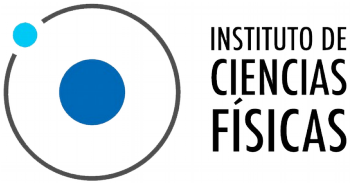
Correlation Function with CUTE
Correlation Utilities and Two-point Estimates

ICF-UNAM
31/07/18

Instituto de Física UNAM

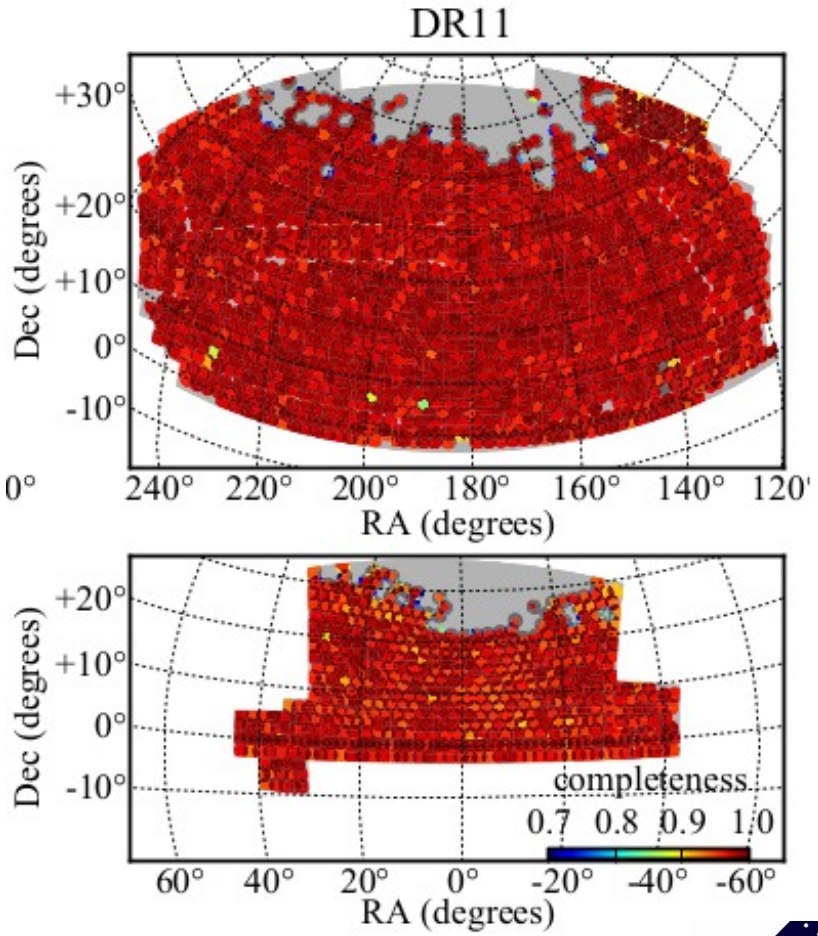
Brenda Izamar Tapia Benavides

IV-Taller de Métodos Numéricos y Estadísticos en Cosmología



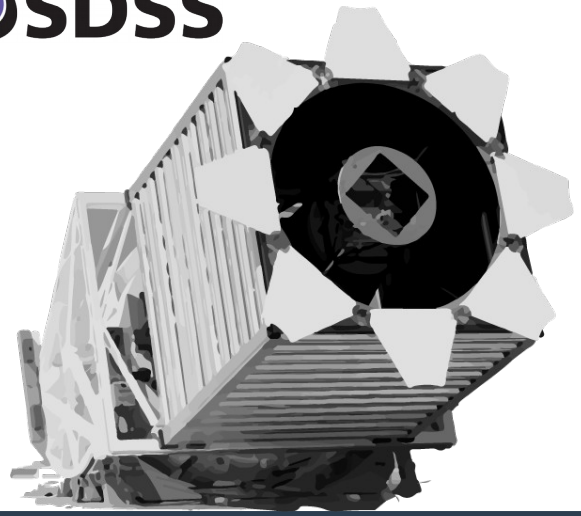
Funciones de Correlación Grandes Sondeos de Galaxias

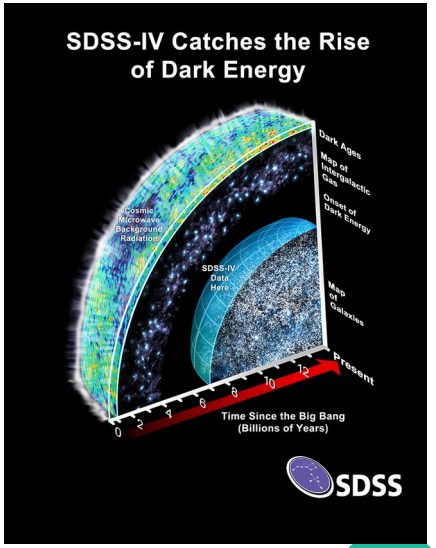
Instituto de Física



- **Se produce una gran cantidad de conjuntos de datos con millones de objetos con los actuales sondeos de galaxias.**
- **Las grandes encuestas de galaxias son un medio para estudiar la evolución de Universo.**

SDSS





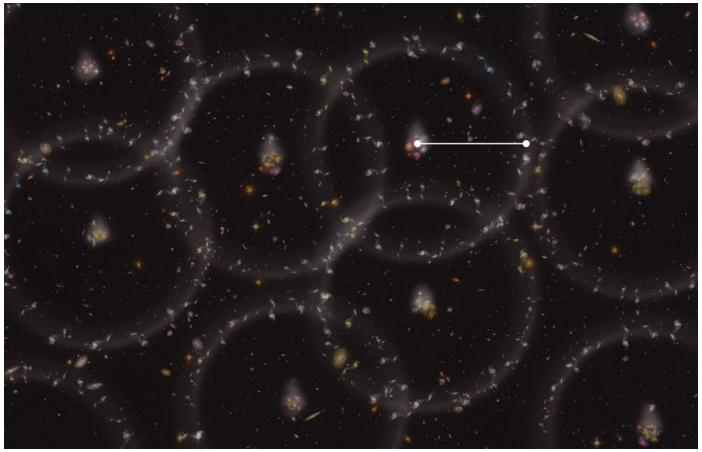
eBOSS

Funciones de Correlación.
Métodos Estadísticos.

Instituto de Física

Función de correlación a dos puntos.

- **La función de correlación de dos puntos, ξ(r), es uno de los observables más simples para cuantificar la agrupación de materia en diferentes escalas.**
- **Se basa en contar pares de objetos separados por una distancia media determinada.**



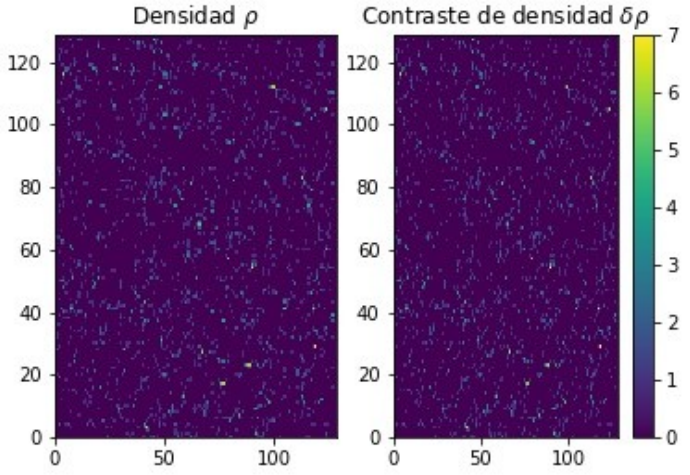
Ejemplo de distribución de objetos.

- **Función de dos puntos tridimensional, ξ(r) 3-D.**

$$\langle dP \rangle = \bar{n}[1 + \xi(\mathbf{r})]\, dV_1\, dV_2.$$

Donde dV_1 , dV_2 son los diferencial de volumen

$$\xi_\delta(\mathbf{r}) \equiv \langle \delta(\mathbf{x})\delta(\mathbf{x}+\mathbf{r}) \rangle$$



Densidad y contraste de densidad de un catálogo de eBOSS

Funciones de Correlación.
Estimador Landy-Szalay.

Instituto de Física

ξ(r) se puede calcular como:

$$1+\xi = \frac{N_p^d(r)\,dr}{N_p^r(r)\,dr}$$

donde N_d es el número de pares separados por r+-dr/2 en los datos, y N_r es el número de pares que uno esperaría encontrar en una distribución aleatoria.

- Considerando un catálogo con límites complicados.

  En este caso se pude usar :

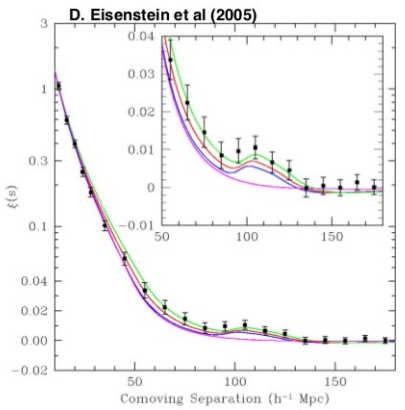
$$\xi_N = \frac{N_r(N_r-1)}{N_d(N_d-1)}\frac{DD}{RR} - 1$$

  donde N_d y N_r son el número de puntos en los datos y en el catálogo random. DD y RR son histogramas que contienen los recuentos de pares de objetos separados por una distancia dada en cada catálogo.

- Haciendo que la varianza del estimador sea mínima se usa la correlación cruzada de objetos aleatorios y de datos, DR.
- Estimador propuesto por Landy & Szalay

$$\xi_{LS} = \frac{\frac{N_r(N_r-1)}{N_d(N_d-1)}DD - \frac{N_r-1}{N_d}DR + RR}{RR}.$$

  donde DD para datos, RR para random, DR datos-random.



CUTE
Correlation Utilities and Two-point Estimates

Instituto de Física

- **Es un código libre y de código abierto para estimaciones cosmológicas de funciones de correlación de dos puntos.**

https://github.com/damonge/CUTE/tree/master/CUTE

- **Escrito en C**
- **Implementación OpenMP y arquitectura CUDA de Nvidia.**
- **Calcula siete diferentes 2PCF.**
- **Puede utilizar 5 diferentes estimadores.**

- **Obtener el catálogo aleatorio.**
- **DD, RR y DR se calculan por autocorrelación entre cada par de catálogos.**
- **Realiza tres operaciones en cada iteración; calcula la distancia entre pares de objetos, determina el conteo a esa distancia y aumenta el conteo de histogramas en ese contenedor.**

```
1  int histogram[nbins];
2  for(i=0;i<np1;i++) {
3    for(j=0;j<np2;j++) {
4      //Calculate distance between two objects
5      double dist=get_dist(x1[i],y1[i],z1[i],
6                           x2[j],y2[j],z2[j]);
7      //Calculate bin number
8      int ibin=bin_dist(dist);
9      //Increase histogram count
10     histogram[ibin]++;
11   }
12 }
```

CUTE
Correlation Utilities and Two-point Estimates

Instituto de Física

- **Paralelización con OpenMP.**

  Para máquinas de memoria compartida multinúcleo.

  El código anterior toma la siguiente forma cuando se paraleliza con OpenMP:

```
int histogram[nbins];
int histo_thread[nbins];
#pragma omp parallel default(none) \
   private(hthread) shared(...) {
   //Initialize private histograms
   for(i=0;i<nbins;i++)
     histo_thread[nbins]=0;
#pragma omp for      //Parallelize loop
   for(i=0;i<npl;i++) {
     for(j=0;j<np2;j++) {
       //Calculate distance between two objects
       double dist=get_dist(x1[i],y1[i],z1[i],
                            x2[j],y2[j],z2[j]);

       //Calculate bin number
       int ibin=bin_dist(dist);
       //Increase histogram count
       histo_thread[ibin]++;
     }
   }
#pragma omp critical {
   //Add private histograms
   for(i=0;i<nbins;i++)
     histogram[i]+=histo_thread[i]
}
}
```

- **Paralelización con CUDA.**

  Realiza las correlaciones en una GPU utilizando arquitectura CUDA de Nvidia.

  CU_CUTE

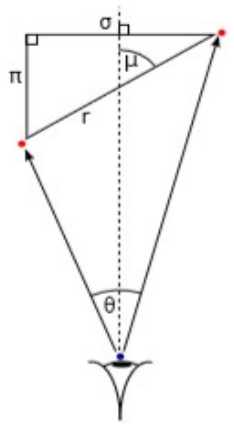
```
__shared__ int histo_thread[nbins];
int stride=blockDim.x*gridDim.x;
//Initialize shared histogram
histo_thread[threadId.x]=0;
__syncthreads();
//Correlate
for(i=0;i<npl;i++) {
  int j=threadIdx.x+blockIdx.x*blockDim.x;
  while(j<np2) {
    //Calculate distance between two objects
    double dist=get_dist(x1[i],y1[i],z1[i],
                         x2[j],y2[j],z2[j]);
    //Calculate bin number
    int ibin=bin_dist(dist);
    //Increase histogram count
    atomicAdd(&(histo_thread[ibin]),1);
    //Increase second index by stride
    j+=stride;
  }
}
//Add block histograms
__syncthreads();
atomicAdd(&(histogram[threadIdx.x]),
          histo_thread[threadIdx.x]);
```

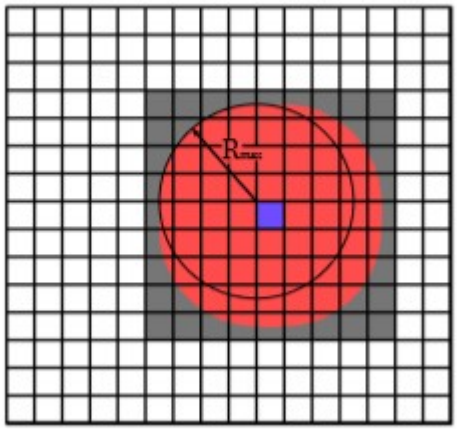
CUTE-Correlation Utilities and Two-point Estimates

Búsqueda de vecinos.

Instituto de Física

- La escala a la que podemos calcular las 2PDF es significativamente más pequeña que el tamaño de nuestros datos.
- Se debe de evitar calcular pares dos veces





- **Caso tridimensional**:

  Se divide en células cúbicas y asociamos las posiciones de los objetos en ellas.

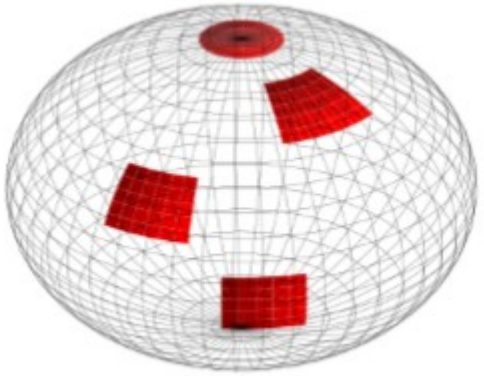
  Dibujamos esferas de radio R_{max} alrededor de cualquier punto.

  Podemos correlacionar todos todos los objetos dentro del cubo y de forma segura ignoramos todos los otros objetos.

- **Caso esférico:**

  Se define una celda esférica con límites constantes, θ. Con lados e longitud

  Usamos pixeles como celdas de manera que tenemos celdas esféricas centradas en pixeles que contienen todas las partículas a la distancia angular θ_max de cualquier partícula.



CUTE-Correlation Utilities and Two-point Estimates
Funciones de Correlación.
Métodos Estadísticos.

Instituto de Física

- **Función de correlación 3D ξ(r,μ), ξ(σ,π).**

$$\pi = r\,\mu, \quad \sigma = \sqrt{r^2 - \pi^2}.$$

$$\xi(r,\mu) = \sum_l \xi_l(r)\, P_l(\mu),$$

donde P_l son los polinomios de Legenre

- **El monopolo $\xi_0(r)$**

$$\xi_0(r) = \frac{1}{2}\int_{-1}^{1} d\mu\, \xi(r,\mu)$$

es el término l=0 de ξ(r,μ)

- **La función de correlación radial, ξ(z,Δz).**

$$\xi_r(\bar{z}, \Delta z) = \xi(\pi(\bar{z}, \Delta z), \sigma = 0),$$

$$\pi(\bar{z}, \Delta z) \simeq \frac{c\Delta z}{H(\bar{z})},$$

Depende de las diferencias de redshift Δz

- **La función de correlación angular, w(θ).**

$$w(\theta) = \langle \delta_s(\hat{\mathbf{n}}_1)\delta_s(\hat{\mathbf{n}}_2)\rangle, \quad \cos\theta = \hat{\mathbf{n}}_1 \cdot \hat{\mathbf{n}}_2, \qquad \delta_s(\hat{\mathbf{n}}) = \int dz\, \phi(z)\, \delta(r(z)\,\hat{\mathbf{n}}),$$

$$w(\theta) = \int dz_1\, \phi(z_1) \int dz_2\, \phi(z_2)\, \xi(r(z_1, z_2, \theta), \mu(z_1, z_2, \theta))$$

$$r(z_1, z_2, \theta) = \sqrt{\chi^2(z_1) + \chi^2(z_2) - 2\,\chi(z_1)\,\chi(z_2)\cos\theta},$$
$$\mu(z_1, z_2, \theta) = \frac{|\chi^2(z_1) - \chi^2(z_2)|}{\sqrt{(\chi^2(z_1) + \chi^2(z_2))^2 - 4\,\chi^2(z_1)\chi^2(z_2)\cos^2(\theta)}}$$

CUTE-Correlation Utilities and Two-point Estimates
Funciones de correlación.

Instituto de Física

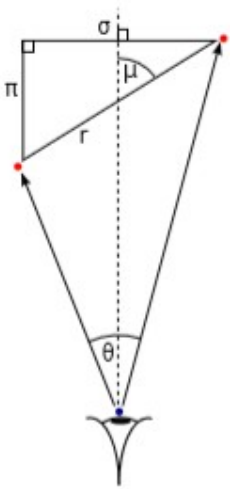
Cada una de las 2PCF requiere un tratamiento diferente de los datos y permite enfoques diferentes.

Función de correlación radial.

- **Es calculada por la correlación de pares de objetos alineados entre ellos.**
- **Se utilizan cajas esféricas.**
- **El conteo es entre pares de galaxias a ángulos pequeños.**

Función de correlación angular.

- **Se correlacionan apares de objetos según su separación angular θ, que se utiliza como una medida de distancia.**
- **Para ángulos pequeños se crea un mapa de pixeles.**
- **Para escalas angulares grandes se calcula:**



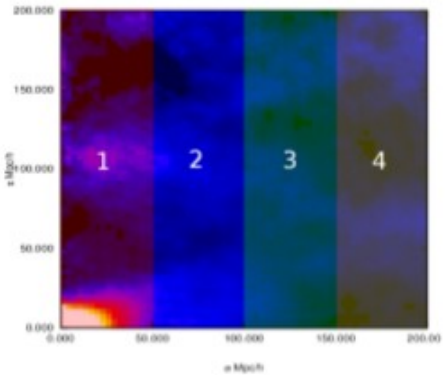
$$\arccos(1-x) \sim \sqrt{2x + \frac{1}{3}x^2 + \frac{4}{45}x^3}$$

CUTE-Correlation Utilities and Two-point Estimates
Funciones de correlación.

Instituto de Física

Función de correlación 2-D

- Aquí los pares se agrupan en histogramas bidimensionales según (r,μ) o (π,σ).
- Se pueden declarar hostogramas 2-D más pequeños en memoria compartida, incluso si los catálogos deben correlacionarse varias veces.
- Tiempos de cálculo razonables.



El monopolo en una caja.

- CUTE-box calcula la función de correlación a partir de datos dentro de una caja cúbica con límites periódicos.
- Solo se calcula la 2PCF isotrópica (el monopolo)
- Calcula 2PCF a partir de recuentos de pares utilizando el estimador

$$1+\xi(r) \equiv \frac{V}{N^2 v(r)} \sum_{i,j \neq i} \Theta(r-dr/2 < |\mathbf{x}_i - \mathbf{x}_j| < r+dr/2)$$

- No se necesita catálogo random, pues las condiciones de frontera son periódicas.

CUTE-Correlation Utilities and Two-point Estimates
Del input al output

Instituto de Física

CUTE necesita un archivo de parámetros que contiene:

- Los archivos de entrada (data, random) y salida; data_filename, random_filename, output_filename.
- El formato de los catálogos; input_format(0,1,2)
- El estimador a utilizar; corr_estimator (PH,DP,HAM,HEW,LS).
- Parámetros cosmológicos; omega_M(materia), omega_L(energía oscura), w(ecuación de estado de energía oscura).

Input. DATA

- Archivos ASCII con 3 o 4 columnas; x0,x1,x2,x3.
- input_format=0,1,2

  0-x0=z,x1=cos(θ),x2=ϕ

  1-x0=z,x1=dec,x2=ra

  2-x0=ra,x1=dec,x2=z
- X3 indica el peso de la galaxia correspondiente

Input RANDOM

- Mismo formato que el archivo data.
- Se puede generar con CUTE, entonces ingresar la máscara.

Output.

- Para la función de correlación radial, angular y el monopolo se obtienen 6 columnas;

  x,xi(x), error(x), DD(x),DR(x),RR(x)
- Para la función de correlación 3-D se obtienen 7 columnas;

  x1,x2,xi(x1,x2),error(x1,x2),DD(x1,x2),DR(x1,x2),RR(x1,x2)

  Donde (x1,x2) son (π,σ)
- Para todas las funciones de correlación y la cross-correlation se tienen cuatro columnas:

  x1,x2,x3,xi(x1,x2,x3),error(x1,x2,x3), DD(x1,x2,x3),DR(x1,x2,x3),RR(x1,x2,x3)

CUTE-Correlation Utilities and Two-point Estimates
Del Input al Output

Instituto de Física

Archivo de parámetros

```
# input-output files and parameters
data_filename= /alicefs/marvam_g/bizb_a/eBOSS_ELG/mocks_recon/qpm_mock_ELG_reconSGC_0099.rdz
random_filename= /alicefs/marvam_g/bizb_a/eBOSS_ELG/mocks_recon/QPM_ELG_randoms20x_reconSGC_0099_shuffle.rdzw
input_format= 2
mask_filename= none
z_dist_filename= none
output_filename= /alicefs/marvam_g/bizb_a/eBOSS_ELG/output_mocks_recon/qpm_mock_ELG_reconSGC_0099.xi

num_lines= all

# estimation parameters
corr_type= 3D_rm
#3D_rm
corr_estimator= LS
np_rand_fact= 50

# cosmological parameters
omega_M= 0.31
omega_L= 0.69
w= -1

# binning
log_bin= 0
n_logint= 10
dim1_max= 200.
dim1_nbin= 200.
dim2_max= 1
dim2_nbin= 100
dim3_min= 0.4  #not used
dim3_max= 0.7 #not used
dim3_nbin= 1 #not used

# pixels for radial correlation
radial_aperture= 1 #not used
```

CUTE-Correlation Utilities and Two-point Estimates

Del input al output

Instituto de Física

Input Data

RA	DEC	z	w
156.585098	26.991519	0.8112672	1.0000000
156.739484	26.656788	0.8164092	1.0000000
156.932208	26.503523	0.8092456	1.0000000
156.925056	26.512722	0.8152930	1.0000000
156.801428	26.720257	0.8173316	1.0000000
156.909462	26.592440	0.8173318	1.0000000
156.879683	26.544148	0.8179059	1.0000000
156.918538	26.531350	0.8155504	1.0000000
156.837452	26.522401	0.8261905	1.0000000
156.913921	26.438556	0.8268342	1.0000000
156.055871	26.990232	0.8565554	1.0000000

Input Random

RA	DEC	z	w
335.939617	1.857266	1.0766716	0.784408
336.048899	1.802211	1.072345	0.778836
335.966717	1.578529	1.0756323	0.78299
336.59046	1.6288	1.0923797	0.806995
336.731253	1.887871	1.0838178	0.794508
336.116424	1.258205	1.0890546	0.802139
336.121735	1.560369	1.0730929	0.779714
336.09783	1.178603	1.0794444	0.788262
335.987359	1.233054	1.0677959	0.77415
335.851937	1.440275	1.0608271	0.767979
336.015014	1.502943	1.0596445	0.766911
336.403601	1.865924	1.053258	0.760266

Output Data

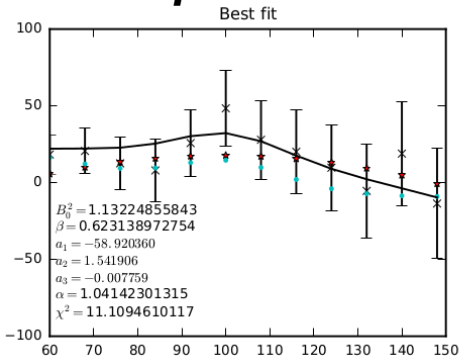
r	μ	ξ(r, μ)	ξ_err(r, μ)	DD	DR	RR
5.000000E-03	5.000000E-01	1.498271E+01	1.070433E+01	5.000000E+00	2.900000E+01	7.370000E+02
1.500000E-02	5.000000E-01	1.138348E+01	8.660519E+00	4.000000E+00	3.800000E+01	7.250000E+02
2.500000E-02	5.000000E-01	1.745835E+00	3.300312E+00	1.000000E+00	3.600000E+01	8.040000E+02
3.500000E-02	5.000000E-01	1.812535E+00	3.384822E+00	1.000000E+00	3.600000E+01	7.380000E+02
4.500000E-02	5.000000E-01	2.091079E+00	3.752151E+00	1.000000E+00	3.200000E+01	7.270000E+02
5.500000E-02	5.000000E-01	2.387872E+00	4.155175E+00	1.000000E+00	2.800000E+01	7.110000E+02
6.500000E-02	5.000000E-01	1.192819E+01	9.474128E+00	4.000000E+00	2.600000E+01	7.420000E+02
7.500000E-02	5.000000E-01	8.528715E+00	7.714428E+00	3.000000E+00	2.600000E+01	7.660000E+02
8.500000E-02	5.000000E-01	7.933187E+00	6.970057E+00	3.000000E+00	3.600000E+01	7.620000E+02
9.500000E-02	5.000000E-01	9.127255E+00	7.954195E+00	3.000000E+00	3.500000E+01	6.560000E+02
1.050000E-01	5.000000E-01	0.000000E+00	0.000000E+00	0.000000E+00	3.500000E+01	7.800000E+02
1.150000E-01	5.000000E-01	8.466273E+00	7.513535E+00	3.000000E+00	3.100000E+01	7.400000E+02

CUTE-Correlation Utilities and Two-point Estimates
Análisis posteriores.

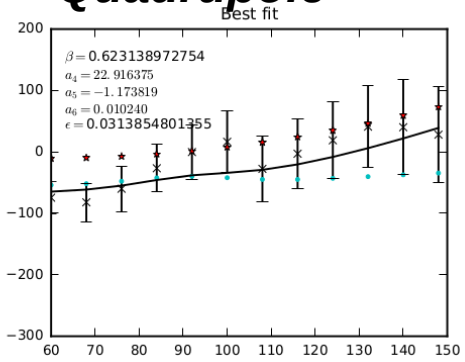
Instituto de Física

qpm_mock_ELGchunk21_SGC_0070

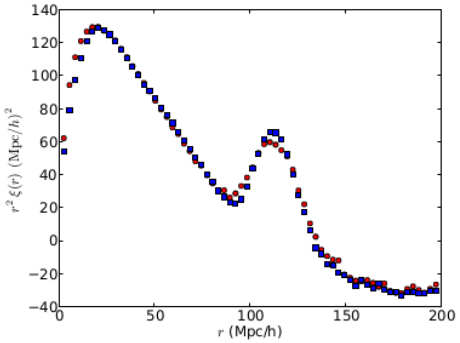
Monopole

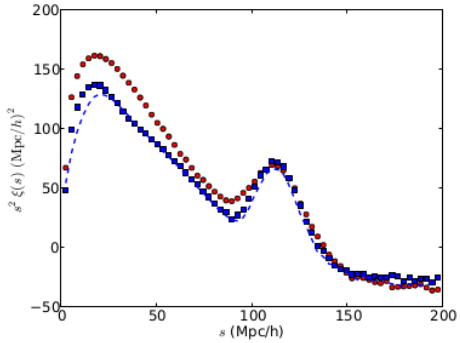


Quadrupole

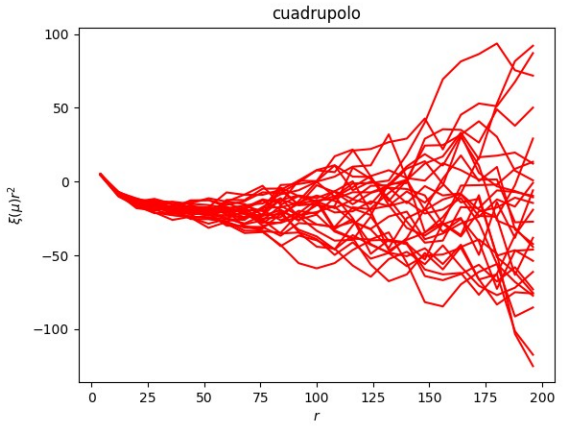


- **Utilizando DD,DR,RR se calcula ξ(r,μ) con el estimador Landy-Szalay**





- **Calculamos el monopolo y cuadrupolo correspondientes.**
- **Oscilaciones Acústicas de Bariones.**
- **Fitting analisis**



CUTE-Correlation Utilities and Two-point Estimates
Instalación

Instituto de Física

- De dónde descargas CUTE:

  https://github.com/damonge/CUTE

  En linux:
- git clone https://github.com/damonge/CUTE.git

Dependencias para compilar y ejecutar CUTE:

- Compilador C gcc
- Bibliotecas GSL
- Bibliotecas OpenMP

Para ejecutar CUTE:

- Generar el ejecutable de CUTE:

  make CUTE
- Correr el programa:

  ./CUTE <param_file>

CUTE-Correlation Utilities and Two-point Estimates

En Resumen

Instituto de Física

- **Calcula funciones de correlación de dos puntos**
- **Utiliza estimadores**

Para Utilizar CUTE:

- **Descargar**
- **Instalar dependencias, modificar el makefile**
- **Archivo de parámetros**
- **Crear ejecutable**
- **Correr el programa**
- **Realizar análisis de los datos**

Referencias:

- **arXiv:1210.1833v2 [astro-ph.IM] 20 Jun 2013.**

  CUTE solutions for two-point correlations from large cosmological datasets. David Alonso.

- **README https://github.com/damonge/CUTE/tree/master/CUTE**