Introduction to COSMOMC

SCHOOL ON COSMOLOGY TOOLS

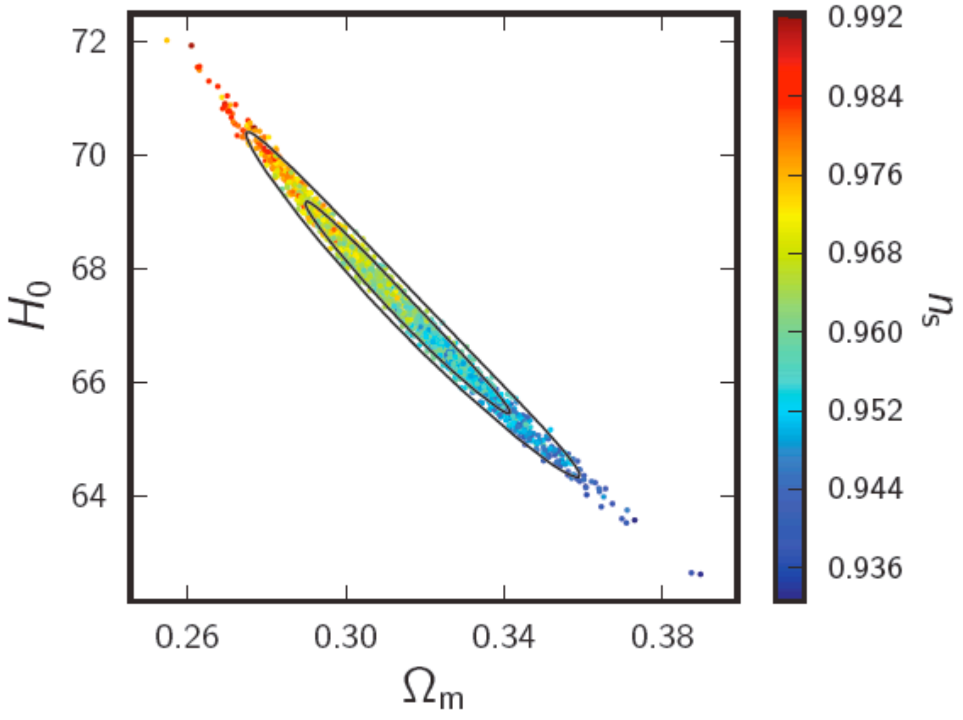
Madrid, 12-15 November 2013.

José Alberto Rubiño Martín

(Email: jalberto@iac.es)

(http://www.iac.es/galeria/jalberto)

COSMOlogical Monte-Carlo



I. *Introduction.*

II. *Structure of the code. Source files.*

III. *How to run the code.*

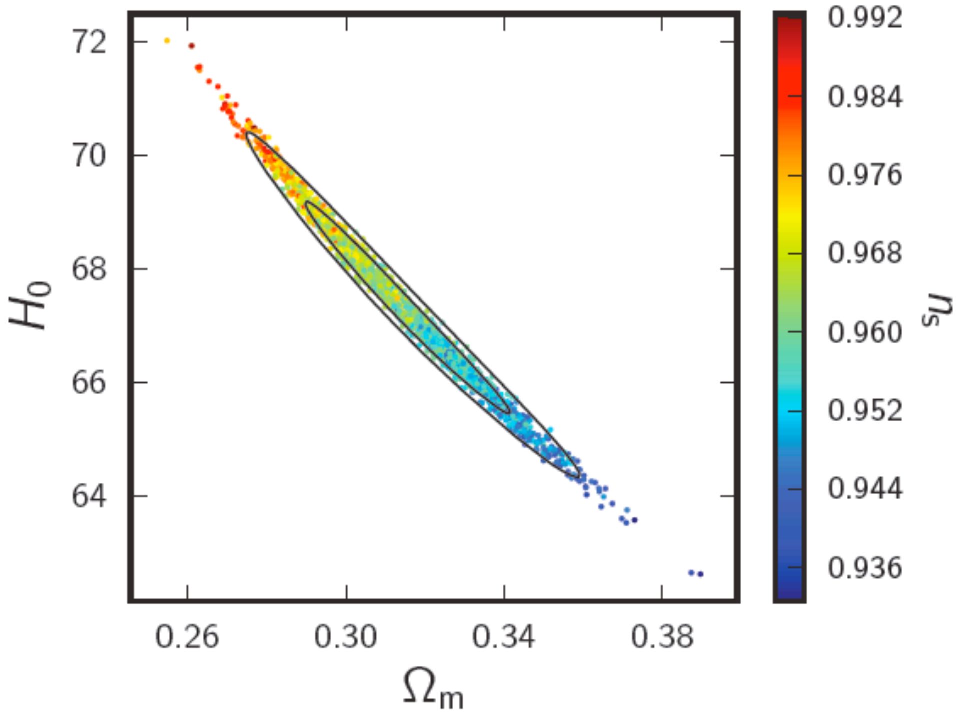
   I. *Parameterizations.*

   II. *WMAP and Planck likelihood.*

   III. *Forecasting*

IV. *Analysing the output. Convergence of the chains.*

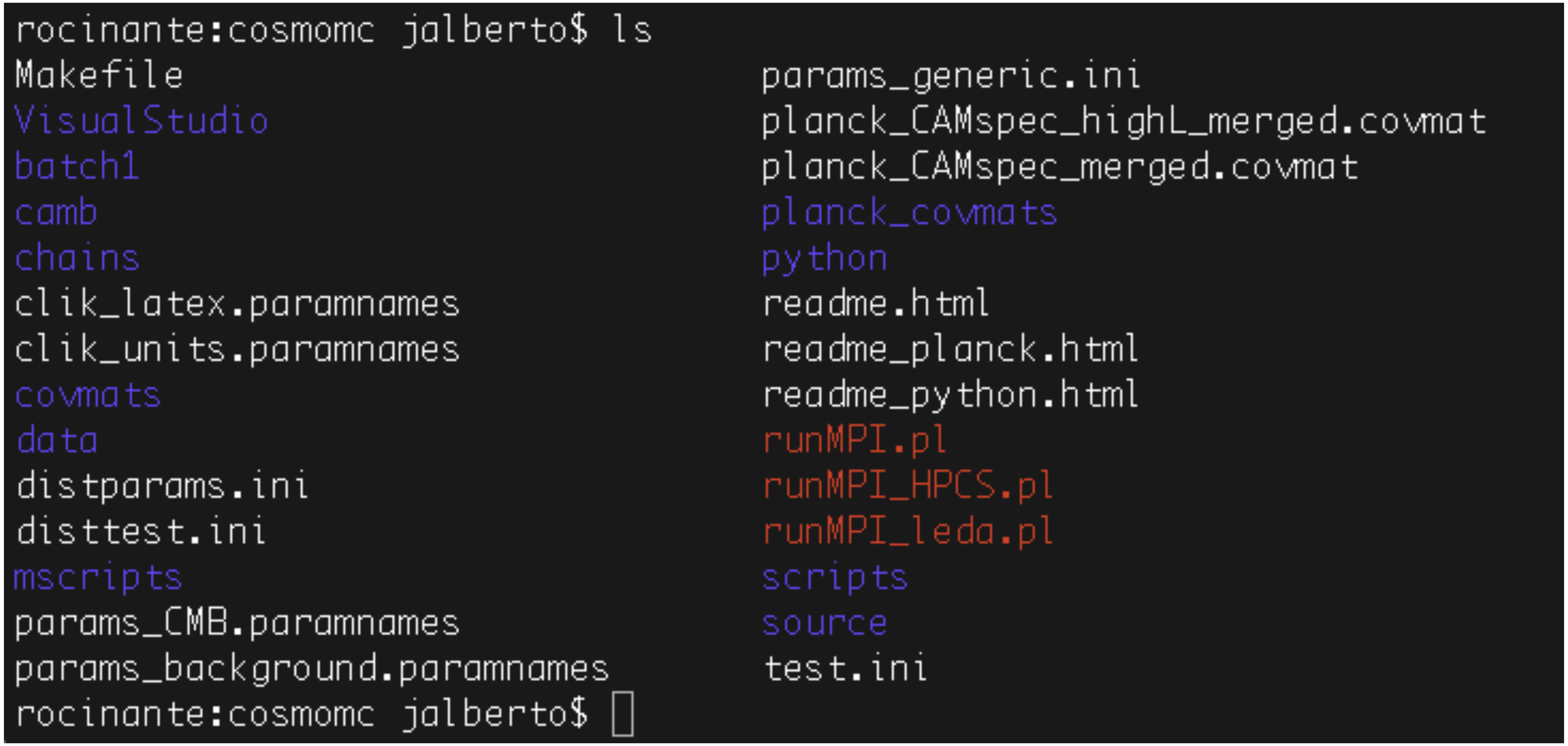
COSMOlogical Monte-Carlo



- CosmoMC is a Fortran 2003 Markov-Chain Monte-Carlo (MCMC) engine for exploring cosmological parameter space, together with code for analysing Monte-Carlo samples and importance sampling.
- Uses CAMB for the Boltzmann solver (http://camb.info)
- Author: Anthony Lewis.
- Code and documentation: http://cosmologist.info/cosmomc
- Specific forum for questions/tips and discussion: http://cosmocoffee.info/

II. Structure of the code

COSMOMC directory



```
rocinante:cosmomc jalberto$ ls
Makefile                                params_generic.ini
VisualStudio                            planck_CAMspec_highL_merged.covmat
batch1                                  planck_CAMspec_merged.covmat
camb                                    planck_covmats
chains                                  python
clik_latex.paramnames                   readme.html
clik_units.paramnames                   readme_planck.html
covmats                                 readme_python.html
data                                    runMPI.pl
distparams.ini                          runMPI_HPCS.pl
disttest.ini                            runMPI_leda.pl
mscripts                                scripts
params_CMB.paramnames                   source
params_background.paramnames            test.ini
rocinante:cosmomc jalberto$ 
```

Source files

CMB_Cls_simple.f90
DataLikelihoods.f90
EstCovmat.f90
GeneralTypes.f90
GetDist.f90
HST.f90
IO.f90
MCMC.f90
ObjectLists.f90
ParamNames.f90
Planck_like.f90
PowellConstrainedMinimize.f90
SDSSLy-a-v3.f90
bao.f90
bbn.f90
calclike.f90
cliklike.f90
cmbtypes.f90
likelihood.f90
lrggettheory.f90
lya.f90
minimize.f90
mpk.f90
params_CMB.f90
postprocess.f90
power_spec.f90
propose.f90
samples.f90
settings.f90
supernovae.f90
supernovae_SNLS.f90
supernovae_Union2.f90
wigglez.f90
Matrix_utils.F90
cmbdata.F90
driver.F90
paramdef.F90

Source files

CMB_Cls_simple.f90
DataLikelihoods.f90
EstCovmat.f90
GeneralTypes.f90
GetDist.f90
HST.f90
IO.f90
MCMC.f90
ObjectLists.f90
ParamNames.f90
Planck_like.f90
PowellConstrainedMinimize.f90
SDSSLy-a-v3.f90
bao.f90
bbn.f90
calclike.f90
cliklike.f90
cmbtypes.f90
likelihood.f90
lrggettheory.f90
lya.f90
minimize.f90
mpk.f90
params_CMB.f90
postprocess.f90
power_spec.f90
propose.f90
samples.f90
settings.f90
supernovae.f90
supernovae_SNLS.f90
supernovae_Union2.f90
wigglez.f90
Matrix_utils.F90
cmbdata.F90
driver.F90
paramdef.F90

params_CMB.f90
This defines what the input variables mean. Change this to use different variables. You can change which parameterization file to use in the Makefile.

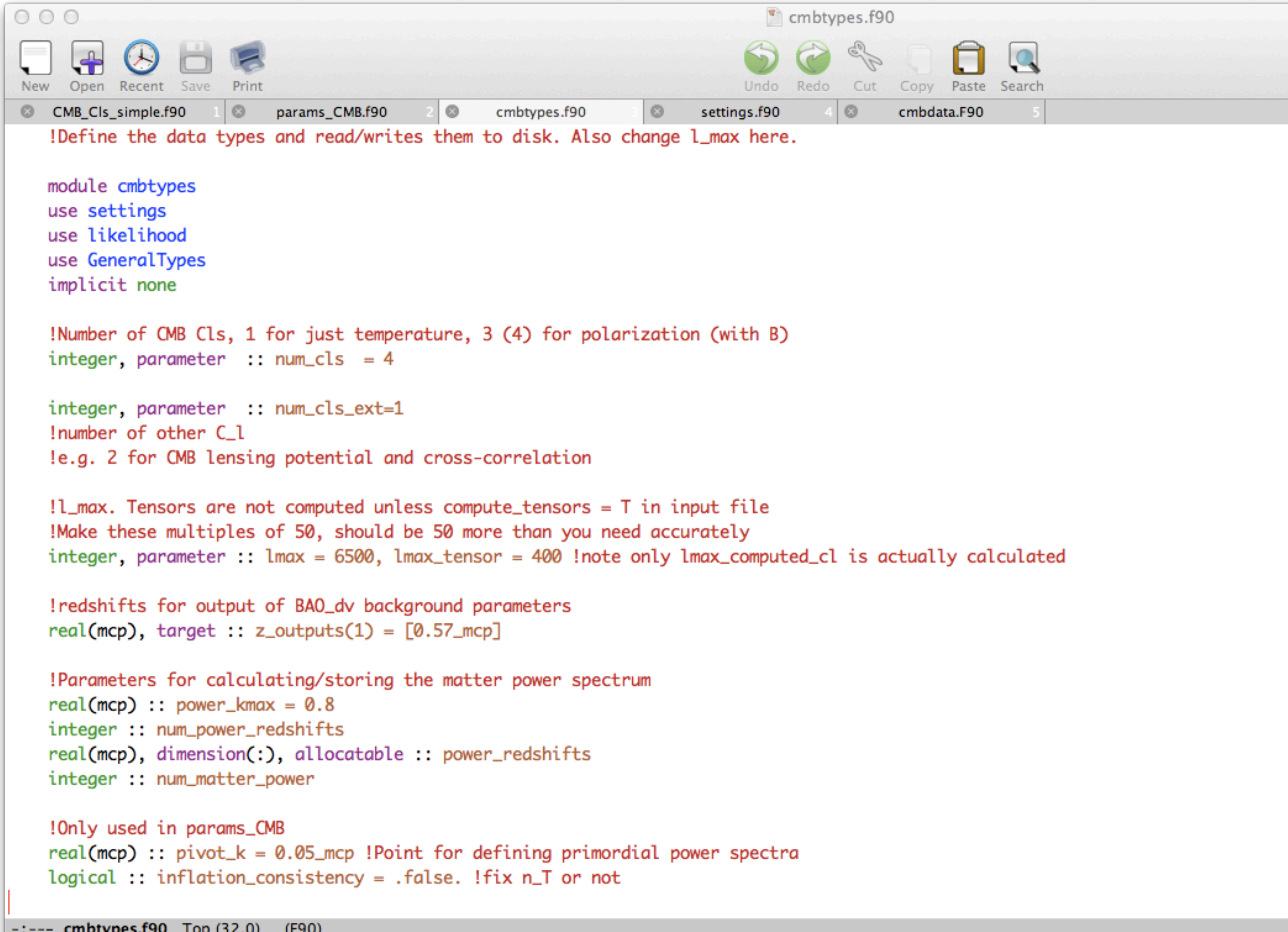
cmbtypes.f90
You can also change the num_cls number of (temperature plus polarization) Cls to compute and store, power spectrum parameter, etc.

settings.f90
This defines the maximum number of parameters and their types.

cmbdata.f90
This reads in the CMB .dataset information and computes likelihoods. You may wish to edit this, for example to use likelihood distributions for the band powers, or to compute the likelihood from actual polarized data. This version assumes polarized data points are an arbitrary combination of the raw TT, TE, EE, and BB Cls, as specified in the window files in data/windows. WMAP data is handled as a special case. Also all_exact case.

CMB_Cls_simple.f90
Routines for generating Cls, matter power spectra and sigma8 from CAMB. Replace this file to use other generators, e.g. a fast approximator like CMBfit, DASH, PICO, etc.

calclike.f90
Calls the data likelihood functions, etc.



```
!Define the data types and read/writes them to disk. Also change l_max here.

module cmbtypes
use settings
use likelihood
use GeneralTypes
implicit none

!Number of CMB Cls, 1 for just temperature, 3 (4) for polarization (with B)
integer, parameter  :: num_cls  = 4

integer, parameter  :: num_cls_ext=1
!number of other C_l
!e.g. 2 for CMB lensing potential and cross-correlation

!l_max. Tensors are not computed unless compute_tensors = T in input file
!Make these multiples of 50, should be 50 more than you need accurately
integer, parameter :: lmax = 6500, lmax_tensor = 400 !note only lmax_computed_cl is actually calculated

!redshifts for output of BAO_dv background parameters
real(mcp), target :: z_outputs(1) = [0.57_mcp]

!Parameters for calculating/storing the matter power spectrum
real(mcp) :: power_kmax = 0.8
integer :: num_power_redshifts
real(mcp), dimension(:), allocatable :: power_redshifts
integer :: num_matter_power

!Only used in params_CMB
real(mcp) :: pivot_k = 0.05_mcp !Point for defining primordial power spectra
logical :: inflation_consistency = .false. !fix n_T or not
```

```
-:--- cmbtypes.f90   Top (32,0)    (F90)
```

III. Running the code

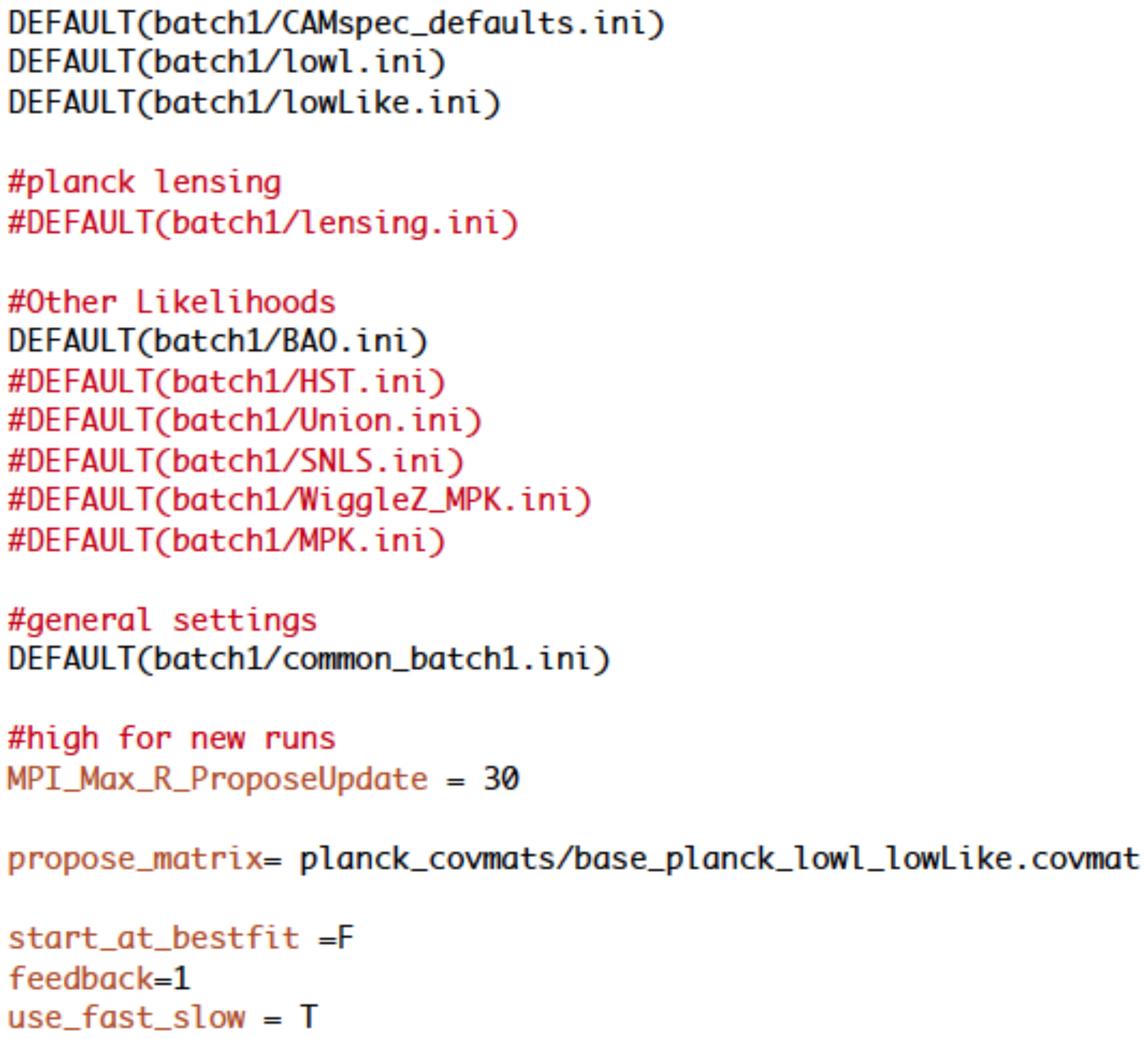
Serial mode:

```
> ./cosmomc test.ini 1
```

MPI mode:

```
> mpirun –np 2 ./cosmomc test.ini
```

COSMOMC ini-file: test.ini



```
DEFAULT(batch1/CAMspec_defaults.ini)
DEFAULT(batch1/lowl.ini)
DEFAULT(batch1/lowLike.ini)

#planck lensing
#DEFAULT(batch1/lensing.ini)

#Other Likelihoods
DEFAULT(batch1/BAO.ini)
#DEFAULT(batch1/HST.ini)
#DEFAULT(batch1/Union.ini)
#DEFAULT(batch1/SNLS.ini)
#DEFAULT(batch1/WiggleZ_MPK.ini)
#DEFAULT(batch1/MPK.ini)

#general settings
DEFAULT(batch1/common_batch1.ini)

#high for new runs
MPI_Max_R_ProposeUpdate = 30

propose_matrix= planck_covmats/base_planck_lowl_lowLike.covmat

start_at_bestfit =F
feedback=1
use_fast_slow = T
```

COSMOMC ini-file: test.ini

```
#sampling_method=7 is a new fast-slow scheme good for Planck
sampling_method = 7
dragging_steps  = 3
propose_scale = 2

indep_sample=0

use_clik=T

#Folder where files (chains, checkpoints, etc.) are stored
root_dir = chains/

#Root name for files produced
file_root=test
action = 0

#these are just small speedups for testing
get_sigma8=F

#Uncomment this if you don't want one 0.06eV neutrino by default

#num_massive_neutrinos=3
#param[mnu] = 0 0 0 0 0
```

Cosmomc has 7 different sampling methods. Number 1 is the Metropolis-Hasting scheme

batch1/common_batch1.ini

```
##Sample file of common parameters for baseline Planck set of runs

batch_name = batch1

local_dir = %LOCALDIR%
#directory, e.g. window functions in directory windows under data_dir
data_dir = %LOCALDIR%data/

INCLUDE(likelihood_batch1.ini)
INCLUDE(params_CMB_defaults.ini)

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

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

#Temperature at which to Monte-Carlo
temperature = 1

#Maximum number of chain steps
samples = 4000000
```

To sample **from** P^(1/T) rather than P (good for the tails of P, the posterior distribution)

batch1/common_batch1.ini

```
#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 = 1.7
```

```
#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 chains
#Set to a negative number not to use this feature. Does not guarantee
good accuracy of confidence limits.
MPI_Converge_Stop = 0.02
```

(Gelman and Rubin R statistic)

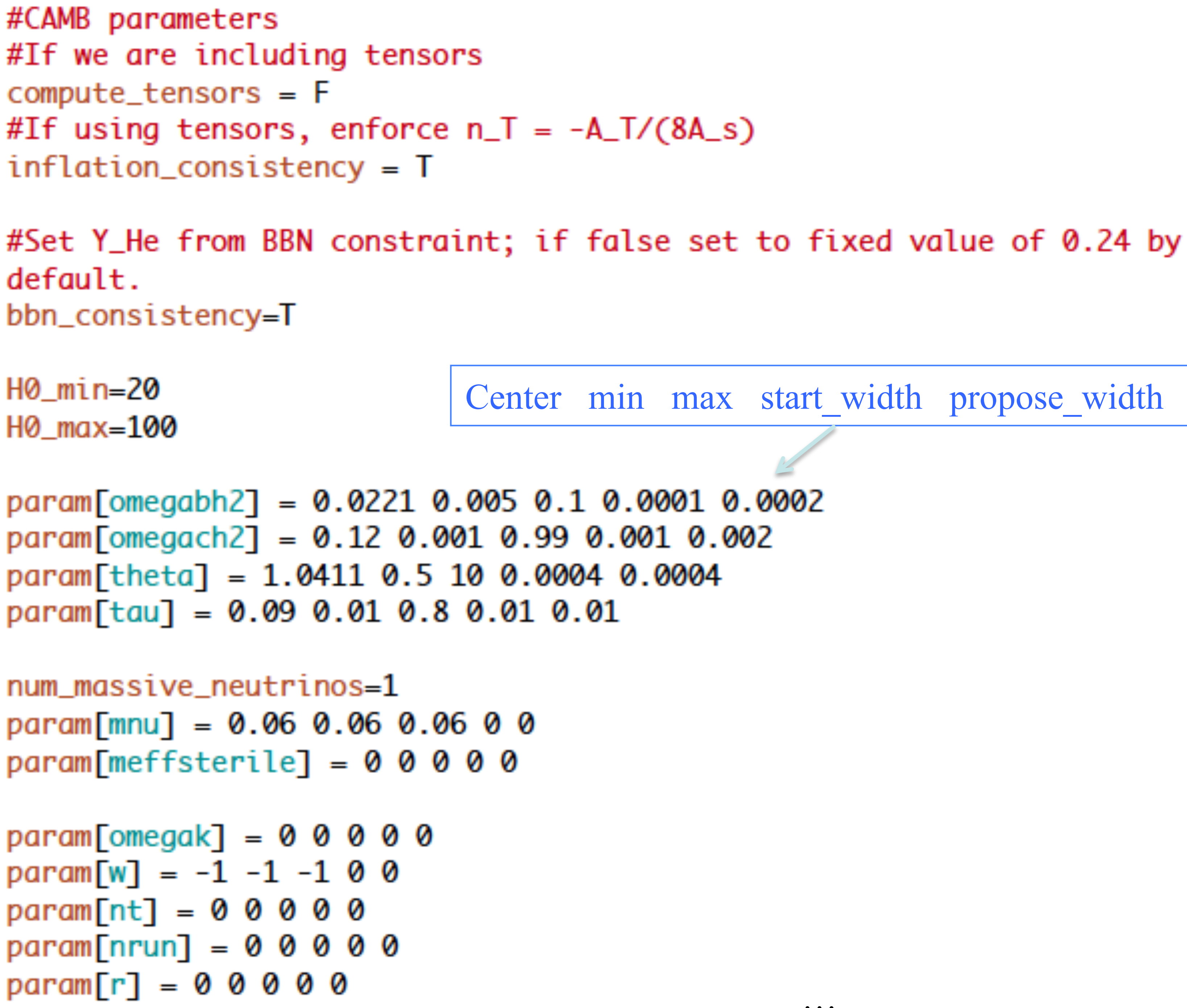
```
#1: Simple Metropolis, 2: slice sampling, 3: slice sampling fast
parameters, 4: directional gridding
#7 is new dragging method
sampling_method = 7

dragging_steps  = 3
use_fast_slow = T
```

batch1/likelihood_batch1.ini

```
use_CMB = T
use_clik = T

use_HST = F
use_mpk = F
use_SN = F
use_BAO=F

use_lya = F
use_clusters = F
use_min_zre = 0

#filenames for matter power spectrum datasets, incl twodf
mpk_numdatasets = 1
mpk_dataset1 = %DATASETDIR%sdss_lrgDR4.dataset


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

use_Age_Tophat_Prior = T
```

This controls of the info that you want to put inside your likelihood function.

batch1/params_CMB_defaults.ini

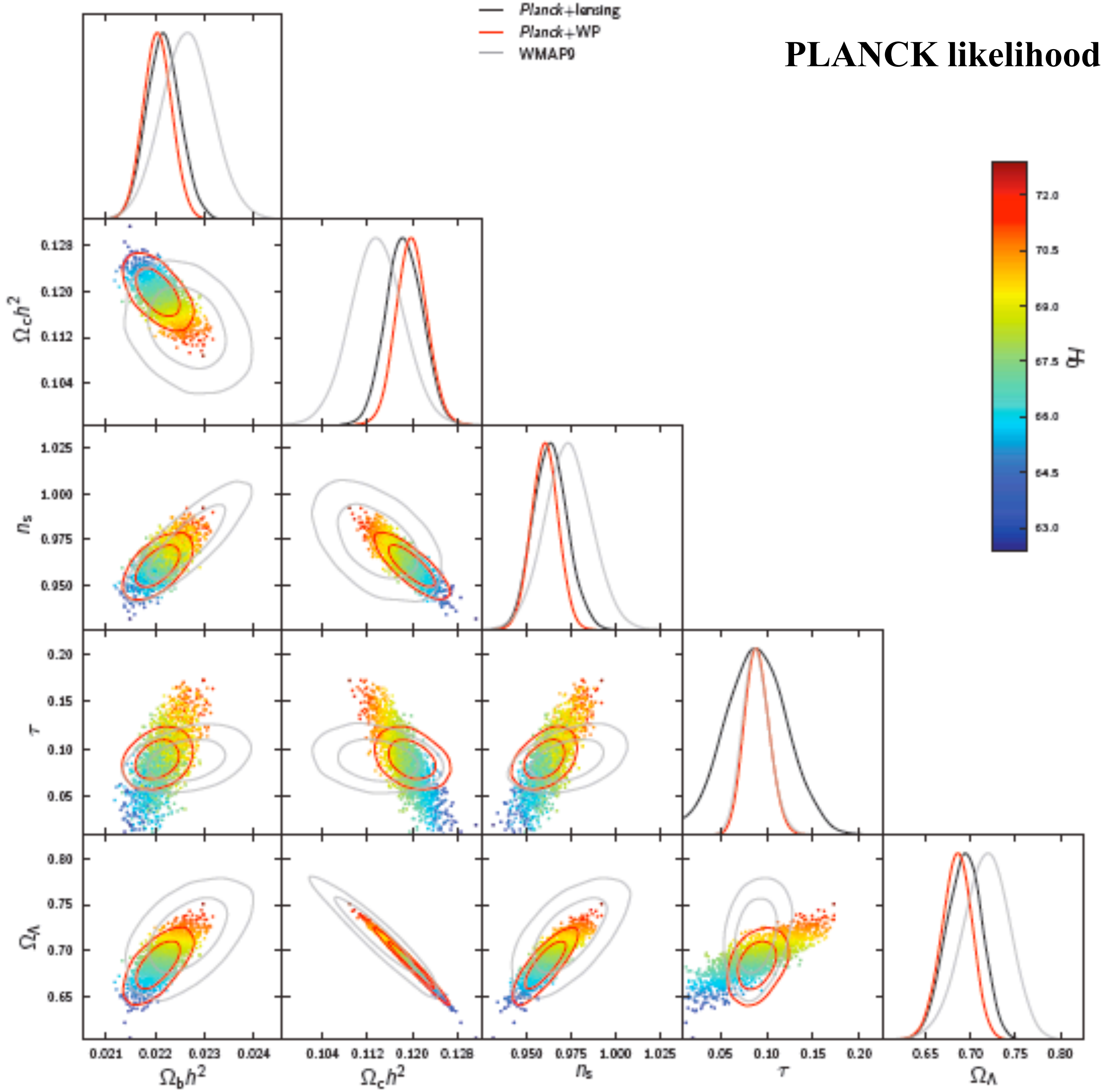


```
#CAMB parameters
#If we are including tensors
compute_tensors = F
#If using tensors, enforce n_T = -A_T/(8A_s)
inflation_consistency = T

#Set Y_He from BBN constraint; if false set to fixed value of 0.24 by
default.
bbn_consistency=T

H0_min=20
H0_max=100

param[omegabh2] = 0.0221 0.005 0.1 0.0001 0.0002
param[omegach2] = 0.12 0.001 0.99 0.001 0.002
param[theta] = 1.0411 0.5 10 0.0004 0.0004
param[tau] = 0.09 0.01 0.8 0.01 0.01

num_massive_neutrinos=1
param[mnu] = 0.06 0.06 0.06 0 0
param[meffsterile] = 0 0 0 0 0

param[omegak] = 0 0 0 0 0
param[w] = -1 -1 -1 0 0
param[nt] = 0 0 0 0 0
param[nrun] = 0 0 0 0 0
param[r] = 0 0 0 0 0
```

Center   min   max   start_width   propose_width

…

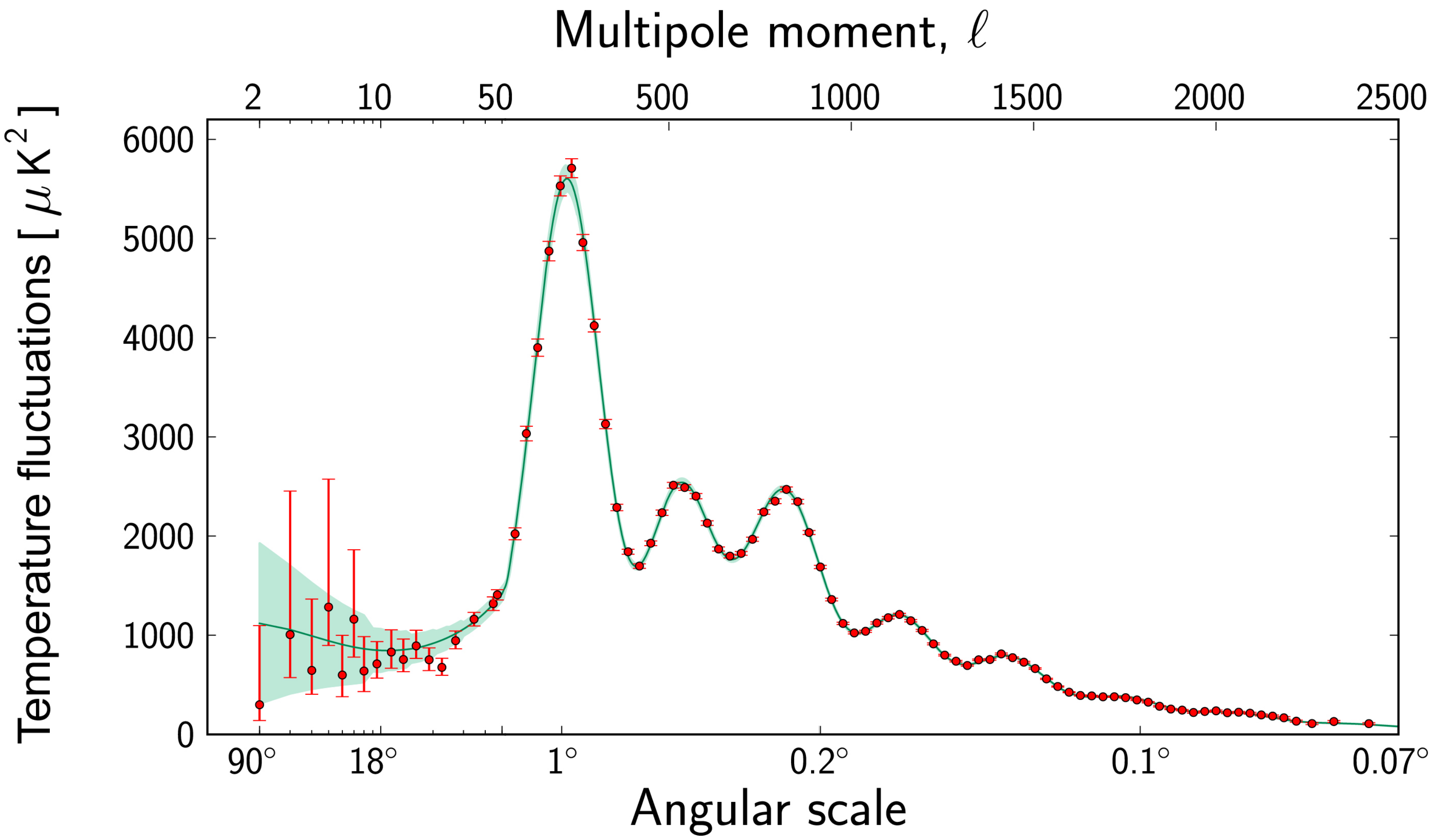
PLANCK likelihood



Likelihood Methodology

- Hybrid multi-frequency likelihood approach
  - *Large scales: map-based Gaussian likelihood*
  - *Small scales: Gaussian likelihood approximation on spectra*
- Marginalization over foregrounds
  - *Large scales: Gibbs marginalization (map level)*
  - *Small scales: Parameterized at the spectrum level*
- Validation
  - *Data selection*
  - *Null tests*
  - *Simulations*
  - *Foreground cleaned CMB maps*

Planck power spectrum (TT)



batch1/CAMspec_defaults.ini

```
#clik likelihood file, when compiling with cliklike
clik_data_camspec =
%DATASETDIR%clik/CAMspec_v6.2TN_2013_02_26_dist.clik

clik_params_camspec = %DATASETDIR%camspec.paramnames
clik_speed_camspec = 5

lmax_computed_cl = 2500

#this contains most of the standard parameters so can be used with
various different runs (but in practice is often not invertible)
#lots of full covmats are provided in planck_covmats
propose_matrix = planck_CAMspec_merged.covmat

#CAMspec nuisance parameters

param[aps100] = 153 0 360 27 27
param[aps143]= 54.9 0 270 4.5 4.5
param[aps217]= 55.8 0 450 7.2 7.2
param[acib143]= 4 0 20 3 3
param[acib217]= 55.5 0 80 3 3
param[asz143]= 4 0 10 1 1
param[psr]= 0.91 0.0 1.0 0.04 0.04
param[cibr]= 0.63 0.0 1.0 0.05 0.05
param[ncib] = 0.6 -2 2 0.05 0.05
param[cal0]= 1 0.98 1.02  0.0004 0.0004
param[cal2]= 1 0.95 1.05 0.001 0.001
param[xi] = 0.1 0 1 .2 .2
```

Parameterizations

- **omegabh2** - the physical baryon density
- **omegach2** - the physical dark matter density
- **theta** - 100*(the ratio of the [approx] sound horizon to the angular diameter distance)
- **tau** - the reionization optical depth
- **omegak** - omega_K
- **mnu** - the sum of the neutrino masses (in eV)
- **nnu** - the effective density parameter for neutrinos N_{eff}
- **w** - the (assumed constant) equation of state of the dark energy (taken to be quintessence)
- **ns** - the scale spectral index
- **nt** - the tensor spectral index
- **nrun** - the running of the scalar spectral index
- **logA** - ln[10^10 A_s] where A_s is the primordial superhorizon power in the curvature perturbation on 0.05Mpc^{-1} scales (i.e. in this is an amplitude parameter)
- **r** - the ratio A_t/A_s, where A_t is the primordial power in the transverse traceless part of the metric tensor

params_CMB.paramnames

III. Analysing the output of COSMOMC.

After running COSMOMC:

- Output files in *chains/*

- *.txt* files contain the chains. The format is:

  weight likelihood param1 param2 param3 ...

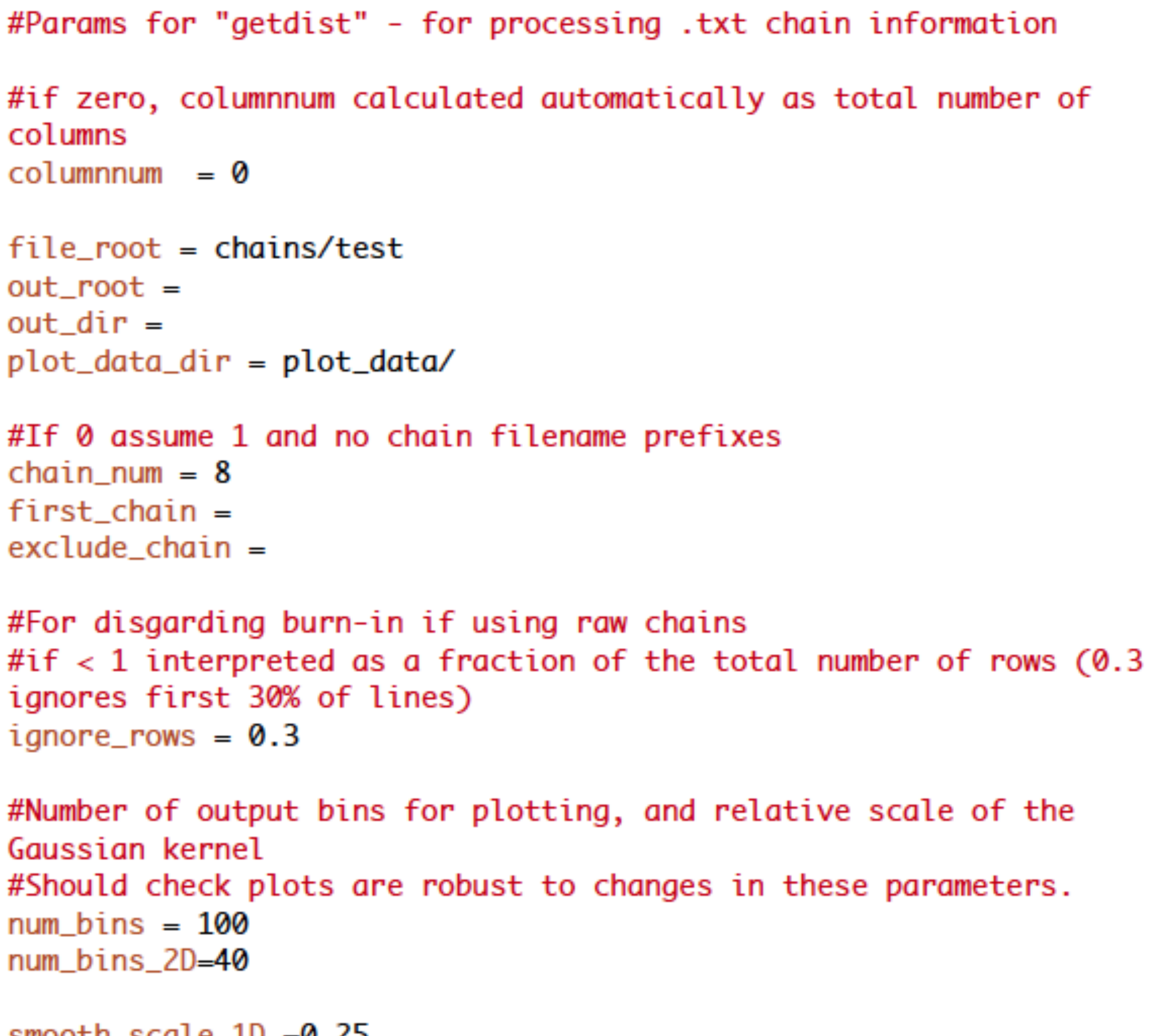
- *.log* files contain STDOUT from each chain

- If running under MPI, STDOUT files are in stdout/

But you need to do post-processing of the chains (thinning, burn-in period) and study convergence → *getdist*.

```
> ./getdist distparams.ini
```

distparams.ini



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

#if zero, columnnum calculated automatically as total number of
columns
columnnum  = 0

file_root = chains/test
out_root =
out_dir =
plot_data_dir = plot_data/

#If 0 assume 1 and no chain filename prefixes
chain_num = 8
first_chain =
exclude_chain =

#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 = 0.3

#Number of output bins for plotting, and relative scale of the
Gaussian kernel
#Should check plots are robust to changes in these parameters.
num_bins = 100
num_bins_2D=40

smooth_scale_1D =0.25
```

```
#if T produced B&W printer friendly output
B&W = F

plot_ext = py
```

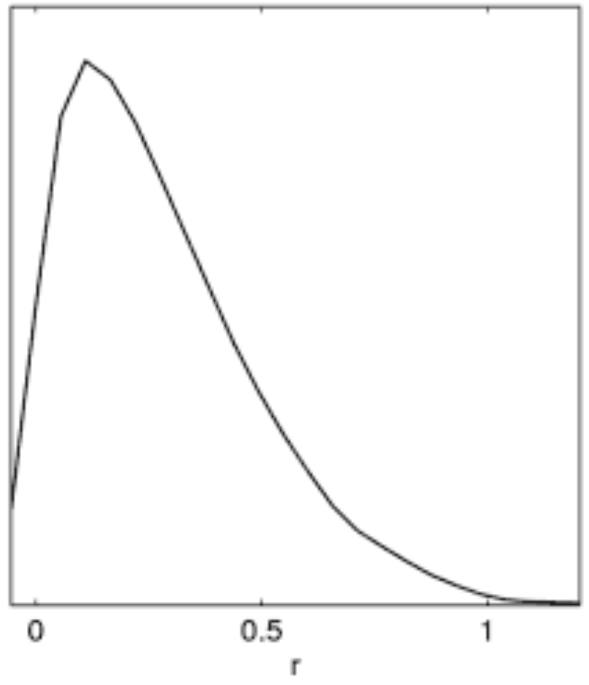
(use the option “m” for Matlab)

```
plot_meanlikes = T
shade_meanlikes = T

# 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
```

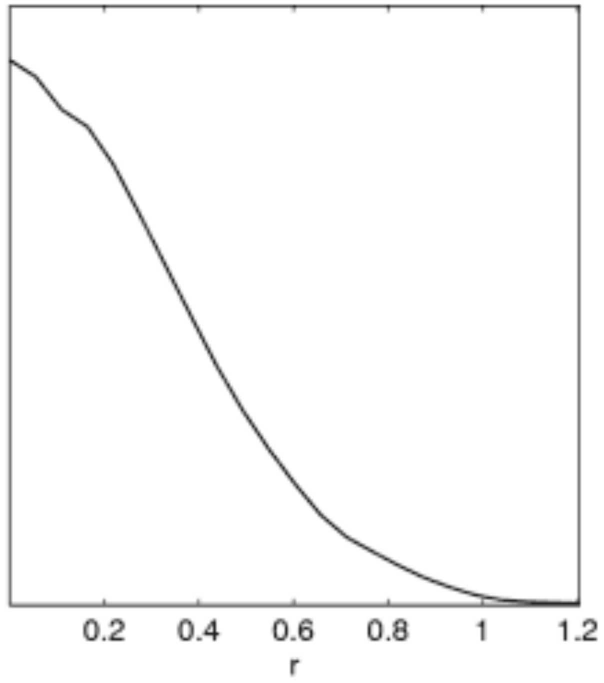
```
#if both zero it will plot most correlated variables
plot_2D_num = 0
plot1 = ns omegabh2
plot2 =
```

```
#Output 2D plots for param combos with 1D marginalized plots along the
diagonal
triangle_plot = T
triangle_params = omegabh2 omegach2 tau omegak mnu nnu yhe Alens ns
nrun logA r H0 omegam omegal sigma8 r02
```

```
#Need to give limits if prior cuts off distribution where not very
small
limits[r02]= 0 N
limits[r10]= 0 N
```



Incorrect result when **limits[r02]** is not set.



Correct result when setting **limits[r02]=0 N**.

➢ Convergence diagnostics are generated in the same directory, with files named **.converge, *.likestats, *.margestats, *.covmat* , and **.corr*

file_root.margestats file contains the means, standard deviations and marginalized limits for the different parameters

file_root.likestats gives the best fit sample model, its likelihood, and limits from the extremal values of the N-dimensional distribution.

file_root.converge contains various convergence diagnostics

file_root.corr contains parameter correlations

file_root.covmat contains a covariance matrix you can use as a proposal matrix for generating future chains

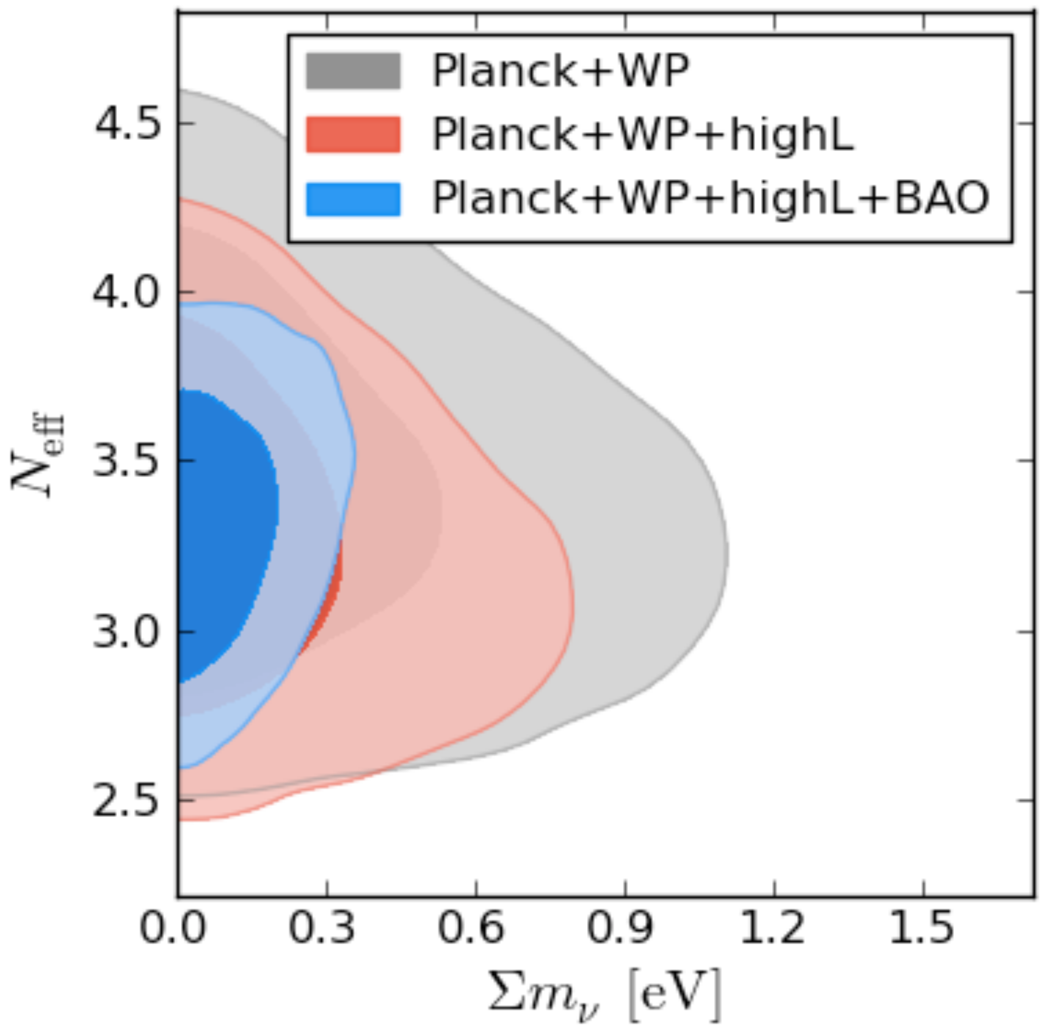
Convergence diagnostics

- **Gelman and Rubin** "variance of chain means"/"mean of chain variances" R statistic. See lecture of Licia Verde for the definitions. This statistic can be computed if you have multiple chains. Typically you want the value to be less than 0.2.

- For individual chains, getdist computes the **Raftery and Lewis** convergence diagnostics. This uses a binary chain derived from each parameter depending on whether the parameter is above or below a given percentile of its distribution. It is basically a criterion for accuracy of the estimation of a certain quantile q. It also assesses the thin factor needed for the binary chain to approximate an independence chain.

- There are other statistics included in gestdist (see documentation for details).

Now you can start running the code to produce plots like this one:



Exercise 4.

1. Run a test case in serial mode to show that it is working.
2. Run a full case of a likelihood of an ideal experiment (forecasting).

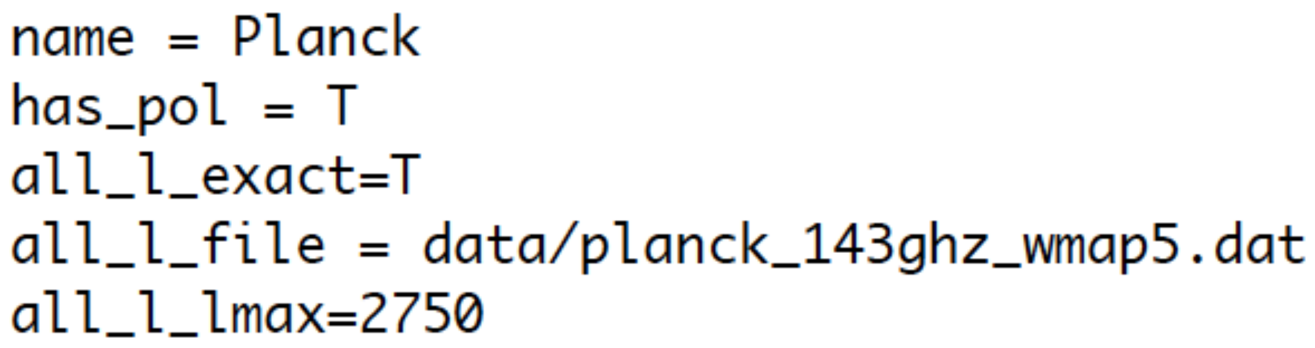
Note: For exercise 2, you will need to use the *all_l_exact* approach for the likelihood. See this post: http://cosmocoffee.info/viewtopic.php?t=231 for detailed information. It uses the full-sky (exact) likelihood given by (Lewis 2005)

$$-2\log P(\hat{C}_l|C_l) = (2l{+}1)\left\{\mathrm{Tr}\left[\hat{C}_l C_l^{-1}\right] + \log|C_l|\right\}$$

forecast.ini

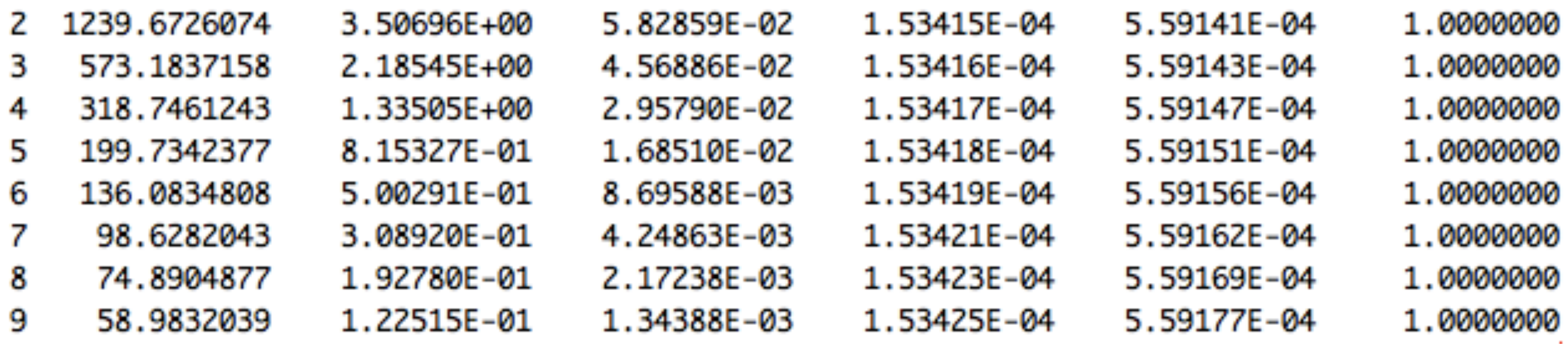
```
cmb_numdatasets = 1
cmb_dataset1 = data/planck_143ghz_wmap5.dataset

lmax_computed_cl = 1500
use_fast_slow = F
sampling_method = 1
```

data/planck_143ghz_wmap5.dataset

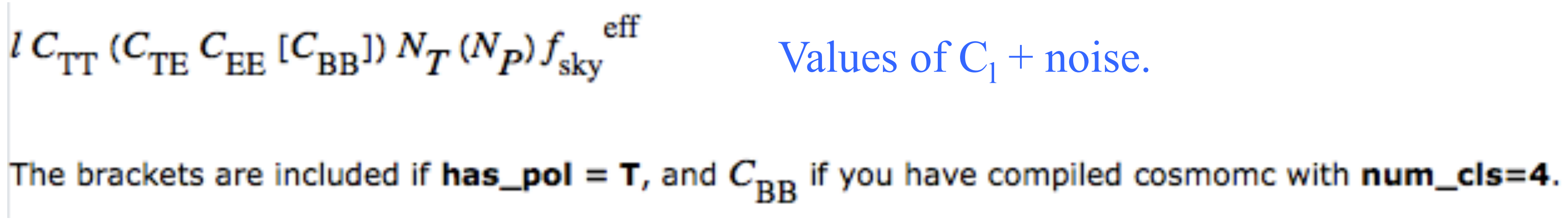


```
name = Planck
has_pol = T
all_l_exact=T
all_l_file = data/planck_143ghz_wmap5.dat
all_l_lmax=2750
```

data/planck_143ghz_wmap5.dat



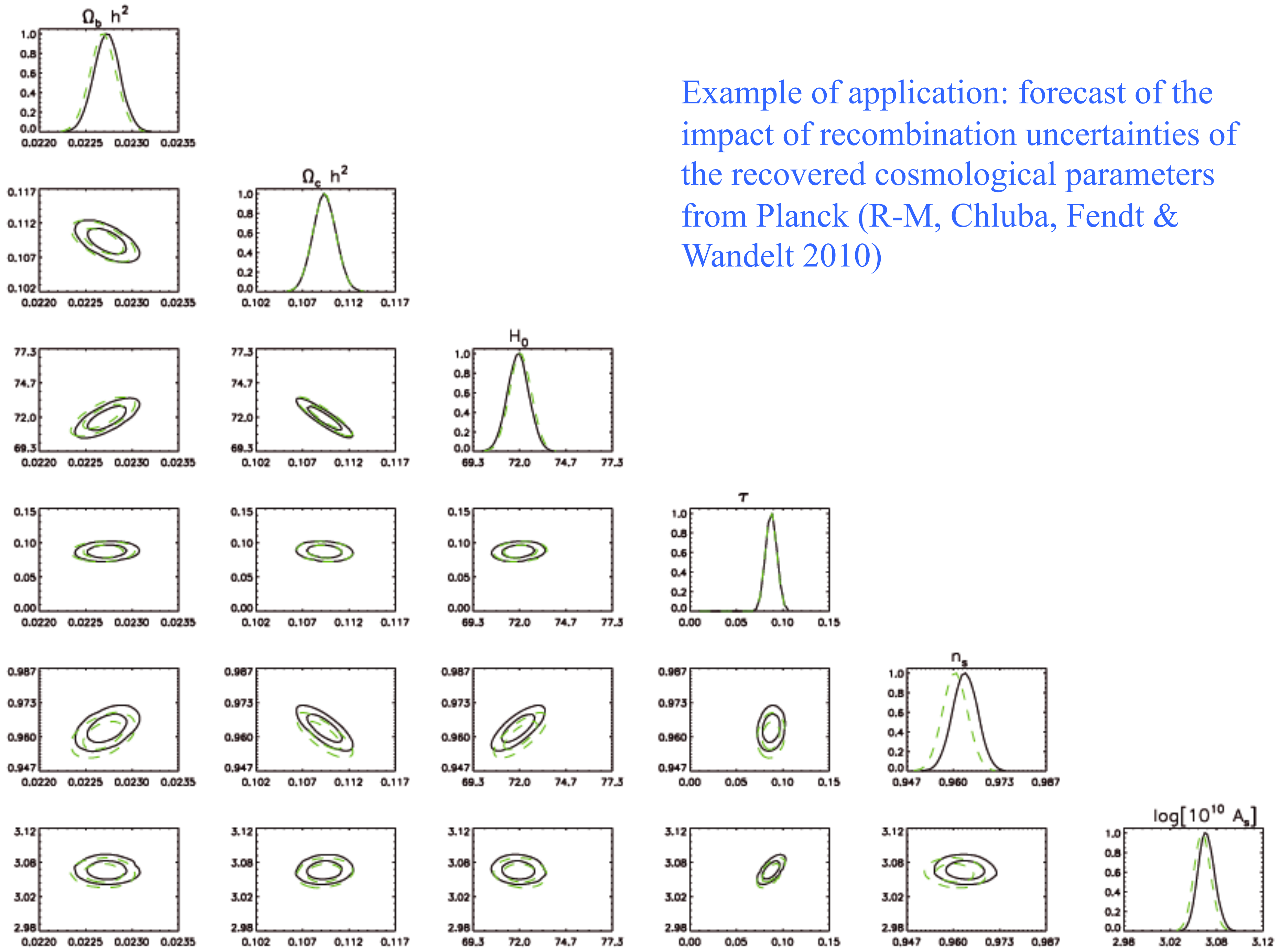
```
2  1239.6726074    3.50696E+00    5.82859E-02    1.53415E-04    5.59141E-04     1.0000000
3   573.1837158    2.18545E+00    4.56886E-02    1.53416E-04    5.59143E-04     1.0000000
4   318.7461243    1.33505E+00    2.95790E-02    1.53417E-04    5.59147E-04     1.0000000
5   199.7342377    8.15327E-01    1.68510E-02    1.53418E-04    5.59151E-04     1.0000000
6   136.0834808    5.00291E-01    8.69588E-03    1.53419E-04    5.59156E-04     1.0000000
7    98.6282043    3.08920E-01    4.24863E-03    1.53421E-04    5.59162E-04     1.0000000
8    74.8904877    1.92780E-01    2.17238E-03    1.53423E-04    5.59169E-04     1.0000000
9    58.9832039    1.22515E-01    1.34388E-03    1.53425E-04    5.59177E-04     1.0000000
```



$l\ C_{TT}\ (C_{TE}\ C_{EE}\ [C_{BB}])\ N_T\ (N_P)\ f_{sky}^{eff}$

Values of C_l + noise.

The brackets are included if **has_pol = T**, and C_{BB} if you have compiled cosmomc with **num_cls=4**.



Example of application: forecast of the impact of recombination uncertainties of the recovered cosmological parameters from Planck (R-M, Chluba, Fendt & Wandelt 2010)