func   = LLLL
```

Getdist

```
pascal@octonion:~/projects/cosmomc_june08/chains> cd ..
pascal@octonion:~/projects/cosmomc_june08> ./getdist distparams.ini
 reading chains/test_1.txt
 reading chains/test_2.txt
 reading chains/test_3.txt
 reading chains/test_4.txt
 outlier fraction   4.51467262E-04
 Number of chains used =             4
 var(mean)/mean(var), 1/2 chains, worst e-value: R-1 =    0.0426
 RL: Thin for Markov:          36
 RL: Thin for indep samples:  38
 RL: Estimated burn in steps: 216 (76 rows)
 mean input multiplicity =    2.8404064
 using         4430  rows, processing           12  parameters
 Approx indep samples:          331
 Writing covariance matrix for           13  parameters
 Best fit -Ln(like) =    1329.3149
 Ln(mean 1/like) =    1335.3423
 mean(-Ln(like)) =    1332.2129
 -Ln(mean like)  =    1331.2590
 doing 2D plots for most correlated variables
 Producing           12  2D plots
 producing            1 2D colored scatter plots
pascal@octonion:~/projects/cosmomc_june08>
```

- `R-1`: convergence diagnostics: < 0.1
- `RL`: Raftery-Lewis statistics:
  number independent samples
  length of burn-in
- produces scripts for plotting with
  Matlab
  SuperMongo
- more detailed information in various files

  ```
  test.converge
  test.covmat
  test.corr
  test.likestats
  test.margestats
  ```

test.converge

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

 1   0.0267  \Omega_b h^2
 2   0.0087  \Omega_c h^2
.
.
.
20   0.0147  H_0

var(mean)/mean(var) for eigenvalues of covariance of means
 1  -0.0000
 2  -0.0000
.
.
.
 7   0.0300

Split tests: rms_n([delta(upper/lower quantile)]/sd) n={2,3,4}:
i.e. mean sample splitting change in the quantiles in units of the st. dev.

 1   0.0330   0.0433   0.0809  \Omega_b h^2 upper
 1   0.0400   0.0723   0.0898  \Omega_b h^2 lower
 2   0.0389   0.0899   0.1141  \Omega_c h^2 upper
 2   0.0772   0.0846   0.0853  \Omega_c h^2 lower
.
.
.
20   0.0307   0.2003   0.2064  H_0 upper
20   0.1509   0.2141   0.2051  H_0 lower

Raftery&Lewis statistics

chain  markov_thin  indep_thin    nburn
  1          16          21         128
  2          18          19         126
  3          32          33         192
  4          36          38         216

Parameter auto-correlations as function of step separation
        5      10      15      20      25      30      35      40      45      50      55      60      65
 1   0.739   0.527   0.384   0.288   0.210   0.132   0.078   0.053   0.043   0.028   0.022   0.027   0.021   0.
```

test.corr

Correlation matrix

0.10000E+01	0.74461E-01	0.64116E+00	0.25554E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.7
0.74461E-01	0.10000E+01	0.22552E+00	-0.30007E+00	0.00000E+00	0.00000E+00	0.00000E+00	-0.1
0.64116E+00	0.22552E+00	0.10000E+01	0.64784E-01	0.00000E+00	0.00000E+00	0.00000E+00	0.5
0.25554E+00	-0.30007E+00	0.64784E-01	0.10000E+01	0.00000E+00	0.00000E+00	0.00000E+00	0.3
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.0
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.0
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.0
0.79335E+00	-0.13260E+00	0.57301E+00	0.36580E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.1
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.0
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.0
0.41576E+00	0.37752E+00	0.34430E+00	0.73439E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.4
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.0
0.14153E+00	-0.12287E+00	0.37520E-01	0.67677E-01	0.00000E+00	0.00000E+00	0.00000E+00	-0.7
0.16243E+00	-0.95165E+00	0.72258E-01	0.33653E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.3
-0.85067E+00	0.10765E+00	-0.88437E+00	-0.24356E+00	0.00000E+00	0.00000E+00	0.00000E+00	-0.7
-0.16243E+00	0.95165E+00	-0.72258E-01	-0.33653E+00	0.00000E+00	0.00000E+00	0.00000E+00	-0.3
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.0
0.14524E+00	-0.21383E+00	0.88395E-02	0.98567E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.2
0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.0
0.37106E+00	-0.85001E+00	0.29907E+00	0.35959E+00	0.00000E+00	0.00000E+00	0.00000E+00	0.5

test.covmat

Covariance matrix

```
0.38015E-06    0.29543E-06    0.12172E-05    0.28194E-05    0.00000E+00    0.00000E+00    0.00000E+00    0.6
0.29543E-06    0.41410E-04    0.44684E-05   -0.34553E-04    0.00000E+00    0.00000E+00    0.00000E+00   -0.1
0.12172E-05    0.44684E-05    0.94806E-05    0.35694E-05    0.00000E+00    0.00000E+00    0.00000E+00    0.2
0.28194E-05   -0.34553E-04    0.35694E-05    0.32020E-03    0.00000E+00    0.00000E+00    0.00000E+00    0.9
0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.0
0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.0
0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.0
0.69231E-05   -0.12077E-04    0.24971E-04    0.92643E-04    0.00000E+00    0.00000E+00    0.00000E+00    0.2
0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.0
0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.0
0.10424E-04    0.98791E-04    0.43110E-04    0.53438E-03    0.00000E+00    0.00000E+00    0.00000E+00    0.2
0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.00000E+00    0.0
0.48807E-04   -0.44221E-03    0.64614E-04    0.67731E-03    0.00000E+00    0.00000E+00    0.00000E+00   -0.5
```

- covariance matrix for parameters
- can be used for future runs (with identical parameters) to estimate step-sizes etc.
- zeros for parameters that are not used
- sequence as in `.ini` file: $\Omega_b h^2, \Omega_m h^2, \ldots$

test.likestats

```
 Best fit sample -log(Like) =    1329.3149414062500

param  bestfit        lower1        upper1        lower2        upper2
    1  0.229689E-01  0.213392E-01  0.243634E-01  0.206308E-01  0.245422E-01   \Omega_b h^2
    2  0.109892E+00  0.953693E-01  0.122926E+00  0.905929E-01  0.131965E+00   \Omega_c h^2
    3  0.104048E+01  0.103342E+01  0.104726E+01  0.103005E+01  0.104992E+01   \theta
    4  0.847495E-01  0.502172E-01  0.132929E+00  0.366787E-01  0.151726E+00   \tau
    8  0.962479E+00  0.928906E+00  0.999629E+00  0.922190E+00  0.100942E+01   n_s
   11  0.305591E+01  0.296719E+01  0.316442E+01  0.294110E+01  0.320605E+01   log[10^{10} A_s]
   13  0.120591E+01  0.126393E-02  0.199869E+01  0.126393E-02  0.199869E+01   A_{SZ}
   14  0.743588E+00  0.672425E+00  0.808656E+00  0.628017E+00  0.822937E+00   \Omega_\Lambda
   15  0.136702E+02  0.133886E+02  0.140204E+02  0.132707E+02  0.140950E+02   Age/GYr
   16  0.256412E+00  0.191344E+00  0.327575E+00  0.177063E+00  0.371983E+00   \Omega_m
   18  0.101038E+02  0.709839E+01  0.136414E+02  0.565552E+01  0.145496E+02   z_{re}
   20  0.719830E+02  0.664313E+02  0.791095E+02  0.638385E+02  0.809112E+02   H_0
```

- Section of n-dimensional parameter space

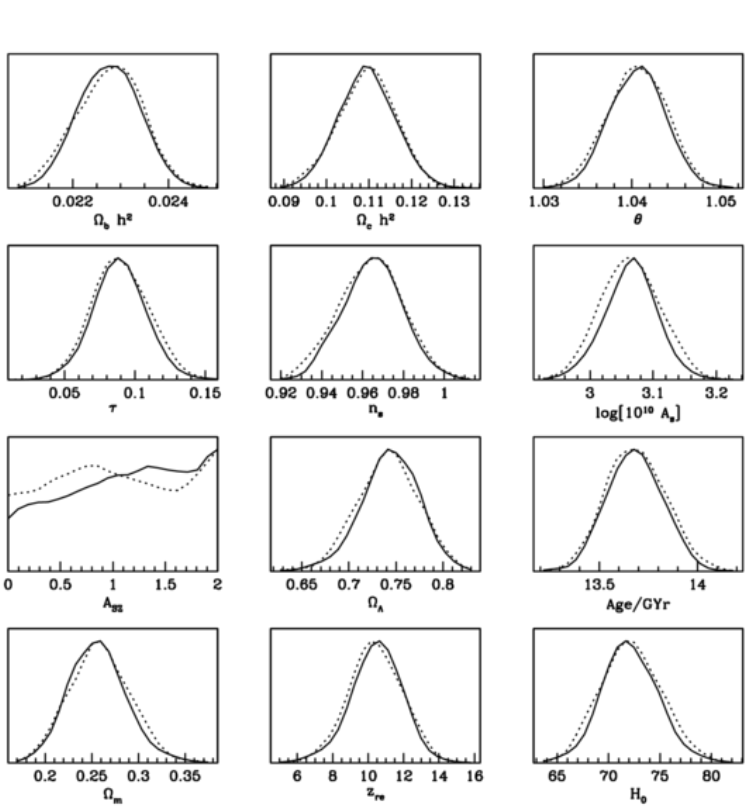
test.margestats

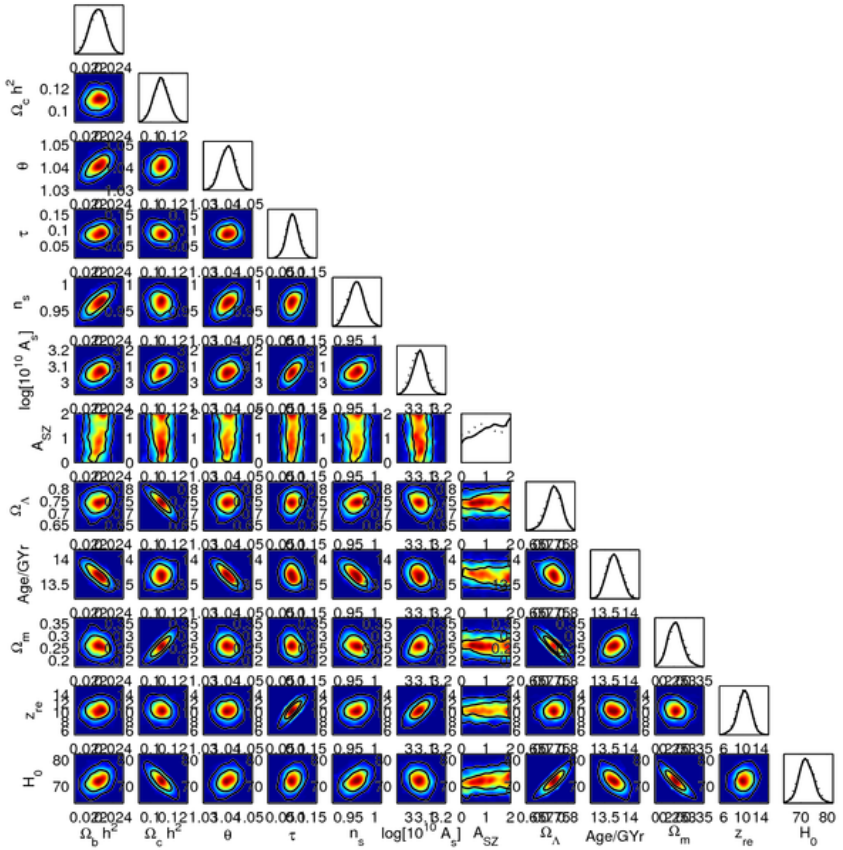
param	mean	sddev	lower1	upper1	lower2	upper2	
1	0.227596E-01	0.616564E-03	0.221280E-01	0.233856E-01	0.215866E-01	0.239445E-01	\Omega_b h^2
2	0.109523E+00	0.643506E-02	0.103008E+00	0.115962E+00	0.971503E-01	0.122237E+00	\Omega_c h^2
3	0.104060E+01	0.307906E-02	0.103750E+01	0.104358E+01	0.103464E+01	0.104655E+01	\theta
4	0.892533E-01	0.178940E-01	0.809748E-01	0.968531E-01	0.598515E-01	0.118514E+00	\tau
8	0.965029E+00	0.141533E-01	0.950608E+00	0.978894E+00	0.938597E+00	0.993358E+00	n_s
11	0.306496E+01	0.406647E-01	0.302500E+01	0.310385E+01	0.298263E+01	0.314496E+01	log[10^{10} A_s]
13	0.110708E+01	0.559296E+00	0.000000E+00	0.200000E+01	0.000000E+00	0.200000E+01	A_{SZ}
14	0.743722E+00	0.302241E-01	0.714661E+00	0.773566E+00	0.678665E+00	0.800096E+00	\Omega_\Lambda
15	0.136817E+02	0.137340E+00	0.135438E+02	0.138217E+02	0.134270E+02	0.139449E+02	Age/GYr
16	0.256278E+00	0.302241E-01	0.226437E+00	0.285352E+00	0.199929E+00	0.321341E+00	\Omega_m
18	0.104721E+02	0.142261E+01	0.913987E+01	0.118523E+02	0.736830E+01	0.131217E+02	z_{re}
20	0.721006E+02	0.268781E+01	0.695004E+02	0.747920E+02	0.669427E+02	0.777550E+02	H_0

```
Limits are: 0.68;0.95
    1 two tail;  \Omega_b h^2
    2 two tail;  \Omega_c h^2
    3 two tail;  \theta
    4 one tail;  \tau
    8 two tail;  n_s
   11 two tail;  log[10^{10} A_s]
   13 one tail;  A_{SZ}
   14 two tail;  \Omega_\Lambda
   15 two tail;  Age/GYr
   16 two tail;  \Omega_m
   18 two tail;  z_{re}
   20 two tail;  H_0
```

- “marginalized” over other parameters (i.e. weighted average)

Plots


0.022 0.024
$\Omega_b h^2$
0.09 0.1 0.11 0.12 0.13
$\Omega_c h^2$
1.03 1.04 1.05
θ
0.05 0.1 0.15
τ
0.92 0.94 0.96 0.98 1
n_s
3 3.1 3.2
$\log[10^{10} A_s]$
0 0.5 1 1.5 2
A_{SZ}
0.65 0.7 0.75 0.8
Ω_Λ
13.5 14
Age/GYr
0.2 0.25 0.3 0.35
Ω_m
6 8 10 12 14 16
z_{re}
65 70 75 80
H_0


$\Omega_b h^2$
$\Omega_c h^2$
θ
τ
n_s
$\log[10^{10} A_s]$
A_{SZ}
Ω_Λ
Age/GYr
Ω_m
z_{re}
H_0

Next time

Plotting+Visualization