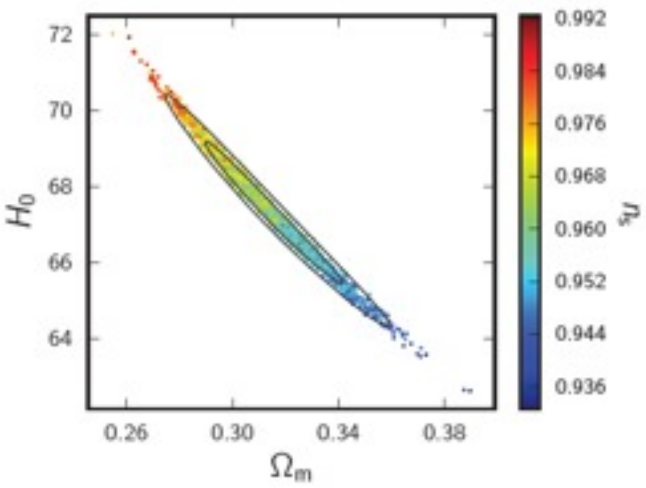
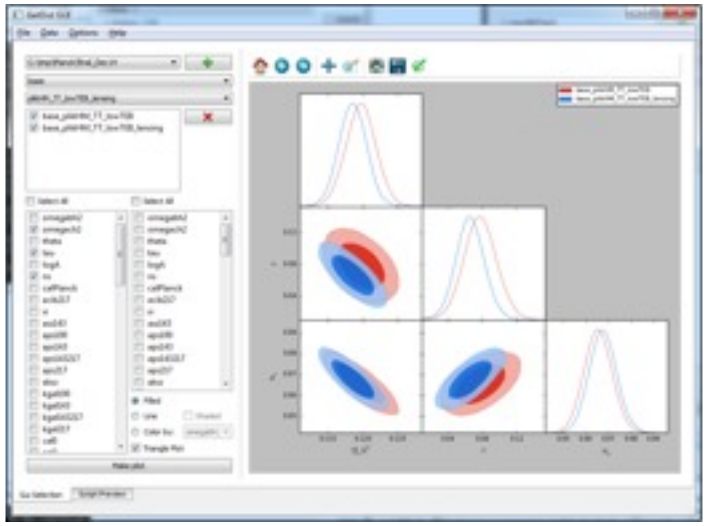




Introduction to CosmoMC

Part I: Motivation & Basic concepts

Antonio J. Cuesta
Institut de Ciències del Cosmos - Universitat de Barcelona

Institut de Ciències del Cosmos
EXCELENCIA MARÍA DE MAEZTU
UNIVERSITAT DE BARCELONA

Dept. de Física Teórica y del Cosmos, Universidad de Granada, 1-3 Marzo 2016

What is this course (and what isn't)

- This course IS a guide to let you use the basic features of CosmoMC
- We WILL learn how to set up the code run a MCMC, and analyze the outputs
- We WILL learn how to customize the code to our needs via parameter files

- This is NOT a course about CAMB or any Boltzmann evolution solvers
- We will NOT add any fluid component to the existing ones in CAMB
- We will NOT describe in detail the installation process
- We will NOT modify the source code

Outline of Part I

- Statistical methods: Bayesians vs. frequentists
- Degeneracies and Marginalization
- The standard LCDM model: the 6 base parameters
- Beyond the standard LCDM model
- Cosmological datasets
- CAMB and CosmoMC
- Parameter estimation with Markov Chain Monte Carlo

Bayesians vs. Frequentists

- Basically two philosophical **interpretations** of probabilities in statistics
- Frequentists: probabilities are long-term **frequencies** (ratios) of events, assuming the experiments can be repeated in **identical** conditions a **large** number of times. Probability is defined as a limit when N->infinity
  - Pros: **objective** definition: there is no room for prejudices.
  - Cons: in practice you can only afford to do **1 experiment**.

Bayesians vs. Frequentists

- Bayesians: probabilities are just **degrees of certainty**. This definition can also allow to define the probability of a given parameter value, or of a model
  - Pros: this definition can be applied even for a **single** experiment. It is also a **flexible** definition: it allows to include external beliefs (**priors**)
  - Cons: probabilities are **no longer objective** quantities - every person can choose a different prior, which changes the final result (the posterior)
- Confidence intervals: **Bayesians** would say that the true value of a parameter is within a credible region with a 95% probability. **Frequentists** would say that if the experiment is repeated many times, the confidence interval would contain the true value in 95% of those cases

When using this code, we are BAYESIANS

The Bayes Theorem

- $$P(F|D) = \frac{P(D|F)\ P(F)}{P(D)}$$

- Here P(D|F) is the *likelihood*, P(F) is the *model prior*, P(D) is the *model evidence*, and P(F|D) is the *posterior*.

- **Frequentists** only worry about **P(D|F)**. **Bayesians** study instead **P(F|D)**.

- Since P(D) is usually considered as a *normalization* factor, the main difference between bayesian and frequentists is the inclusion of a *external* **prior**.

The Bayes Theorem

- $$P(F|D) = \frac{P(D|F)\ P(F)}{P(D)}$$

- In order to minimize the effect of the prior one uses "**uninformative**" priors. These priors are designed to have a low impact by assuming e.g., a **flat distribution** on a parameter x, i.e. P(x)=*constant*

- Even in that case, you **are** introducing some information. Another person is allowed to say that the correct choice is P(log x)=const, which is of course different: P(log x)=P(x)dx/dlogx=xP(x). There is no "correct" choice, but sometimes one choice is better motivated than others

In CosmoMC, the prior is introduced by choosing the variable you want to sample, e.g. A_s vs. $\log(A_s)$

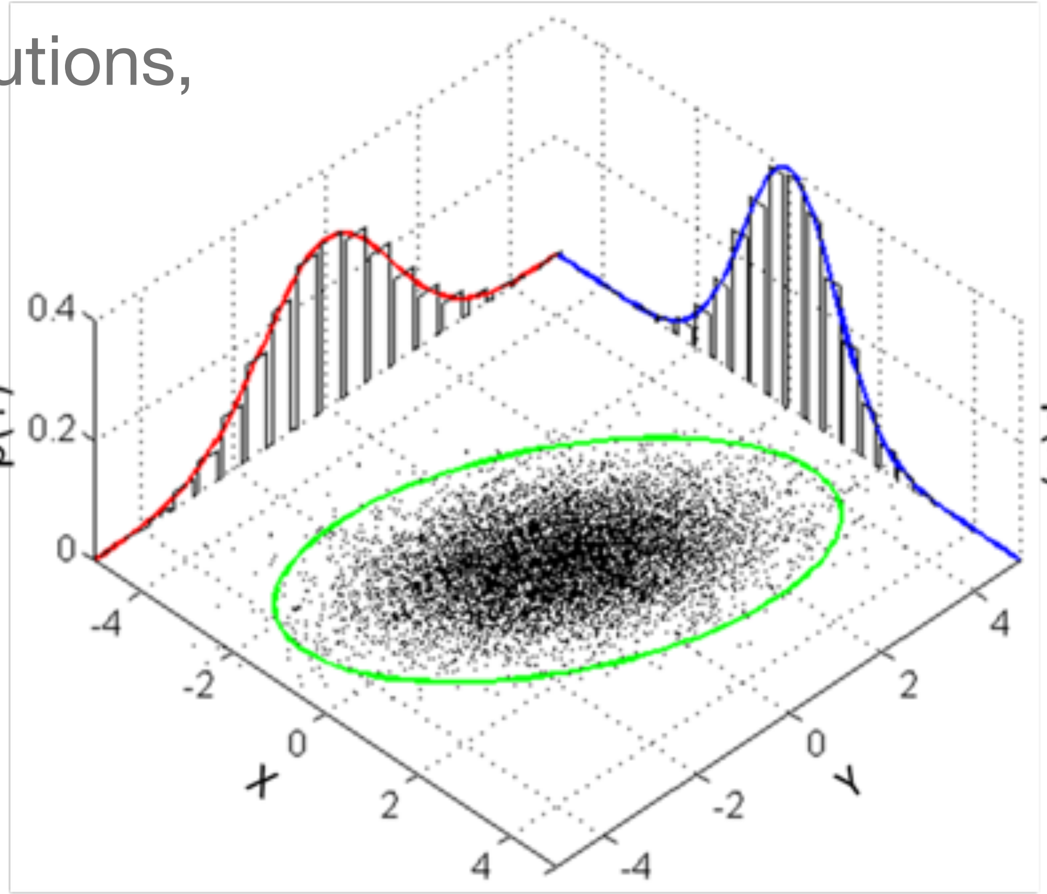
Marginalization

- Once we have the posterior, when dealing with **multivariate distributions** we will want distributions having **less** parameters than the full (original) one.

- This is done merely by **integrating over the parameters** in which are considered NOT interesting

$$p_X(x) = \int_y p_{X,Y}(x,y)\,\mathrm{d}y = \int_y p_{X|Y}(x|y)\,p_Y(y)\,\mathrm{d}y$$

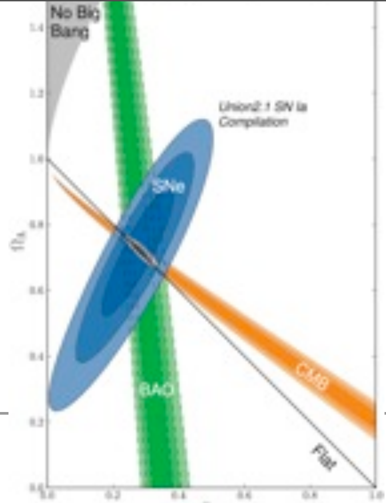
- This is usually done to obtain 1D or 2D distributions, e.g. '**Probability Distribution Function** (PDF) of a given variable x', or '**contour plot** of variables x and y'



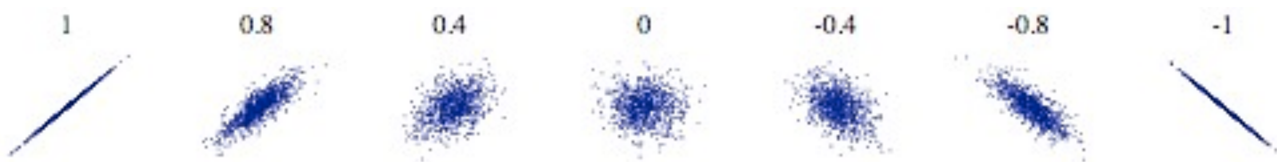
- Useful when doing graphical representations (plots) of the posterior distribution

Antonio J. Cuesta

Degeneracies



- Definition: When two (or more) variables are **correlated**, it is conventional to say that they are **degenerate** with the other parameter(s).



- This is both “good” and “bad”. The “bad” part is that the parameter are allowed to take **any value** along the degeneracy direction. The “good” news is that, if A and B are degenerate, even if there is no experiment that can constrain parameter A, you can improve limits on the allowed values of A ***indirectly***, by designing an experiment that can constrain parameter B.

- Note that this is **not an intrinsic property of the parameters** themselves, but of their posterior distributions: we will see that two parameters might be degenerate in one experiment (CMB) but not in other experiment (CMB+BAO)

The standard cosmological model

- Cosmologists also like to talk about a "**standard model**", which is the current simplest framework to describe the cosmological observations.
    - Not exactly a theory but more like a parametrization of our ignorance.
- The standard cosmological model is the **ΛCDM model**. In this model, the Universe contains both dark and baryonic matter, and the accelerated expansion rate is due to a cosmological constant.
- Also, although rarely explicitly mentioned, the theory of gravity assumed is **General Relativity** in which spacetime is defined by the **Friedman-Lemaître-Robertson-Walker metric** (FLRW)
- We know (as in particle physics) that is **not a complete description** and soon will have to incorporate other parameters (neutrino mass, primordial non-Gaussianity, primordial gravitational waves).

The standard cosmological model

- There are 6 parameters in the 'base' ΛCDM model. These are often chosen as

$$\{ \Omega_b h^2 , \Omega_c h^2 , H_0 , \tau_{reio} , A_s , n_s \}$$

  - The physical density of baryonic matter $\Omega_b h^2$
  - The physical density of cold dark matter $\Omega_c h^2$
  - The local (z=0) expansion rate H_0
  - The optical depth to reionization τ_{reio}
  - The amplitude of the primordial (scalar) power spectrum A_s
  - The spectral index of the primordial (scalar) power spectrum n_s

Beyond the standard cosmological model

- Current research also attempts to constrain parameters not included in the 'base' ΛCDM model. These are usually implemented in Boltzmann codes:

- The curvature of the Universe Ω_k
- The amount of matter in the form of massive neutrinos $\sum m_\nu$
- The effective number of relativistic species at recombination N_{eff}
- The equation of state of dark energy (and its time dependence) w_0 w_a
- The ratio between the tensor and scalar primordial power spectrum r
- All of these are already included in Planck public chains.

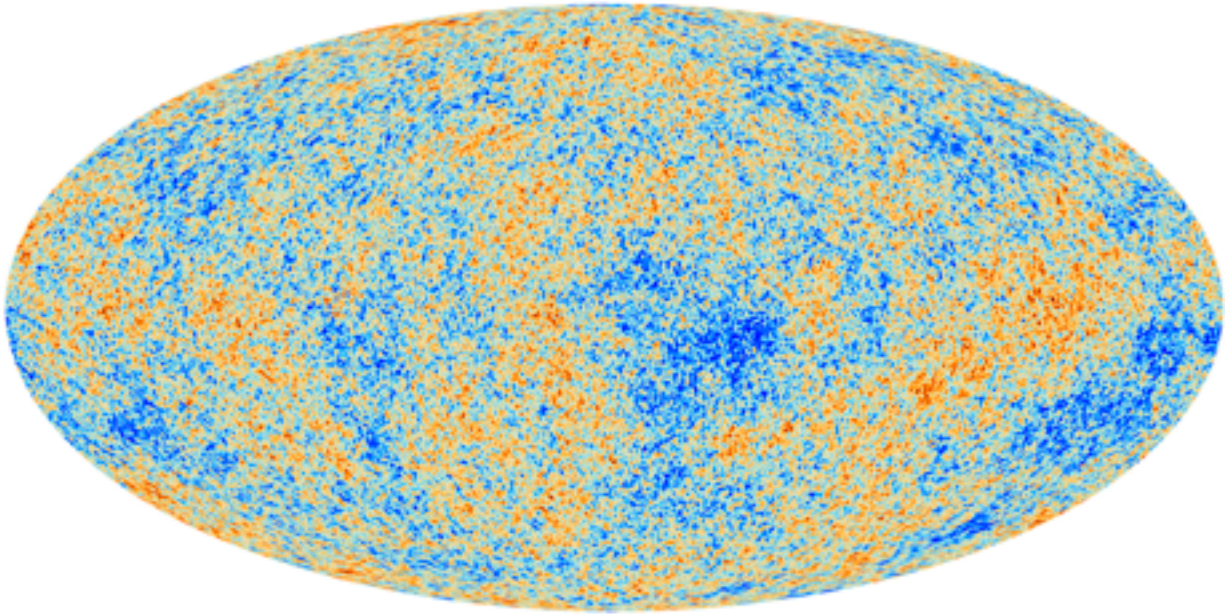
- Also considered sometimes : number of e-folds N,
- Non gaussianity f_{NL} ,
- The running of the spectral index α
- The evolution of the electromagnetic coupling constant,
- Modified gravity parameters...

Beyond the standard cosmological model

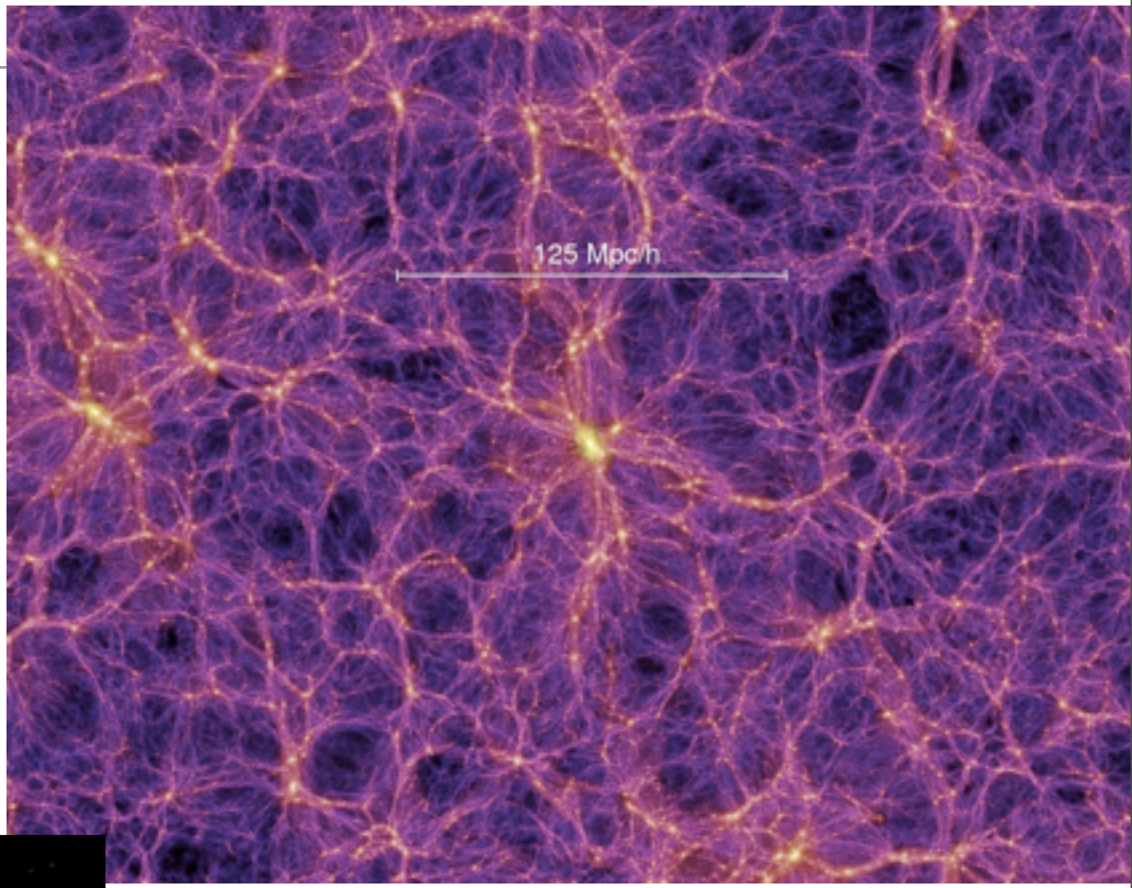
- Absolutely no agreement in the cosmology community about what extensions of the ΛCDM model will be required by future data

Conference
"Beyond ΛCDM"
University of Oslo

Observables in cosmology



CMB

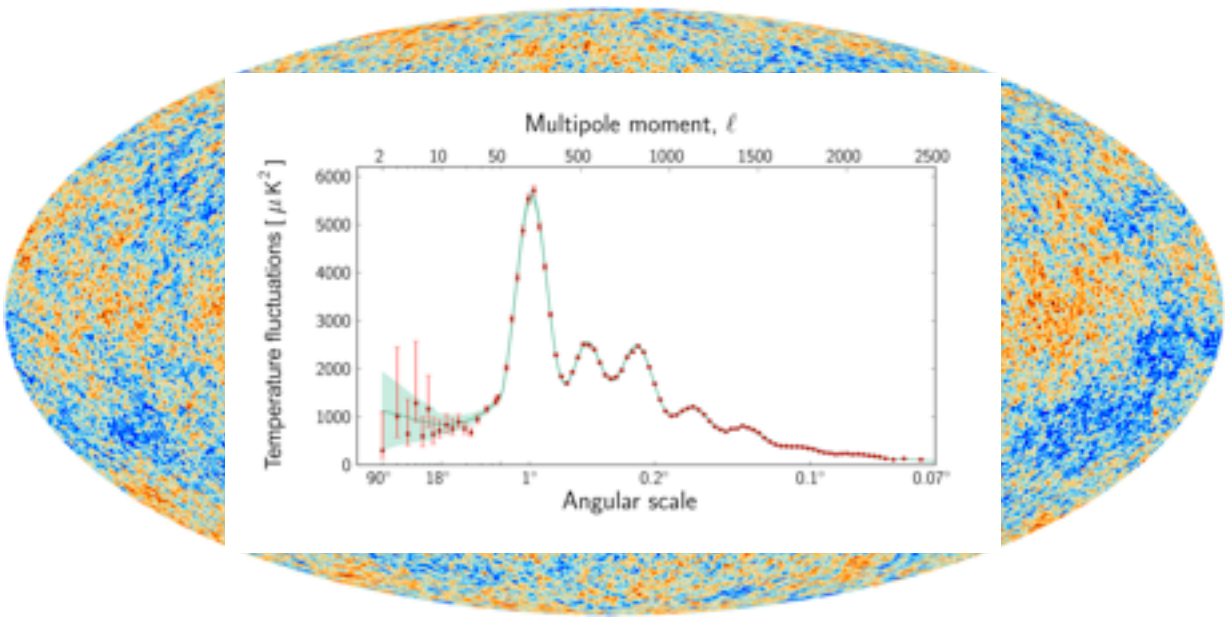


LSS

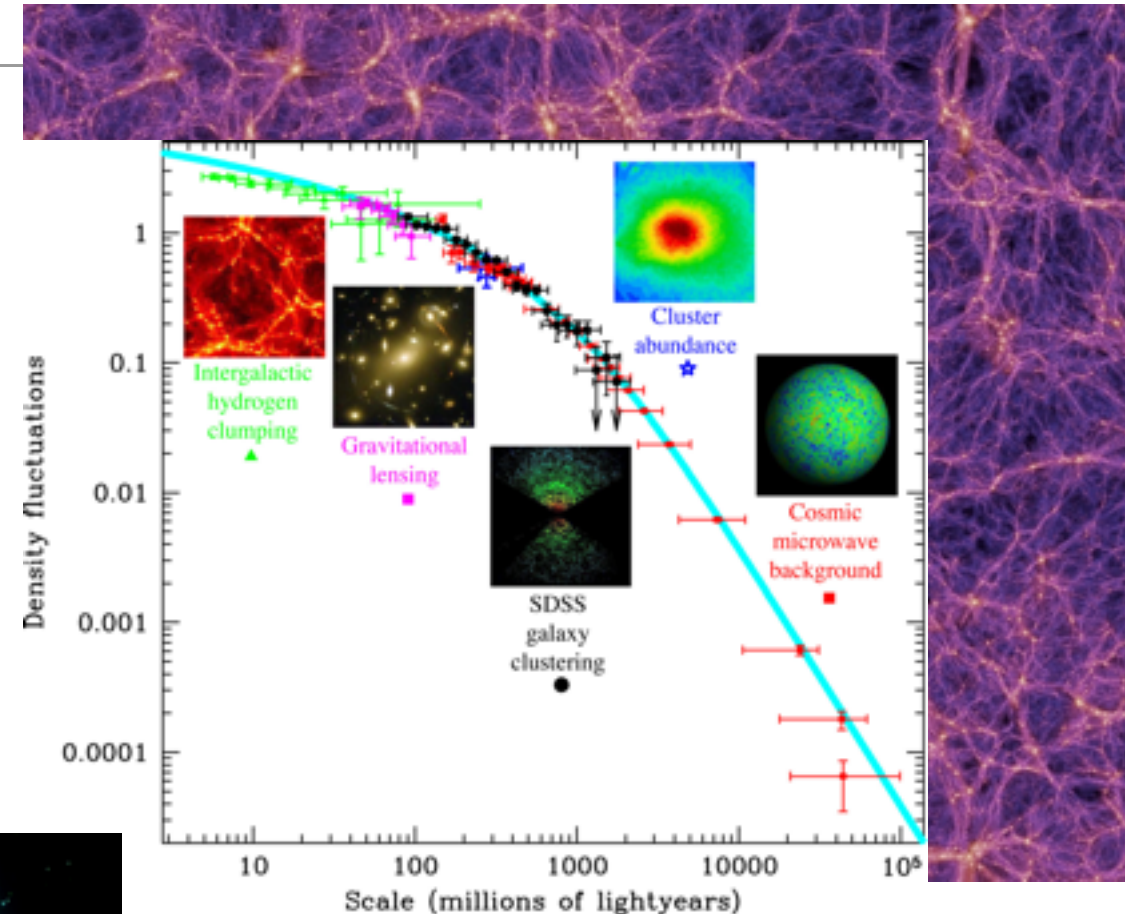
SNe

Today we talk about:
precision cosmology (%)
data-driven cosmology

Observables in cosmology

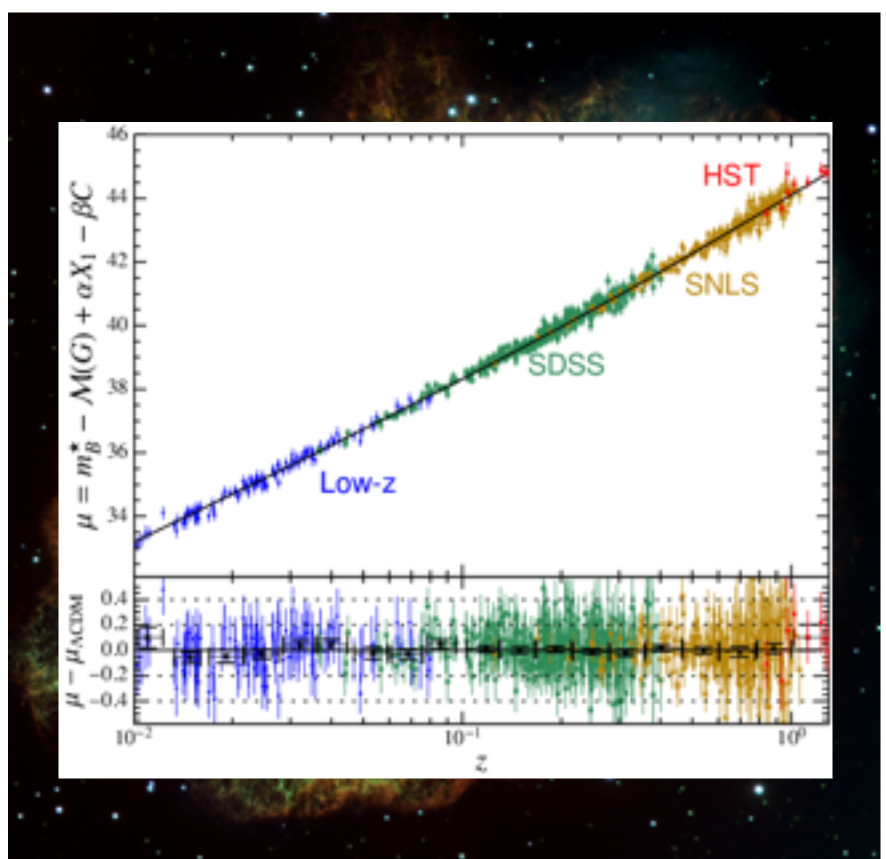


CMB



LSS

Today we talk about:
precision cosmology (%)
data-driven cosmology



SNe

Cosmological Datasets

- This is NOT a complete list, but the datasets usually considered (which we will use in this course) are:

- **CMB**. Angular power spectrum of temperature and polarization of the Cosmic Microwave Background. Latest measurements by Planck satellite.

- **BAO**. Cosmic distance scale measurements using Baryon Acoustic Oscillations. Latest measurements by Baryon Oscillation Spectroscopic Survey (BOSS) by the Sloan Digital Sky Survey (SDSS)

- **SN**. Type-Ia Supernovae. Distance modulus from the luminosity distance derived by the explosion of SnIa. Latest compilation by Joint Light Curve analysis (JLA)

Cosmological Datasets

- **P(k)**. Matter power spectrum. Measured by the clustering of the galaxy distribution or the distribution of the Lyman Alpha Forest.

- **H_0**. Hubble constant. Direct measurements of the expansion rate in the Local Universe (the Hubble constant) using cepheids, SN host galaxies, and water maser distances.

- **RSD**. Redshift space distortions. Measuring the clustering of cosmic tracers such as galaxies and Lyman-α forest in redshift space it is possible to derive structure growth and test GR. Every galaxy survey has provided a measurement.

- Others: cosmic chronometers, cluster mass function, cosmic shear, ... sometimes systematic errors are questioned and not usually included

CAMB and CosmoMC

- The parameter estimation is done in two steps, done by two **different** codes

- **CAMB**, which is a *Boltzmann* code, solves the evolution of the background and perturbation equations for each component (neutrinos, photons, cold dark matter, baryons, ...). (see Ma & Bertschinger astro-ph/9506072), This returns quantities such as the CMB power spectrum multipoles C_ℓ the matter power spectrum P(k,z) or the distance-redshift relation $\chi(z)$.

- **CosmoMC**, which is a parameter *sampling* code, implements the Metropolis-Hastings algorithm to sample a given parameter space. It is bundled up with likelihood codes from the most recent data such as BAO likelihood from BOSS or SN likelihood from JLA (can be used with CMB from Planck2015)

  Current version is July2015, which was released together with Planck2015
  Both codes are written in Fortran90

CAMB

- CAMB is actually included when you download CosmoMC. It is located in the directory /path/to/cosmomc/camb

- If you download it separately, you can run it by just doing ./camb params.ini

- This is exactly what CosmoMC does in every iteration (=every MCMC step), except that no params.ini is created, instead the parameters are passed directly to CosmoMC

```
AJCuestaMacBookAir:camb ajcuesta$ ./camb params.ini
Reion redshift         =  10.713
Om_b h^2               =  0.022600
Om_c h^2               =  0.112000
Om_nu h^2              =  0.000640
Om_Lambda              =  0.724000
Om_K                   =  0.000000
Om_m (1-Om_K-Om_L)     =  0.276000
100 theta (CosmoMC)    =  1.039532
N_eff (total)          =  3.046000
 1 nu, g= 1.0153 m_nu*c^2/k_B/T_nu0=    353.71 (m_nu=  0.060 eV)
Reion opt depth        =  0.0900
Age of universe/GYr    =  13.777
zstar                  =  1088.72
r_s(zstar)/Mpc         =  146.38
100*theta              =  1.039841
zdrag                  =  1059.70
r_s(zdrag)/Mpc         =  149.01
k_D(zstar) Mpc         =  0.1392
100*theta_D            =  0.160271
z_EQ (if v_nu=1)       =  3216.47
100*theta_EQ           =  0.847737
tau_recomb/Mpc         =  284.95  tau_now/Mpc =  14362.3
```

Base parameters

- Also, instead of outputting C_ℓ and P(k,z) to a file, they are passed to CosmoMC

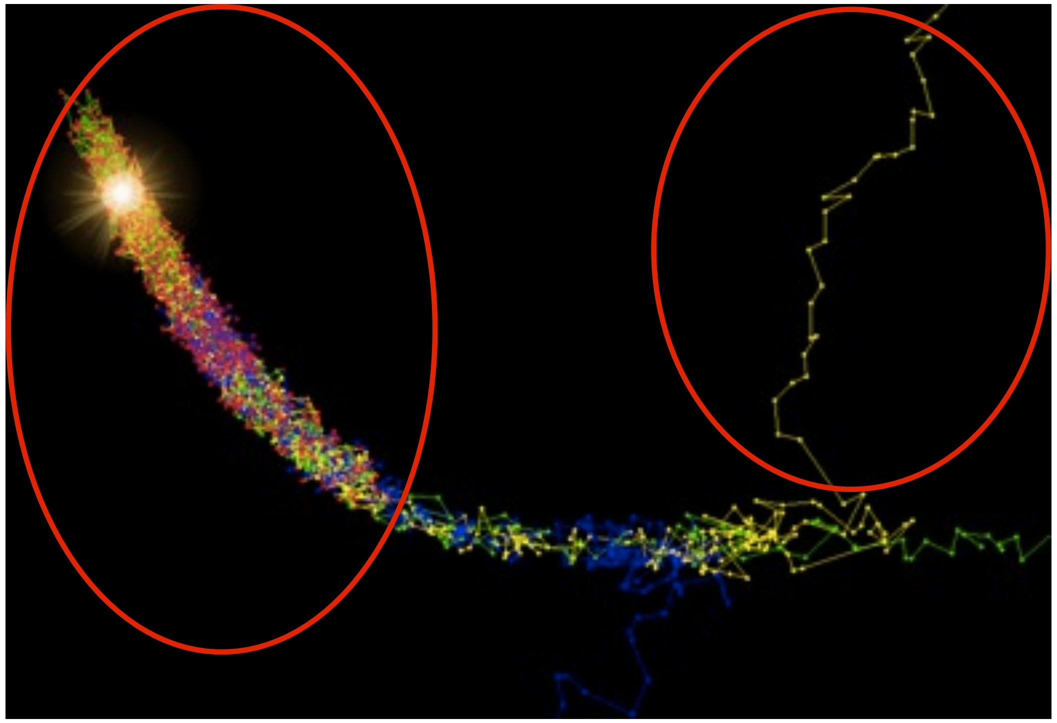
Markov Chain Monte Carlo (MCMC)

- It is a **random walk** to effectively sample a multi-dimensional parameter space (when the number of dimensions is large, sampling a grid is expensive)

- We will use the **Metropolis-Hastings** algorithm:
  if a step puts you in a place with more probability -> ACCEPT
  if not -> ACCEPT with probability=likelihood ratio=P_{new}/P_{old})
  otherwise, REJECT (try a different move from previous step)

Burn-in phase (retains memory of the initial condition)

After enough iterations all initial conditions will end up at the region in the parameter space with maximum likelihood



Markov Chain Monte Carlo (MCMC)

- In CosmoMC the **burn-in** phase is set by default to the first **30%** of the chain, but it can be changed by ignore_rows in distparams.ini to any value

- The code also needs to know how big the steps should be. In a MCMC you want to fully sample the parameter space until you reach regions of low likelihood. Typically the **acceptance rate** of steps should be around **30%**.

- If you **underestimate** the errors in your parameters, acceptance rate will be 100%, which means that you **never reach regions** of lower likelihood. On the other hand, if you **overestimate** the errors, acceptance rate will be 0%, which means that the code will take a **long time** to get a useful number of points.

- In order to do this, you should set in **params.ini** a good estimate of your expected **covariance matrix** or **parameter errors**. It is not important to be precise, since the covariance matrix of the parameters will be updated from the chains themselves.

Likelihood and χ^2 (Wilks' Theorem)

- The measurements in cosmological datasets are translated to **likelihoods**. The total likelihood, assuming the measurements of the experiments are not correlated (usually the case), is the product of individual likelihoods.

- However in practice, CosmoMC uses the log of the likelihood (with opposite sign), which is closely related to the value of the χ^2 distribution. These are related by the equation:

$$\mathscr{L} \propto \exp(-\chi^2/2)$$

- Where the χ^2 for simple cases is just (measurement-theory)^2/error^2, or more generally $\chi^2 = (\mathbf{D}-\mathbf{T})^t\,\mathbf{C}^{-1}\,(\mathbf{D}-\mathbf{T})$ where **D** is a vector of data measurements and **T** is the theory vector generated at each MCMC step. $\mathbf{C}$ is the **covariance matrix** whose elements are the covariances between parameters $\mathbf{C}=\{\sigma_{\theta i,\theta j}\}$, so that the diagonal elements are $\sigma^2_{\theta i}$ and the off-diagonal are $\sigma_{\theta i \theta j} = \rho_{ij}\,\sigma_{\theta i}\,\sigma_{\theta j}$

Bayesian parameter estimation

- Once that the ***posterior*** is derived, one can compute the bounds on each model parameter from the ***marginalized*** probability distribution for such parameter

- If the distribution is **two-sided**, the bounds are usually quoted in terms of (mean ± sigma) or (mean ± 2sigma) which for a Gaussian distribution correspond to the 68% and 95% confidence level bounds, i.e. the "true" value of the parameter has a 68% or 95% probability of being inside said interval

- For parameters whose distribution is **one-sided**, i.e. when only the upper limit (or only the lower limit) is interesting, the bound quoted is such that 32% (=100%-68%) or more usually 5%(=100%-95%) of the distribution is to the right (to the left) of the quoted bound

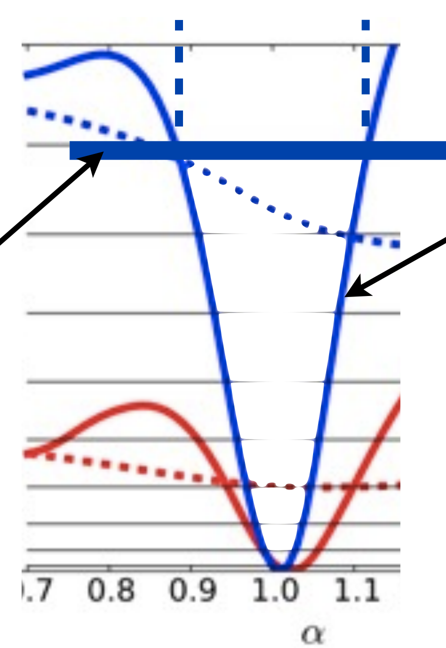
Bayesian parameter estimation

- In case of **non-symmetric** distributions one quotes the mean value together with an **upper** bound and a **lower** bound, e.g., $x = mean^{+(upper-mean)}_{-(mean-lower)}$

- The way to compute this upper and lower limits is **not unique.** The standard has become to be the “**water level**” method, in which one “fills” the upside-down distribution until one reaches the desired confidence level (68%,95%...)

The “water level” is just
a y-axis value which crosses
the distribution in 2 points

The code just shifts
this level starting
at the maximum likelihood



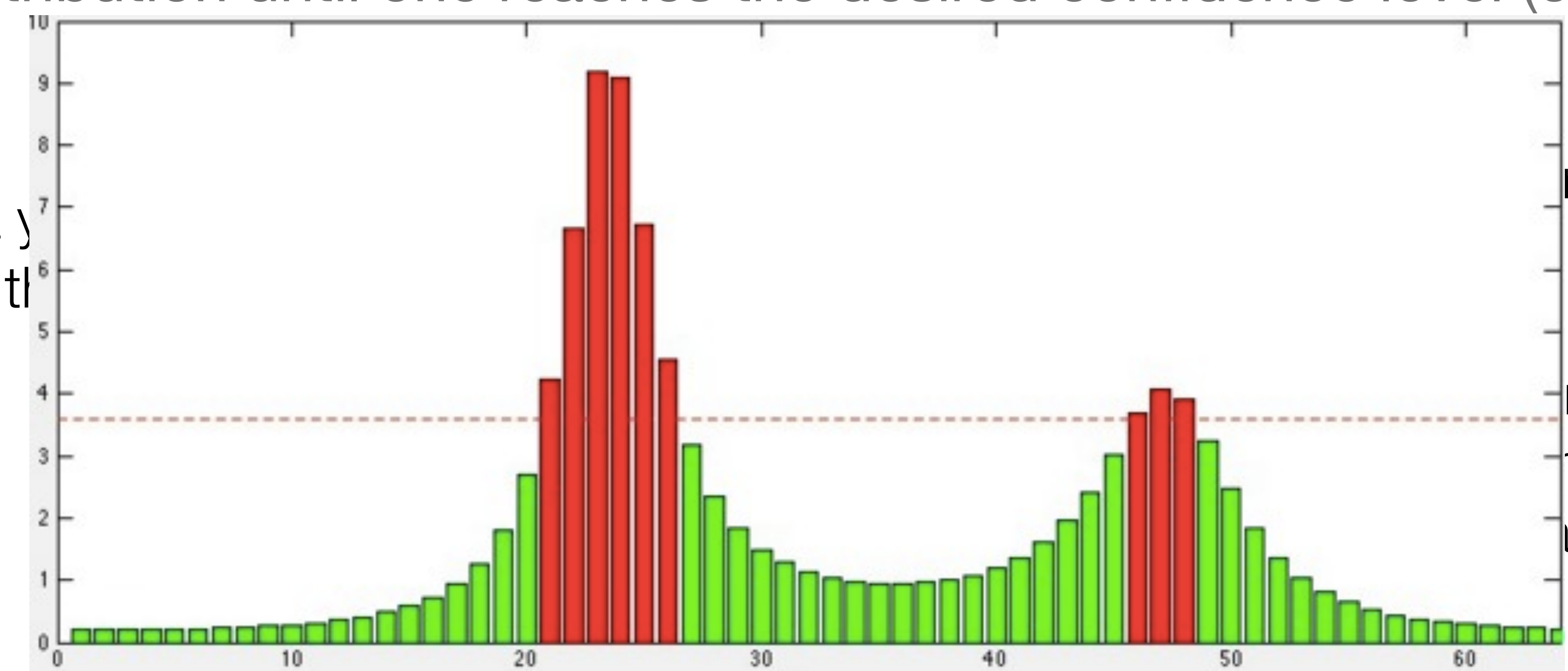
95% of the distribution
is contained here

The bounds are not
necessarily symmetric
around the mean value

Bayesian parameter estimation

- In case of **non-symmetric** distributions one quotes the mean value together with an **upper** bound and a **lower** bound, e.g., $x = \text{mean}^{+(\text{upper-mean})}_{-(\text{mean-lower})}$

- The way to compute this upper and lower limits is **not unique**. The standard has become to be the "**water level**" method, in which one "fills" the upside-down distribution until one reaches the desired confidence level (68%,95%...)



a y ribution

tl here

re not

nmetric

n value

Other MCMC codes

- Although **CosmoMC** has been the de-facto standard for the past decade (WMAP and Planck), this might change in the future depending on usage
- A strong competitor is **MontePython**, a python code that interfaces the Boltzmann solver CLASS. The advantage of CLASS over CAMB is that it matches the notation in Ma & Bertschinger, so it is easier to generalize https://github.com/baudren/montepython_public
- A rising code is **cosmosis**, another python code that is extremely modular, which makes it easy to switch between MCMC samplers, Boltzmann codes,... https://bitbucket.org/joezuntz/cosmosis/wiki/Home
- Other samplers have been used in the past (CosmoPMC, PICO, etc.) but they do not seem to be in active development or use approximated methods