

Statistical Hypothesis Tests

Markus Ahlers & D. Jason Koskinen

March 2022

UNIVERSITY OF
COPENHAGEN



Statistical Hypothesis Tests

- Typical problem in physics and astronomy:

You have collected data with your experiment or observatory and want to test a theory (signal hypothesis H_1)?

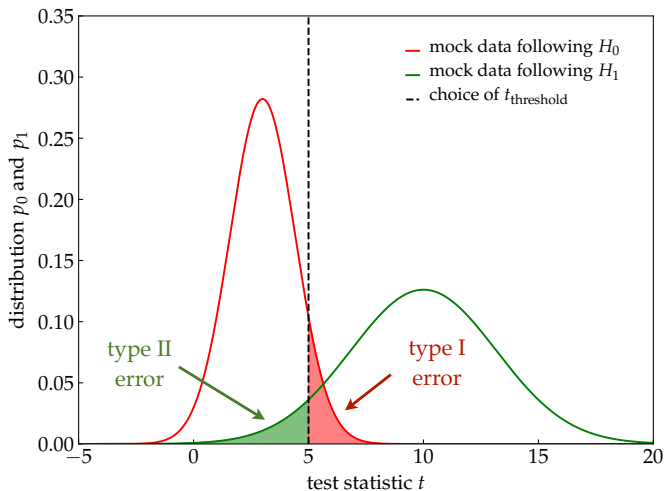
- How can you judge if the hypothesis is correct/wrong?
- How does the alternative hypothesis (null hypothesis H_0) look like?
- How confident can you be that your conclusions are correct?
- In most cases there is a chance that your decision is wrong:
 - ✗ You **decided** that H_1 is **correct**, but it is actually **wrong**? (**type I error**)
 - ✗ You **decided** that H_1 is **wrong**, but it is actually **correct**? (**type II error**)

Statistical Hypothesis Tests

- A **statistical hypothesis test** is based on a quantity called **test statistic** that allows to quantify the degree of confidence that your decision was right or wrong.

- A useful test statistic:
 - is **sensitive** to the signal hypothesis H_1 (that's a must!)
 - is **efficiently calculable** (e.g. fast calculation on your computer)
 - has a **well-known behaviour** for data following the null hypothesis H_0 (more on this later)

Test Statistic Distribution



In a hypothesis test we have to choose a **critical** t -value to either reject or accept the hypothesis.

Test Statistic Distribution

- **significance** (α) :

Probability that background would have created outcome with same t or larger (**type I error**):

$$\alpha = \int_{t_{\text{obs}}}^{\infty} dt p_0(t) = \text{“}p\text{-value”}$$

- **Note:** It is a **convention** that t *increases* for a more “signal-like” outcome. If not, just define a new test statistic $t' = -t$.
- **power of test** ($1 - \beta$) :

Probability that signal would have created outcome with same t or less (**type II error**):

$$\beta = \int_{-\infty}^{t_{\text{obs}}} dt p_1(t)$$

Statistical Hypothesis Tests

→ A good statistical test will have good “separation” of p_0 and p_1 to allow a minimize type I/II errors. Separation from background allows to quantify **significance** of event excesses:

- **discovery** (in particle physics) :

$$\alpha \simeq 5.7 \times 10^{-7} (“5\sigma”)$$

- **evidence** (in particle physics) :

$$\alpha \simeq 2.7 \times 10^{-4} (“3\sigma”)$$

- Often, we want to estimate the **performance** of a statistical test prior to a measurement by simulations. We can determine this by tuning the signal strength, e.g. the IceCube experiment uses:

- **discovery potential:**

$$\alpha \simeq 5.7 \times 10^{-7} (“5\sigma”) \quad \text{and} \quad \beta = 0.5$$

- **90% sensitivity level:**

$$\alpha = 0.5 \quad \text{and} \quad \beta = 0.1$$

Today's Program

- **Today**, we will explore various examples of hypothesis tests and test statistics:
- **Maximum likelihood ratio test**
 - This is the most powerful test statistic (**Neyman-Pearson theorem**).
 - Allows to quantify background distributions p_0 (**Wilks theorem**).
 - We will study the applicability of **Wilks theorem** by a **numerical example (exercise 1)**.
 - Discussion of **trials factor** corrections.
- **Kolmogorov-Smirnov test & Auto-Correlation**
 - We will introduce this test by the **cumulative auto-correlation function** of event distributions on a sphere.
 - This test allows to study hidden structure in event distributions, e.g. deviations from an isotropic distribution.
 - We will **generate Monte Carlo data** following isotropic and simple anisotropic distributions and study the performance of the test (**exercise 2**).

Extra Material

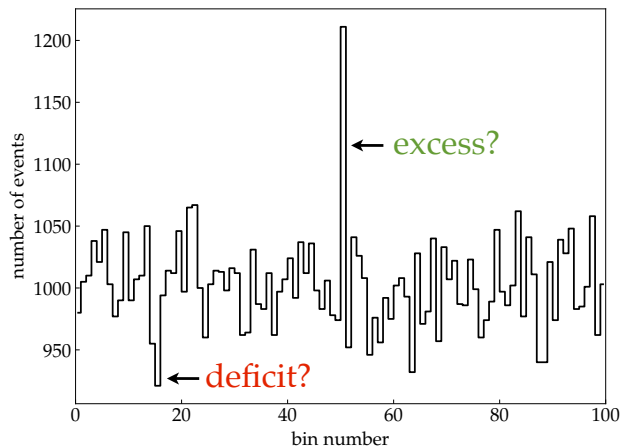
- **Angular power spectrum** (optional, will not be covered during lecture)
 - The power spectrum C_ℓ can be used as a test statistic that allows to study distributions of data (large number of events, temperature fluctuations (CMB),...) on a sphere.
 - Brief introduction of spherical harmonics $Y_{\ell m}$ as basis functions on a sphere (exercise 3).
 - Introduction of the **two-point angular correlation function** and its relation to the power spectrum.
 - Introduction of the power spectrum.
 - Extraction of power spectra from Monte Carlo data and background (exercise 4).

Part I

Maximum Likelihood Ratio

Recap: Maximum Likelihood Ratio

- Consider data (N_{tot} “events”) distributed in N_{bins} bins.
- **Question:** Is there an **excess** or **deficit** in the data?



Recap: Maximum Likelihood Ratio

- Likelihood for data vector \mathbf{x} and parameter vector $\boldsymbol{\mu}$:

$$\mathcal{L}(\boldsymbol{\mu}|\mathbf{x}) = \underbrace{\prod_{i=1}^{N_{\text{bins}}} \frac{\mu_i^{x_i}}{x_i!} e^{-\mu_i}}_{\text{Poisson distributions}}$$

- Null hypothesis (“no signal”)

$$\mu_i = \mu_{\text{bg}} = \text{const}$$

- Signal hypothesis (“signal (excess or deficit) in bin 1”)

$$\mu_i = \begin{cases} \mu_{\text{sig}} + \mu_{\text{bg}}^* & i = 1 \\ \mu_{\text{bg}}^* & 2 \leq i \leq N_{\text{bins}} \end{cases}$$

! **Important note:** $\mu_{\text{bg}}^* \neq \mu_{\text{bg}}$

Maximum of Null Hypothesis

- **for convenience** : likelihood \rightarrow natural 'log-likelihood' (LLH)

$$\ln \mathcal{L}(\boldsymbol{\mu}|\mathbf{x}) = \sum_{i=1}^{N_{\text{bins}}} (x_i \ln \mu_i - \mu_i) + \underbrace{\text{const}}_{\text{independent of } \boldsymbol{\mu}}$$

- In general, maximum of LH (or LLH) can be derived numerically.
This example is easy enough to solve analytically:
- maximum LH value determined by:

$$\frac{d \ln \mathcal{L}}{d\mu_{\text{bg}}} = 0 = \sum_{i=1}^{N_{\text{bins}}} \left(\frac{x_i}{\mu_{\text{bg}}} - 1 \right)$$

- maximum $\hat{\mu}_{\text{bg}}$ obeys:

$$\hat{\mu}_{\text{bg}} = \frac{N_{\text{tot}}}{N_{\text{bins}}}$$

Maximum of Signal Hypothesis

- For the signal hypothesis we have to find the maximum w.r.t. signal and background strength:

$$\frac{d \ln \mathcal{L}}{d\mu_{\text{bg}}^*} = 0 \quad \text{and} \quad \frac{d \ln \mathcal{L}}{d\mu_{\text{sig}}} = 0$$

- Signal term μ_{sig} is (by construction) only present in bin 1.
- maximum $\{\hat{\mu}_{\text{bg}}^*, \hat{\mu}_{\text{sig}}\}$ obeys:

$$\hat{\mu}_{\text{bg}}^* = \frac{N_{\text{tot}} - x_1}{N_{\text{bins}} - 1}$$

$$\hat{\mu}_{\text{sig}} = x_1 - \hat{\mu}_{\text{bg}}^* = \frac{x_1 N_{\text{bins}} - N_{\text{tot}}}{N_{\text{bins}} - 1}$$

Maximum LH Ratio

- test statistic λ is defined as maximum likelihood ratio:

$$\lambda(\mathbf{x}) = -2 \ln \frac{\mathcal{L}(\mathbf{x} | \hat{\mu}_{\text{bg}}, 0)}{\mathcal{L}(\mathbf{x} | \hat{\mu}_{\text{bg}}^*, \hat{\mu}_{\text{sig}})}$$

- after some algebra using the solutions of $\hat{\mu}_{\text{bg}}$, $\hat{\mu}_{\text{bg}}^*$, and $\hat{\mu}_{\text{sig}}$:

$$\lambda(\mathbf{x}) = 2x_1 \ln \left(\frac{N_{\text{bins}}}{N_{\text{tot}}} x_1 \right) + 2(N_{\text{tot}} - x_1) \ln \left(\frac{N_{\text{bins}}}{N_{\text{tot}}} \frac{N_{\text{tot}} - x_1}{N_{\text{bins}} - 1} \right) \quad (1)$$

- **Note:** The first (or second) term in Eq.(1) vanishes in the special case $x_1 = 0$ (or $N_{\text{tot}} - x_1 = 0$).
 - **bonus exercise:** Derive $\hat{\mu}_{\text{bg}}$, $\hat{\mu}_{\text{bg}}^*$, $\hat{\mu}_{\text{sig}}$, and Eq.(1).
- **exercise 1** : Let's explore the behaviour of Eq.(1).

Exercise 1

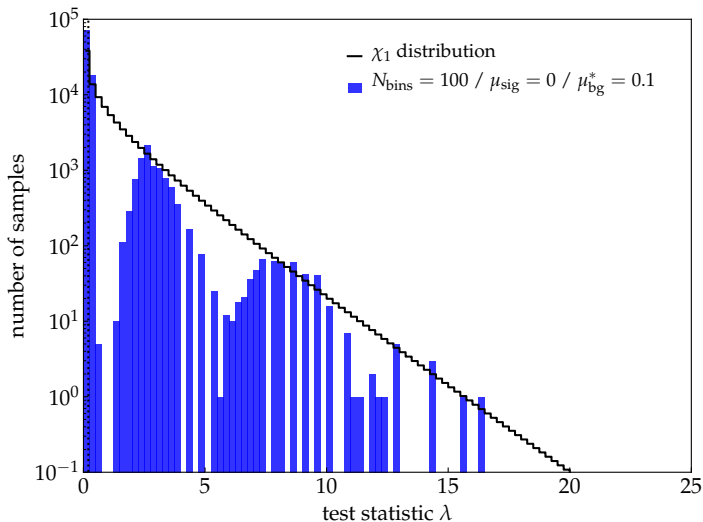
- Generate Monte Carlo data assuming $N_{\text{bins}} = 100$ bins.
- Consider two categories:
 - **three background cases:**
choose $\mu_{\text{sig}} = 0$ and $\mu_{\text{bg}} = 0.1, 10, \text{ or } 1000$.
 - **two signal cases:**
choose $\mu_{\text{bg}}^* = 1000$ and signal in first bin ($i = 1$) with $\mu_{\text{sig}} = 100$ and 200 .
- For each case generate many (10^5) pseudo-experiments, i.e. trials, $\mathbf{x} = \{x_1, \dots, x_{N_{\text{bins}}}\}$ of Monte Carlo data and calculate $\lambda(x_1, N_{\text{tot}} = \sum_{i=1}^{N_{\text{bins}}} x_i)$:

$$\lambda = 2x_1 \ln \left(\frac{N_{\text{bins}}}{N_{\text{tot}}} x_1 \right) + 2(N_{\text{tot}} - x_1) \ln \left(\frac{N_{\text{bins}}}{N_{\text{tot}}} \frac{N_{\text{tot}} - x_1}{N_{\text{bins}} - 1} \right)$$

- Make histograms of the λ values to estimate the null and signal distributions.

Exercise 1: Background Cases

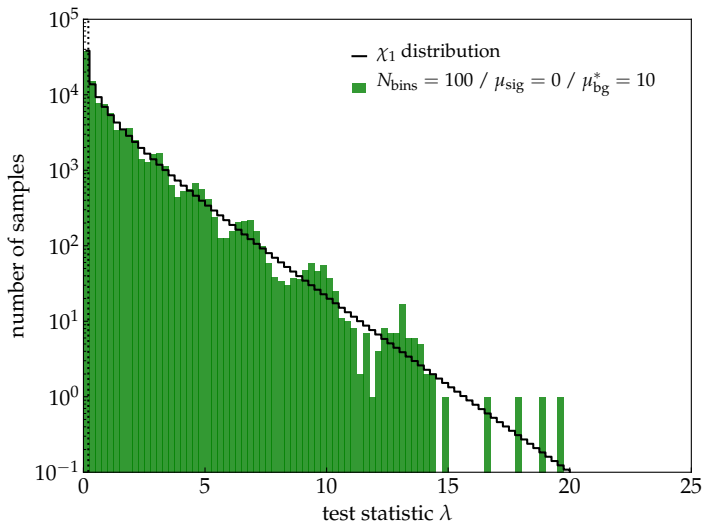
simulation (10^5 samples)



for python code see : `maxLH_produce.py` & `maxLH_show.py`

Exercise 1: Background Cases

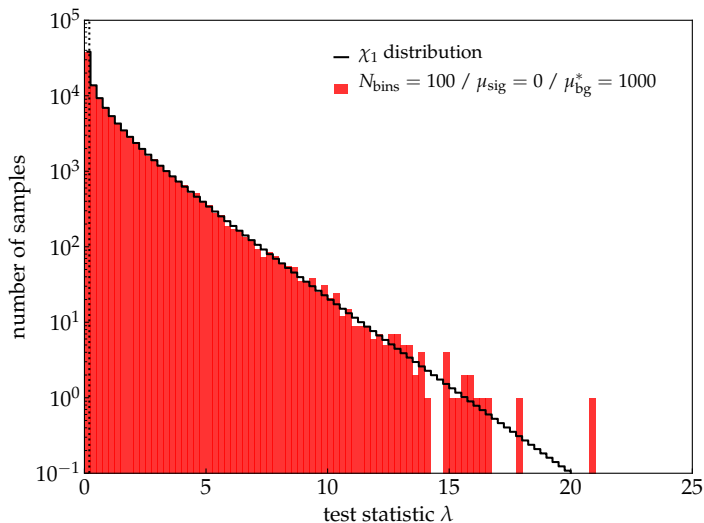
simulation (10^5 samples)



for python code see : `maxLH_produce.py` & `maxLH_show.py`

Exercise 1: Background Cases

simulation (10^5 samples)



for python code see : `maxLH_produce.py` & `maxLH_show.py`

Wilks Theorem (1938)

THE LARGE-SAMPLE DISTRIBUTION OF THE LIKELIHOOD RATIO FOR TESTING COMPOSITE HYPOTHESES¹

BY S. S. WILKS

(...)

Theorem: If a population with a variate x is distributed according to the probability function $f(x, \theta_1, \theta_2 \dots \theta_h)$, such that optimum estimates $\bar{\theta}_i$ of the θ_i exist which are distributed in large samples according to (3), then when the hypothesis H is true that $\theta_i = \theta_{0i}$, $i = m + 1, m + 2, \dots, h$, the distribution of $-2 \log \lambda$, where λ is given by (2) is, except for terms of order $1/\sqrt{n}$, distributed like χ^2 with $h - m$ degrees of freedom.

bonus exercise: Try to find this publication online.

Wilks Theorem

- **Brief reminder:**

- Let \mathbf{x} be data that follows a probability function $f(\mathbf{x}|\theta_1, \dots, \theta_n)$.
- The corresponding likelihood function for the null hypothesis $\mathcal{L}_0(\theta_1, \dots, \theta_n|\mathbf{x})$ has a maximum at $\hat{\theta}_1, \dots, \hat{\theta}_n$.
- The corresponding likelihood function for the alternative hypothesis $\mathcal{L}_A(\theta_1, \dots, \theta_n|\mathbf{x})$ has a maximum at $\hat{\theta}_1, \dots, \hat{\theta}_{m-1}, \hat{\theta}_m$, for $m > n$.

- **Wilks theorem:**

For a large number of samples \mathbf{x} , the distribution of the test statistic

$$-2 \ln \frac{\mathcal{L}_0(\hat{\theta}_1, \dots, \hat{\theta}_n|\mathbf{x})}{\mathcal{L}_A(\hat{\theta}_1, \dots, \hat{\theta}_{m-1}, \hat{\theta}_m|\mathbf{x})}$$

approaches a χ_k^2 distribution with $k = m - n$ in the limit of a large number of events, N_{tot} .

Quick Example

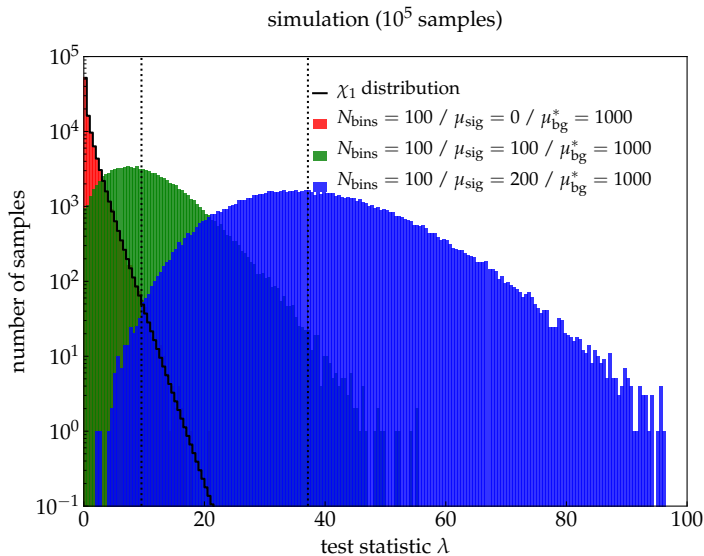
- For large N_{tot} we can apply Wilks theorem and assume that the background distribution follows a χ_1^2 distribution.

$$p\text{-value} = \int_{\lambda_{\text{obs}}}^{\infty} dx \chi_k^2(x) = 1 - \text{erf}(\sqrt{\lambda_{\text{obs}}/2})$$

- Assume $N_{\text{tot}} = 10^5$, $N_{\text{bins}} = 100$ and first bin contains:
 - 1100 events : maximum likelihood value $\lambda_{\text{obs}} \simeq 9.8$
Wilks theorem: $p \simeq 0.0017$
 - 1150 events : maximum likelihood value $\lambda_{\text{obs}} \simeq 21.7$
Wilks theorem: $p \simeq 3.2 \times 10^{-6}$
 - 1200 events : maximum likelihood value $\lambda_{\text{obs}} \simeq 38.0$
Wilks theorem: $p \simeq 7.1 \times 10^{-10}$

→ the 5σ discovery threshold corresponds to $x_1 \simeq 1162$ events

Exercise 1, cont.: Signal vs. Background



for python code see : `maxLH_produce.py` & `maxLH_show.py`

Sensitivity and Discovery Potential

- performance of the test
 - **sensitivity level:**
defined as the level of μ_{sig} such that 90% of the signal distribution is above 50% of the background distribution
 - **discovery potential:**
defined as the level of μ_{sig} such that 50% of samples have a chance probability of 5.7×10^{-7} to be generated by background only
- This is a **challenge for brute-force background simulation** – you need $N_{\text{samples}} \gg 10^7$ for accuracy!
- However, **Wilks theorem** allows to extrapolate the background distribution very easily:
- For χ_1 distribution we know that the “ 5σ ” level corresponds to:

$$\lambda_{\text{threshold}} = 5^2 = 25$$

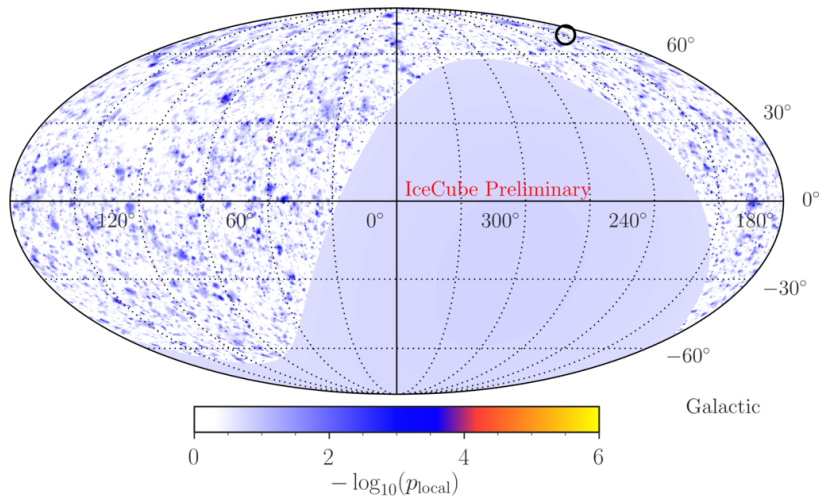
Trial Correction

- What happens if we want to find a signal not just in bin 1 but in *any* of the N_{bins} bins?
- We can simply repeat the test over all bins and identify the bin with minimum p -value p_* .
- **Problem:** There are many bins (“hypothesis”) and we have to account for the fact that there can be a chance fluctuation in the local p -values.
- If N_{bins} are independent of each other (as in our example) then we can define a post-trial p -value as

$$p_{\text{post}} = 1 - \underbrace{(1 - p_*)^{N_{\text{bins}}}}_{\text{background probability}} \simeq N_{\text{bins}} p_*$$

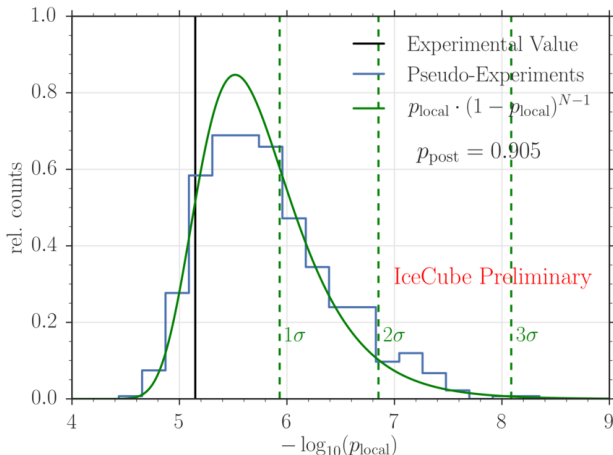
- Number of independent “trials”, N_{trials} , is often difficult to estimate.

Example: IceCube Neutrino Data



"All-sky" point-like source search:
each location tested for an excess!

Example: IceCube Neutrino Data

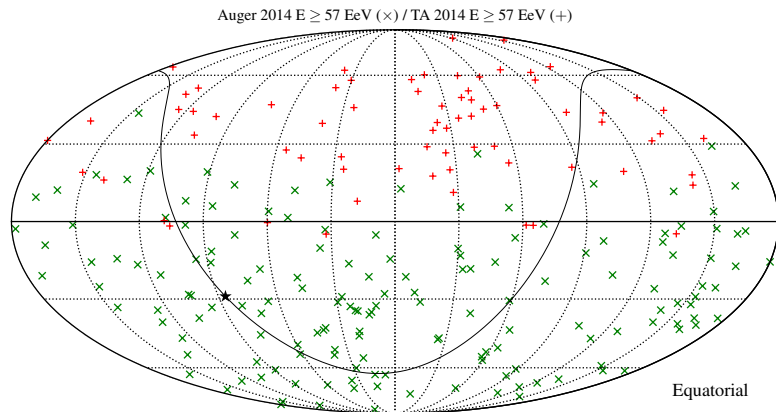


- Trial factor: $N_{\text{trials}} \sim N_{\text{bins}} \sim \mathcal{O}(1000)$
- **IceCube procedure:** choose maximal p_{local} in sky map as a new **test statistic** and compare against maximal p_{local} of randomly generated sky maps

Part II

Kolmogorov Smirnov Test & Auto-Correlation

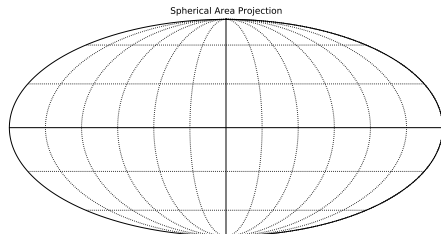
Example: Arrival Direction of Cosmic Rays



Anisotropies in the arrival directions of ultra-high energy cosmic rays (data from the observatories Telescope Array (TA) and Auger).

Mini-Exercise: Isotropy on a Sphere

- Our data and Monte Carlo has been analyzed in *mostly* linear cartesian coordinate reference frames.
- When switching to spherical coordinates we moving to azimuth and zenith angles which are **not** uniform in (x,y) .
 - Isotropy is 'uniform' randomly dispersed data on the surface area: 4π steradians for a sphere.
 - Our normal linear sampling in x and y (or even linear in azimuth and zenith) will overpopulate the poles.
- Find a way to generate Monte Carlo uniformly on a surface in spherical coordinates.



Auto-Correlation

- So far, we have only looked into local excesses in individual bins.
- This method was not sensitive to the correlation between events, e.g. in neighbouring bins or in small clusters.
- Consider N_{tot} events distributed on a sphere with position \mathbf{n}_i (unit vector).
- For two events with label i and j ($i \neq j$) we can define an angular distance:

$$\cos \varphi_{ij} = \mathbf{n}_i \cdot \mathbf{n}_j$$

- The **cumulative two-point auto-correlation function** is defined as

$$\mathcal{C}(\{\mathbf{n}_i\}, \varphi) = \frac{2}{N_{\text{tot}}(N_{\text{tot}} - 1)} \sum_{i=1}^{N_{\text{tot}}} \sum_{j=1}^{i-1} \Theta(\cos \varphi_{ij} - \cos \varphi) \quad (2)$$

with **step function** $\Theta(x) = 1$ for $x \geq 0$ and $\Theta(x) = 0$ for $x < 0$.

→ This expression counts the pairs of events within angular distance φ .

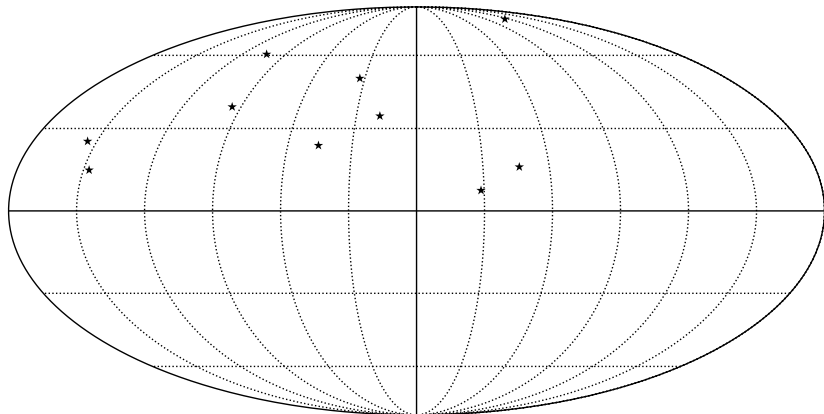
- **Note** : The step function $\Theta()$ is sometimes referenced as the Heaviside function.

Exercise 2: Event Distributions

- Generate Monte Carlo data of events on a sphere for two categories:
- **isotropic distribution:**
 - generate N_{tot} events randomly distributed on a sphere
 - e.g. python module `healpy` allows for pixelised sky maps with equal pixel area
 - Derive the two-point auto-correlation function for the distribution.
 - What distribution do you expect for a large number of events?
- **biased distribution** (bonus exercise):
 - generate N_{tot} events following a non-isotropic distribution
 - e.g. only sample events within a limited azimuth or zenith range, or events following a dipole distribution
 - How does the auto-correlation function compare to that of the isotropic distribution?

Exercise 2: Isotropic Distribution

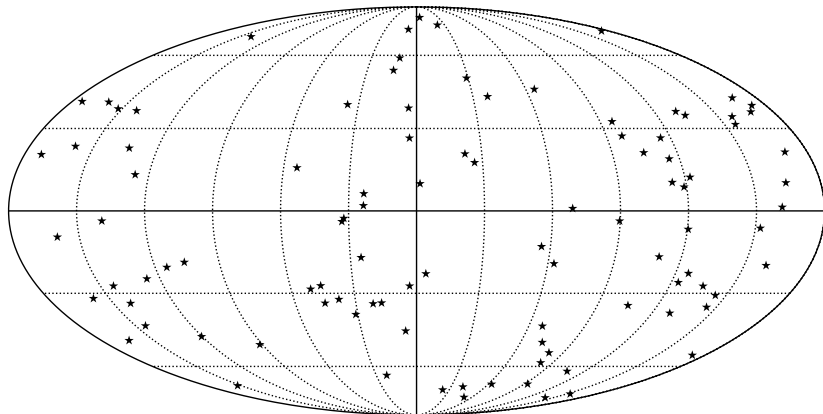
simulation ($N_{\text{tot}} = 10$)



for python code see : `twopoint.py`

Exercise 2: Isotropic Distribution

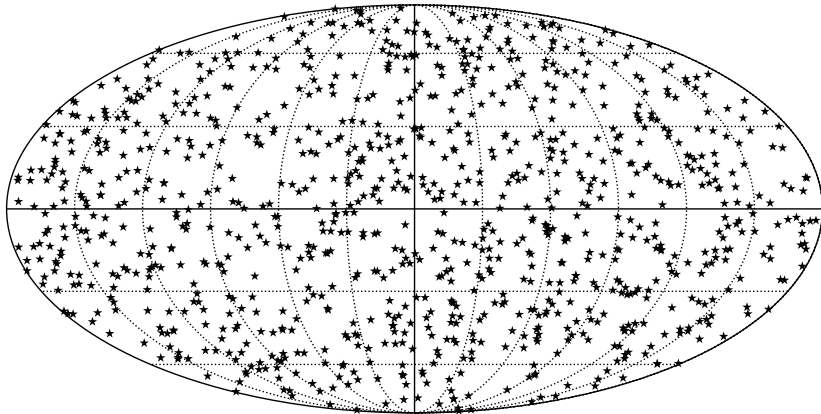
simulation ($N_{\text{tot}} = 100$)



for python code see : `twopoint.py`

Exercise 2: Isotropic Distribution

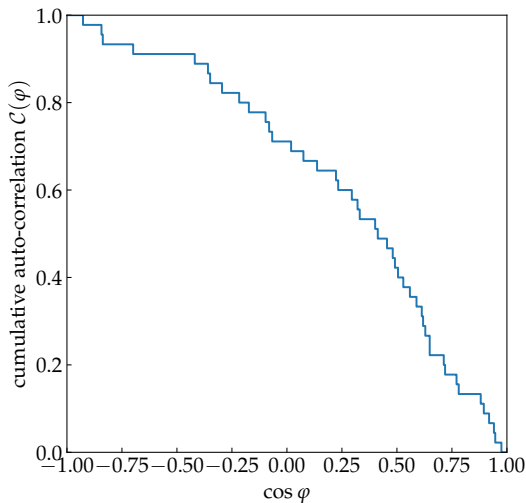
simulation ($N_{\text{tot}} = 1000$)



for python code see : `twopoint.py`

Exercise 2: Isotropic Distribution

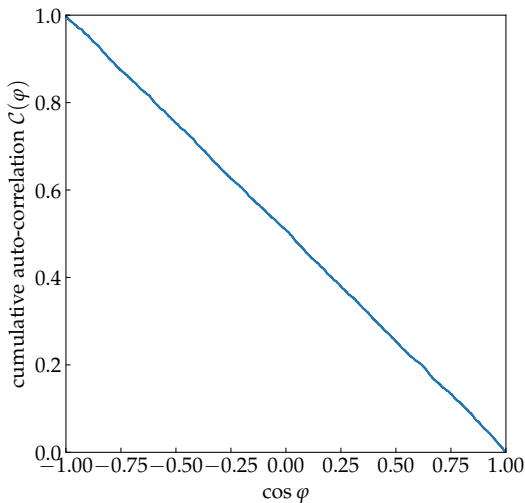
simulation (10 events)



for python code see : `twopoint.py`

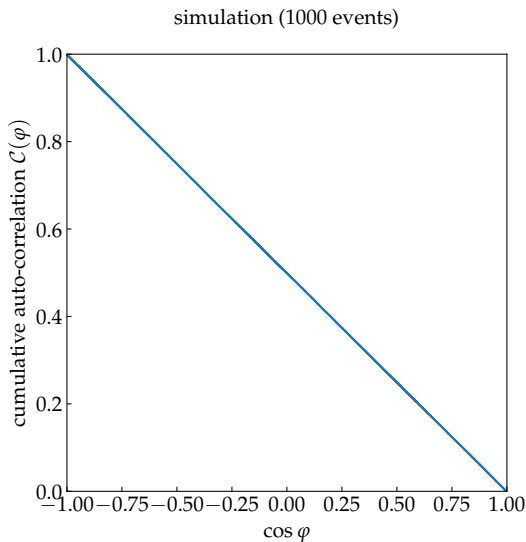
Exercise 2: Isotropic Distribution

simulation (100 events)



for python code see : `twopoint.py`

Exercise 2: Isotropic Distribution



for python code see : `twopoint.py`

Exercise 2: Large-N limit

- In the limit of a large number of events, N_{tot} the cumulative distribution is just given by the relative size of the solid angle $\Delta\Omega$ with half-opening angle φ

$$\lim_{N_{\text{tot}} \rightarrow \infty} \mathcal{C}(\{\mathbf{n}_i\}, \varphi) \rightarrow \mathcal{C}_{\text{iso}}(\varphi) = \frac{\Delta\Omega}{4\pi}$$

- solid angle

$$\Delta\Omega = 2\pi(1 - \cos \varphi)$$

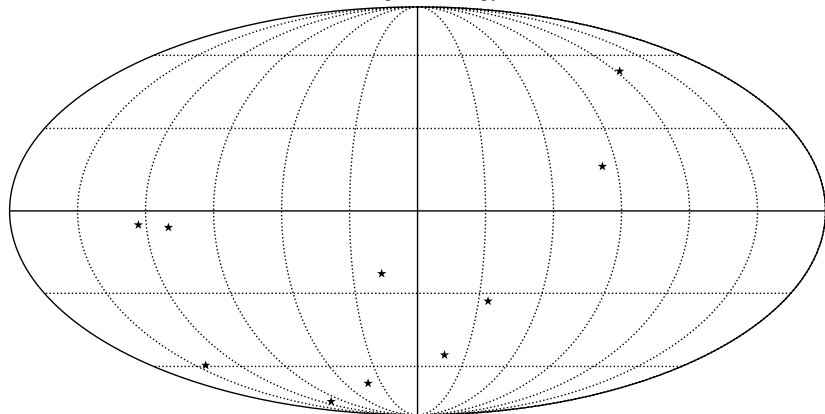
- isotropic distribution:

$$\mathcal{C}_{\text{iso}}(\varphi) = \frac{1}{2}(1 - \cos \varphi)$$

! **Note:** an isotropic distribution of a **finite** number of events will always show deviations from \mathcal{C}_{iso} .

Exercise 2: Anisotropic Distribution

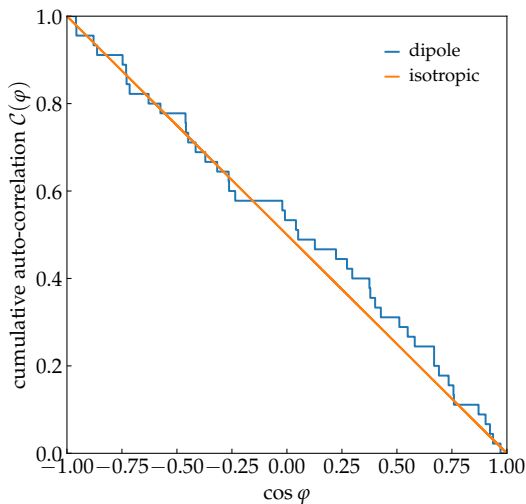
simulation with dipole anisotropy (10 events)



for python code see : `twopoint.py`

Exercise 2: Anisotropic Distribution

simulation (10 events)



for python code see : `twopoint.py`

Kolmogorov-Smirnov (KS) Test

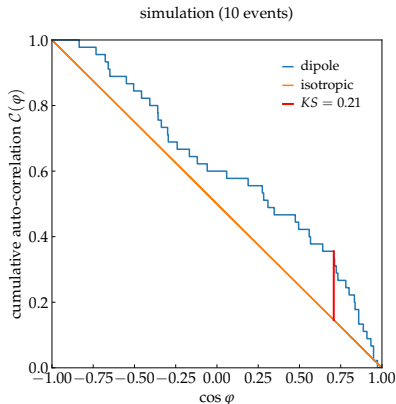
- We want to define a quantity that is a statistical measure for the difference between the empirical distribution and background distribution.
- Area between two curves?

$$\int d \cos \varphi |\mathcal{C}(\{\mathbf{n}_i\}, \varphi) - \mathcal{C}_{\text{iso}}(\varphi)|$$

- Or, more general (L^p norm)?

$$\left[\int d \cos \varphi |\mathcal{C}(\{\mathbf{n}_i\}, \varphi) - \mathcal{C}_{\text{iso}}(\varphi)|^p \right]^{\frac{1}{p}}$$

- **Kolmogorov-Smirnov:** $p \rightarrow \infty$.



Kolmogorov-Smirnov (KS) Test

- In general, given two cumulative probability distributions, $0 \leq A(x) \leq 1$ and $0 \leq B(x) \leq 1$, we can define the **Kolmogorov-Smirnov test** as:

$$KS = \sup_x |A(x) - B(x)|$$

- Cumulative auto-correlation function $\mathcal{C}(\{\mathbf{n}_i\}, \varphi)$ follows the probability distributions to find a pair of events within an angular distance φ .
- We will use this in the following to define a test statistic, that describes **deviation from an isotropic background distribution**:

$$KS(\{\mathbf{n}_i\}) = \sup_{\varphi} |\mathcal{C}(\{\mathbf{n}_i\}, \varphi) - \mathcal{C}_{\text{iso}}(\varphi)|$$

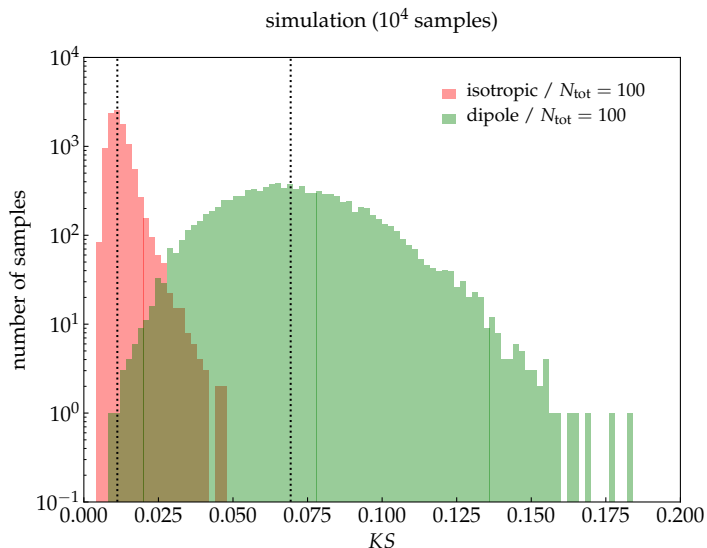
Kolmogorov-Smirnov (KS) Test

- **Plan:** For a fixed number of events N_{tot} we can simulate isotropic event distributions (null hypothesis) and their KS values (test statistic).
- Separation of KS for observed data from background distribution allows to **estimate significance of an excess**.
- number of event pairs increases as

$$N_{\text{pair}} = \frac{1}{2}N_{\text{tot}}(N_{\text{tot}} - 1) \propto N_{\text{tot}}^2$$

- ✗ Cumulative auto-correlation function in Eq. (2) becomes numerically inefficient.

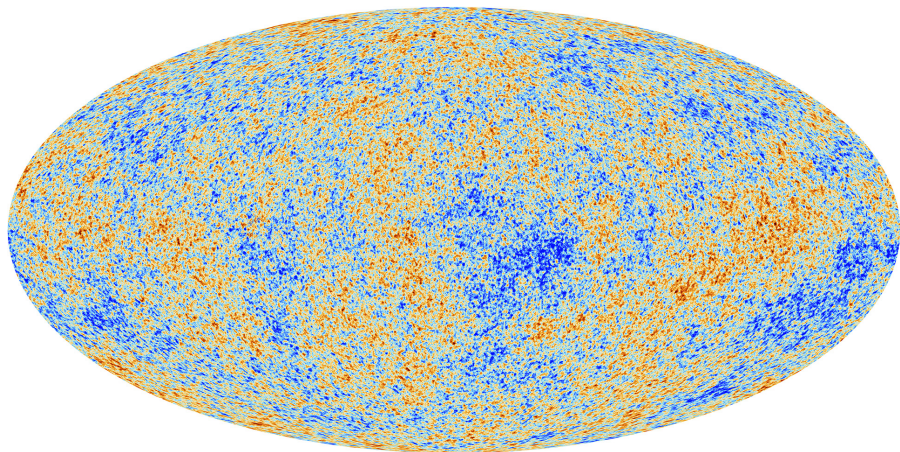
Kolmogorov-Smirnov (KS) Test



for python code see : `KS_produce.py` & `KS_show.py`

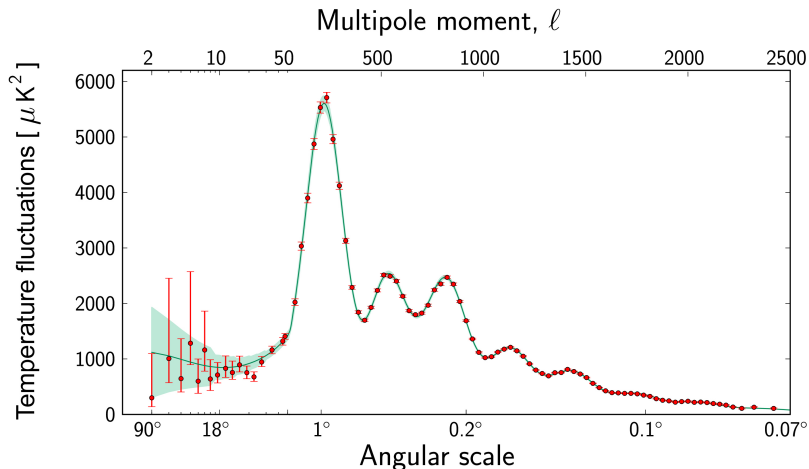
Part III
Angular Power Spectrum
(optional)

Example: Temperature Fluctuation in CMB



Temperature anisotropies of the cosmic microwave background (CMB) observed by the Planck satellite.

Example Temperature Fluctuation in CMB



The angular power spectrum C_ℓ of the temperature fluctuations.

Auto-Correlation for Large N_{tot}

- In the Kolmogorov-Smirnov test we observed that for large N_{tot} the number of pairs increase as N_{tot}^2 and the calculation can become very inefficient.
- In **large- N_{tot} limit** we can approximate the event distribution by a smooth function

$$g(\Omega) = \lim_{N_{\text{bins}} \rightarrow \infty} \frac{\Delta n(\Omega)}{N_{\text{tot}} \Delta \Omega}$$

- On a smooth distribution we can define the **two-point auto-correlation function** as

$$\xi(\varphi) = \int d\Omega_1 \int d\Omega_2 \delta(\mathbf{n}(\Omega_1) \mathbf{n}(\Omega_2) - \cos \varphi) g(\Omega_1) g(\Omega_2)$$

- **Note:** This is the differential version of cumulative auto-correlation function.

Auto-Correlation for Large N_{tot}

- **comment 1** : *cumulative* two-point auto-correlation function:

$$C(\varphi) = \int_{\cos \varphi}^1 d \cos \varphi' \bar{\zeta}(\varphi')$$

- **comment 2** : isotropic distribution $g(\Omega) = 1/(4\pi)$

$$\bar{\zeta}(\varphi) \stackrel{+}{=} \frac{1}{2} \rightarrow C_{\text{iso}}(\varphi) = \int_{\cos \varphi}^1 d \cos \varphi' \frac{1}{2} = \frac{1}{2}(1 - \cos \varphi) \quad (\checkmark)$$

† follows from:

$$\delta(\mathbf{n}(\Omega_1)\mathbf{n}(\Omega_2) - \cos \varphi) = 2\pi \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} P_{\ell}(\cos \varphi) Y_{\ell m}^*(\Omega_1) Y_{\ell m}(\Omega_2)$$

Spherical Harmonics

- Every smooth function $g(\theta, \phi)$ on a sphere can be decomposed in terms of spherical harmonics $Y_{\ell m}$:

$$g(\theta, \phi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(\theta, \phi)$$

- coefficients given by:

$$a_{\ell m} = \int d\Omega Y_{\ell m}^*(\theta, \phi) g(\theta, \phi)$$

→ for real-valued functions:

$$a_{\ell m}^* = (-1)^m a_{\ell -m}$$

Spherical Harmonics

- The low- ℓ components are
 - $\ell = 0$: **monopole** $Y_{00} = 1/\sqrt{4\pi}$
 - $\ell = 1$: **dipole**

$$Y_{10} = \sqrt{\frac{3}{4\pi}} \cos \theta \quad Y_{1-1} = \sqrt{\frac{3}{8\pi}} \sin \theta e^{-i\varphi} \quad Y_{11} = -\sqrt{\frac{3}{8\pi}} \sin \theta e^{i\varphi}$$

- $\ell = 2$: **quadrupole**, $\ell = 3$: **octupole**, etc.
- **angular power spectrum:**

$$C_\ell = \frac{1}{2\ell + 1} \sum_{m=-\ell}^{\ell} |a_{\ell m}|^2$$

- simple relation to ξ via Legendre polynomials P_ℓ :

$$\xi(\varphi) = 2\pi \sum_{\ell} (2\ell + 1) C_\ell P_\ell(\cos \varphi)$$

Exercise 3

- visualize spherical harmonics for various combinations of ℓ and m
- for example, in python use healpy:

```
nside = 128
npix = H.nside2npix(nside)

LMAX = 4*nside
almsize = np.int(((LMAX+2)*(LMAX+1))/2)
alm = np.zeros(almsize, dtype=np.complex)

l = 10
m = 4

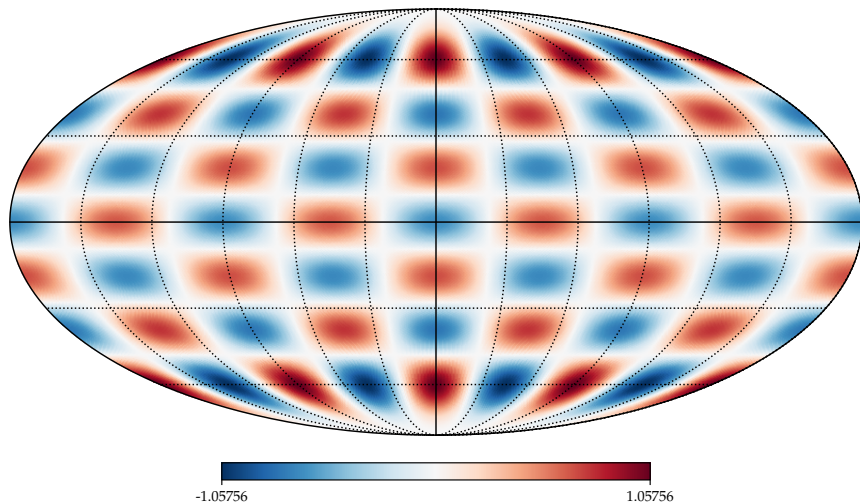
index = H.sphtfunc.Alm.getidx(LMAX, l, m)
alm[index] = 1.0

map = H.alm2map(alm, nside, lmax=LMAX)
mapmax = max(max(map), max(-map))
maptitle = r'$\ell= ' + str(l) + ' $ \& $m= ' + str(m) + '$'

H.mollview(map, cmap=cm.RdBu_r, max=mapmax, min=-mapmax, title=maptitle)
H.graticule()
show()
```

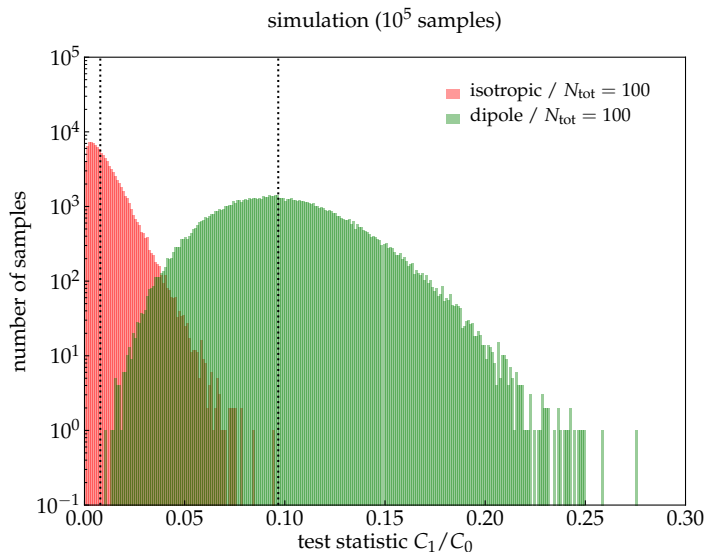
Exercise 3 : Example Map of Spherical Harmonic

$\ell = 10$ & $m = 4$



for python code see : `Ylm.py`

Power Spectrum



for python code see : `C1_produce.py` & `C1_show.py`

Power Spectrum

- In general, we want to judge if a distribution of events shows evidence for an excess in the power spectrum compared to background expectations.
- **Strategy:** Generate background maps from data via scrambling:
 - a) choose two random bins i and j
 - b) interchange the events in the two bins
 - c) repeat from a) until $N_{\text{scramble}} \gg N_{\text{bins}}$
- The distribution of the power spectrum of these maps gives an estimate of the median and variance of the background power.
- Expected median noise level:

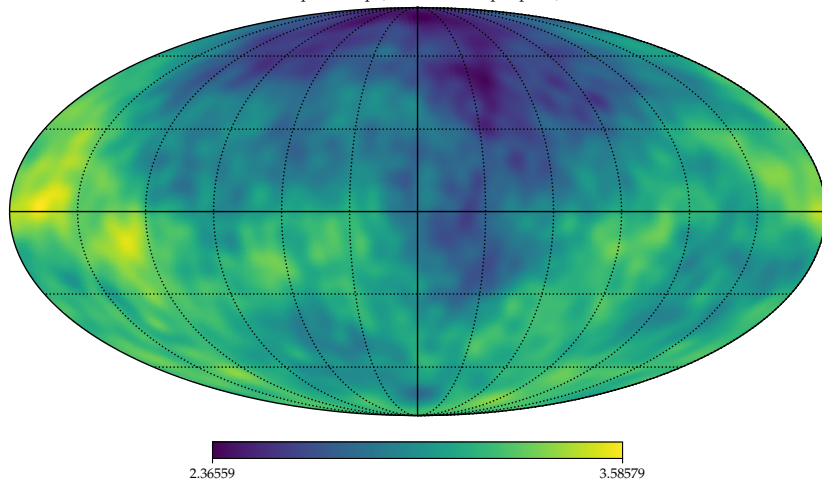
$$\mathcal{N} = \frac{1}{N_{\text{tot}}}$$

Exercise 4

- Load the two data files `truemap1.fits` and `eventmap1.fits` (the second file is a bin-wise Poisson sample with mean given in the first map)
- Display the maps
- Determine and compare the power spectra C_ℓ/C_0 of the two maps, e.g. with `HealPix` or `healpy`
- Generate a background map via data scrambling, as described on the previous slide.
- Compare the power spectrum of the event map to the expected noise level $1/N_{\text{tot}}$.

Exercise 4 : Template vs. Event Map

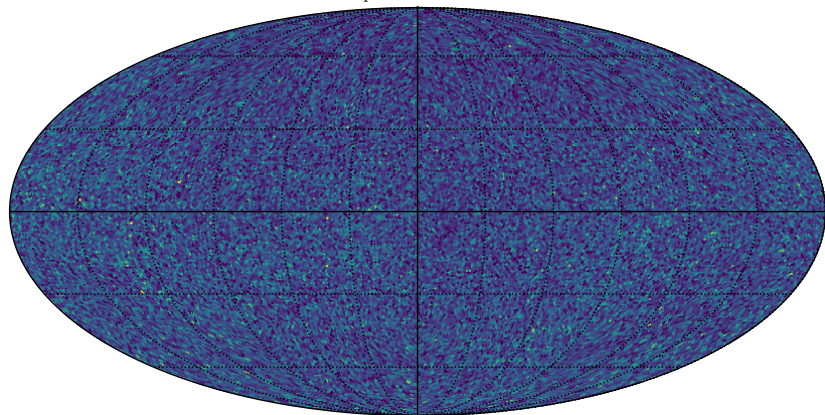
template map (Poisson mean per pixel)



for python code see : `powerspectrum.py`

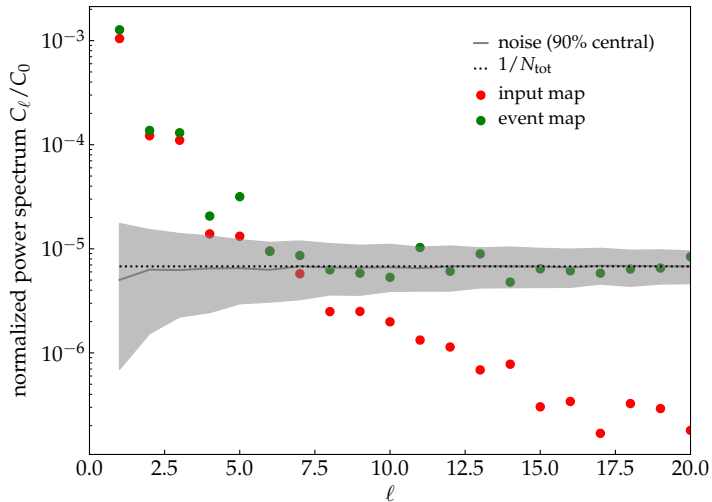
Exercise 4 : Template vs. Event Map

data map with 147473.0 events



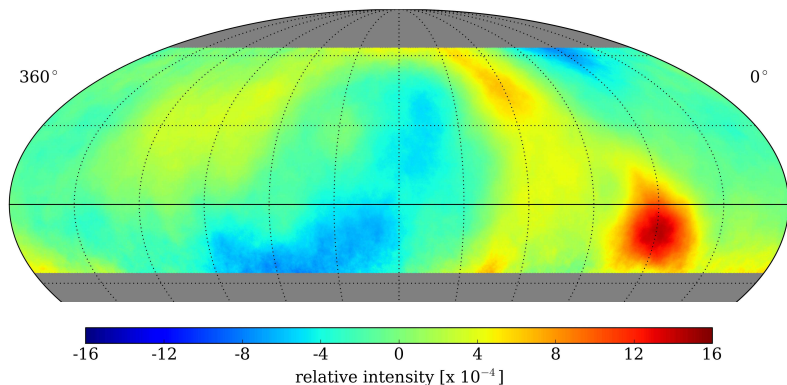
for python code see : `powerspectrum.py`

Exercise 4 : Power Spectra



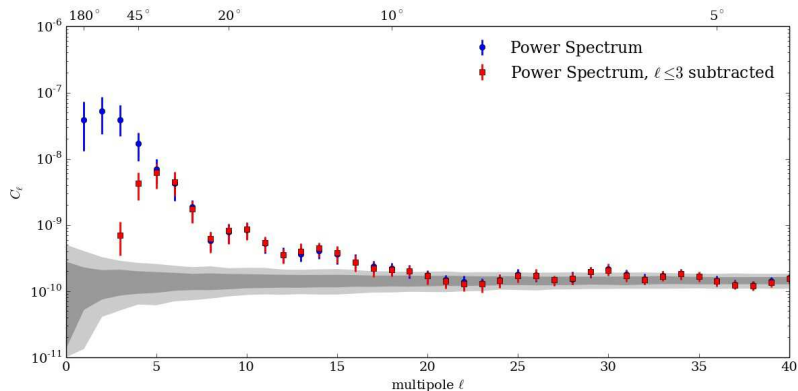
for python code see : `powerspectrum.py`

Example: HAWC Anisotropies



Study of cosmic ray arrival directions with the High Altitude Water Cherenkov (HAWC) detector.

Example: HAWC Anisotropies



Study of cosmic ray arrival directions with the High Altitude Water Cherenkov (HAWC) detector.