

Astrofísica Computacional - Proyecto 2

Rodrigo Aguilar Meneses

16 de junio de 2020

Resumen

Things were made

1. Método Montecarlo

En síntesis, los métodos Montecarlo por cadenas de Markov son algoritmos que generan distribuciones posteriores muestreando funciones de likelihood de forma representativa en un espacio de parámetros. Estos métodos escalan linealmente con el número de parámetros considerados, por tanto son mucho más eficientes que otros métodos.

El método de Montecarlo por cadenas de Markov utilizado consiste en generar muestras aleatorias en algún espacio y aceptarlas o rechazarlas de acuerdo a su likelihood. La likelihood es la probabilidad de observar los datos que se obtuvieron, y es una medida de la calidad de un ajuste de un modelo estadístico a una muestra de datos, para algún valor dado de los parámetros desconocidos. Si podemos considerar los datos como muestras de variables aleatorias obtenidas a partir de una media, la likelihood es gaussiana. Puesto que los elementos aleatorios generados con Python pueden cumplir con estas características, se trabajó con la siguiente función likelihood:

$$-\ln \mathcal{L}(\vec{x}, \vec{y} | \vec{\theta}) = \frac{1}{2} \sum_i \left(\frac{(y_i - \lambda(x_i, \vec{\theta}))^2}{\sigma_i^2} \right) \quad (1)$$

La idea del método de Montecarlo por cadenas de Markov (MCMC) es maximizar el valor de la likelihood de acuerdo a los siguientes criterios:

- Se acepta la muestra si la likelihood de una muestra es mayor que la de la previa.
- Si la likelihood de una muestra es menor que la previa, se obtiene un número aleatorio, y si la razón de la nueva likelihood a la vieja es mayor que ese número, la aceptamos.

Una buena representación gráfica del método se muestra en las figuras 1 y 2. Se parte de algún punto \vec{p} en el espacio de parámetros y se obtiene su

likelihood. Después se propone aleatoriamente otro punto hacia el cual avanzar, y de acuerdo a las condiciones previas se determina si se avanza hacia éste o no, repitiendo los pasos hasta que se avanza. Se espera que partiendo desde distintos puntos iniciales, las cadenas converjan al mismo punto.

Dado que comenzar la cadena de Markov en cualquier punto del espacio de parámetros debería resultar en convergencia, usualmente se elije deshechar el primer 10 % de las iteraciones (burning process).

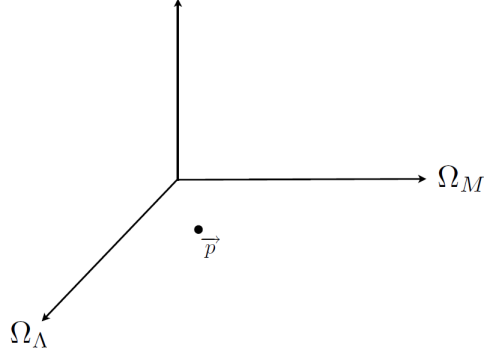


Figura 1: Estado inicial.

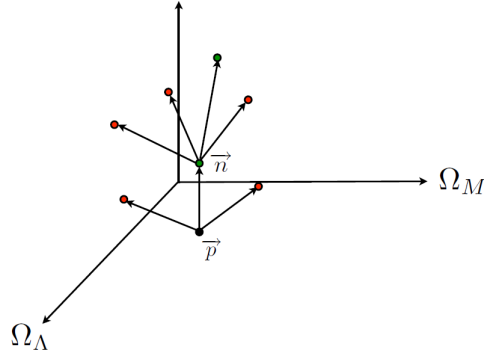


Figura 2: Estado después de una iteración exitosa.

2. Análisis de Supernovas

Se trabajó con un conjunto de datos obtenidos de [1], donde se pretenden obtener restricciones cosmológicas a partir de un análisis conjunto de supernovas de tipo Ia realizado por las colaboraciones SDSS-II y SNLS. El estimador de distancias en este análisis está cuantificado por un modelo lineal para el módulo de distancia μ , dado por

$$\mu = m_B^* - (M_B - \alpha \times X_1 + \beta \times C) \quad (2)$$

donde m_B^* corresponde al pico observado para la magnitud en un sistema de referencia en reposo, y α , β y M_B son parámetros de ruido en la estimación de distancia que dependen de las propiedades de la galaxia huésped. Estos son los parámetros que nos interesa determinar. A partir de las observaciones empíricas, se definen los parámetros X_1 y C , correspondientes a la elongación temporal de la curva de luz y al color de la supernova en su brillo máximo respectivamente.

3. Resolución por MCMC

Se implementó el algoritmo discutido en Python utilizando sólo las librerías convencionales (numpy, scipy, etcétera). Se corrieron un total de 8 cadenas con 10000 pasos cada una, partiendo de los siguientes valores iniciales (α , β , M_B , Ω):

1. 0.16, 3.36, -19.05, 0.29
2. 0.12, 2.48, -19.10, 0.26
3. 0.13, 2.92, -19.22, 0.24
4. 0.14, 2.52, -19.01, 0.22
5. 0.15, 3.01, -19.15, 0.23
6. 0.11, 3.15, -18.96, 0.27
7. 0.18, 2.73, -19.34, 0.21
8. 0.17, 3.50, -19.55, 0.25

La convergencia obtenida para α y β se muestra en la figura 3. Por otro lado, la convergencia individual de cada parámetro iteración a iteración se muestra en las figuras 4, 5, 6 y 7 sin burning de datos, y la triangle plot de los resultados se presenta en la figura 8.

A partir de esta última, podemos fijar los siguientes valores (sin incluir la incertidumbre):

- $\alpha = 1.3$
- $\beta = 1.63$
- $M_B = -19.4$
- $\Omega = 0.82$

Para α y β , podemos ver en la figura su alta incertidumbre, puesto que cada parámetro tiene al menos 2 picos de altura similar. Para M_B sí existe un pico definido con mayor altura que los demás, pero aún así hay otros dos que están sobrando. Finalmente, para Ω se tiene una curva con forma deseable, sin embargo es demasiado ancha. Los valores que se debieron haber obtenido son

- $\alpha = 0.14$
- $\beta = 3.1$
- $M_B = -19.5$
- $\Omega = 0.5$

Incluyendo la incertidumbre en los valores obtenidos, los valores teóricos están en su intervalo, pero la incertidumbre sigue siendo muy grande.

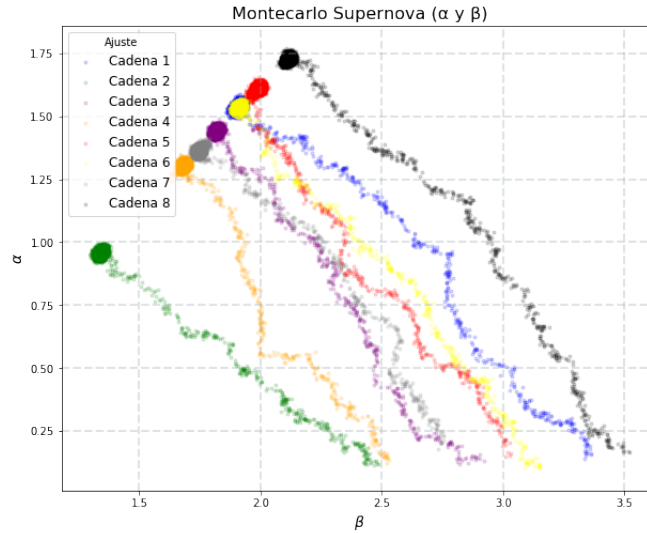


Figura 3: Convergencia de las cadenas para α y β .

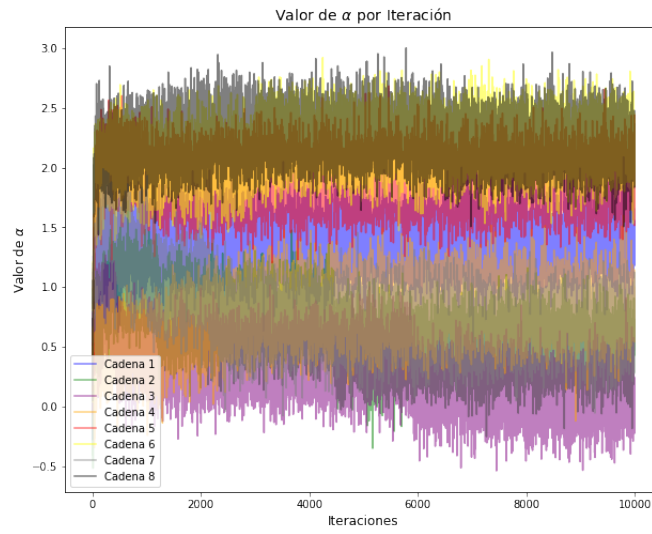


Figura 4: Convergencia iteración a iteración para α .

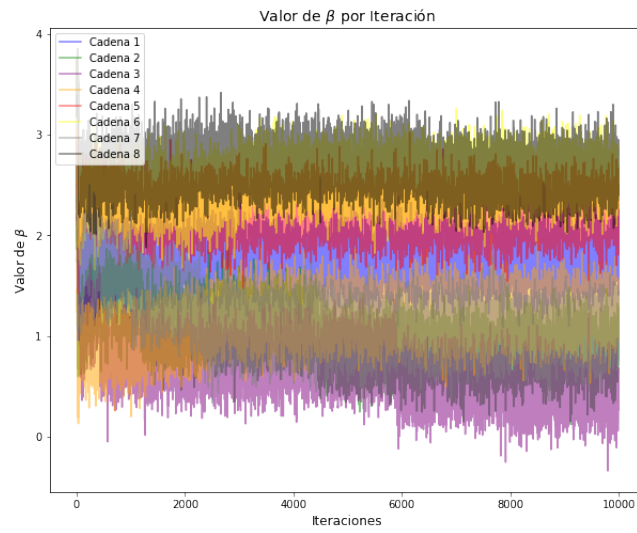


Figura 5: Convergencia iteración a iteración para β .

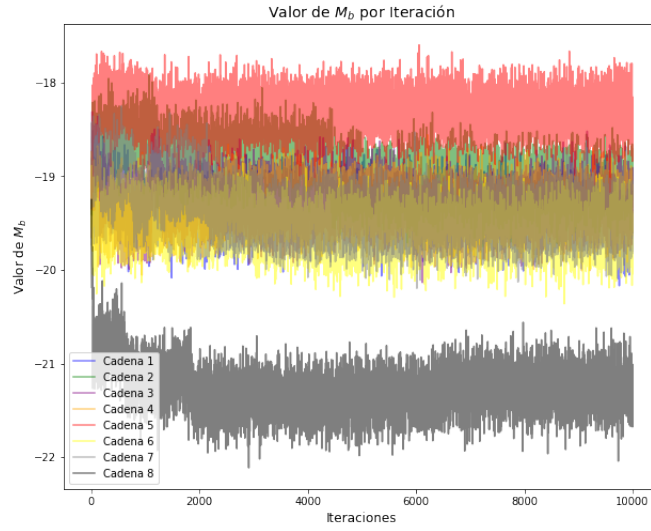


Figura 6: Convergencia iteración a iteración para M_B .

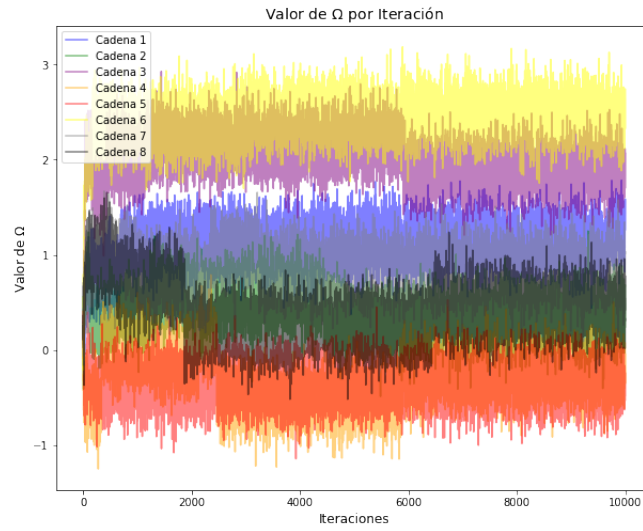


Figura 7: Convergencia iteración a iteración para Ω .

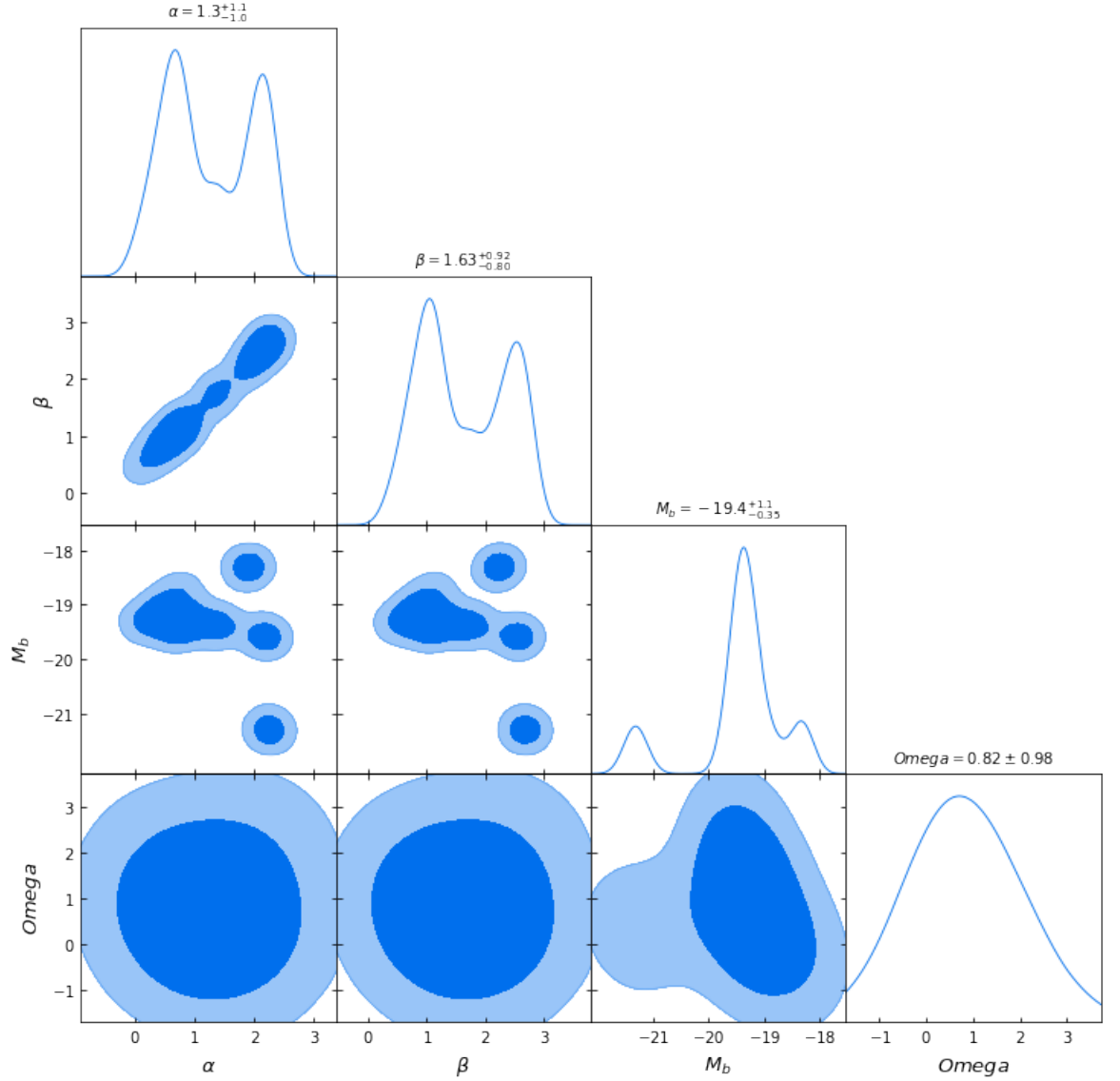


Figura 8: Triangle plot para el MCMC de supernovas en Python.

4. Resolución por EMCEE

En segundo lugar, se realizó el mismo análisis utilizando la librería `emcee` de Python. Esta librería proporciona funciones con las que es posible realizar el algoritmo MCMC. En este caso se corrieron 8 cadenas con 5000 datos cada una, partiendo de los valores iniciales

- 0.16, 3.36, -19.05, 0.29
- 0.12, 2.48, -19.10, 0.26
- 0.13, 2.92, -19.22, 0.24
- 0.14, 2.52, -19.01, 0.22
- 0.15, 3.01, -19.15, 0.23
- 0.11, 3.15, -18.96, 0.27
- 0.18, 2.73, -19.34, 0.21
- 0.17, 3.50, -19.55, 0.25

correspondientes a α , β , M_B , Ω . Nuevamente tenemos la convergencia para α y β en la figura 9. Las convergencias individuales corresponden a las figuras 10, 11, 12 y 14 sin burning de datos. La triangle plot de los resultados se muestra en la figura

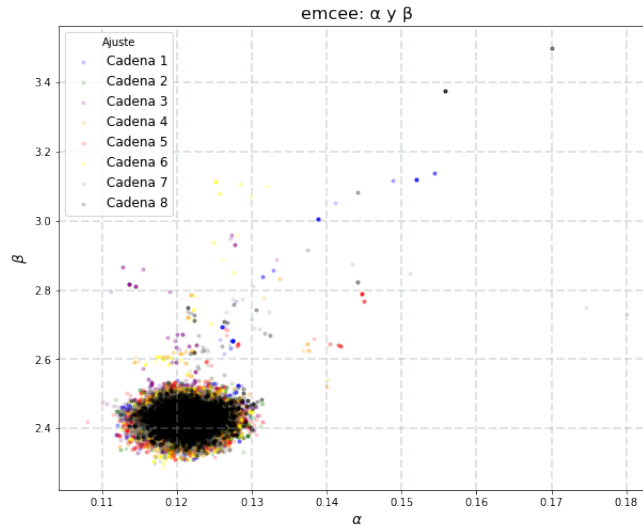


Figura 9: Convergencia de los parámetros α y β por emcee.

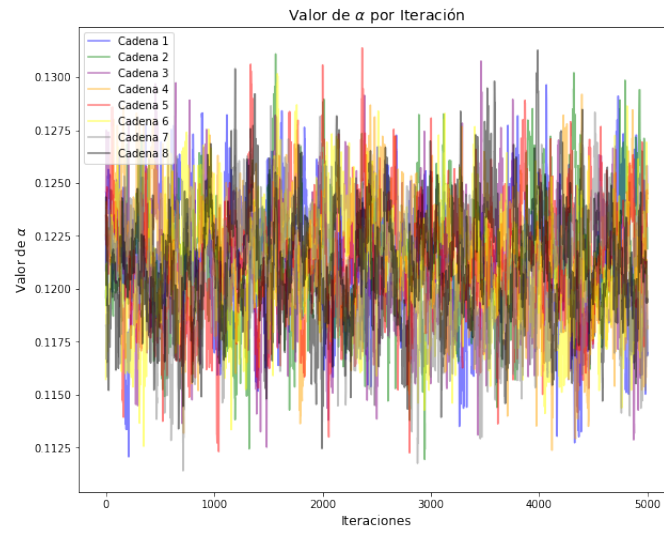


Figura 10: Convergencia iteración a iteración para α .

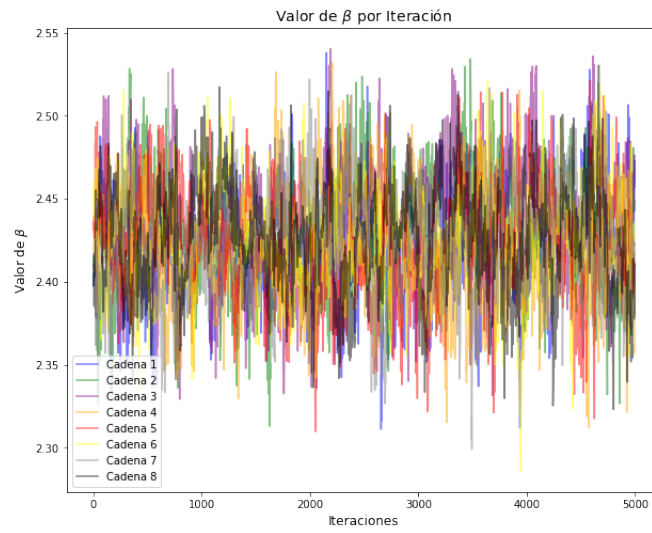


Figura 11: Convergencia iteración a iteración para β .

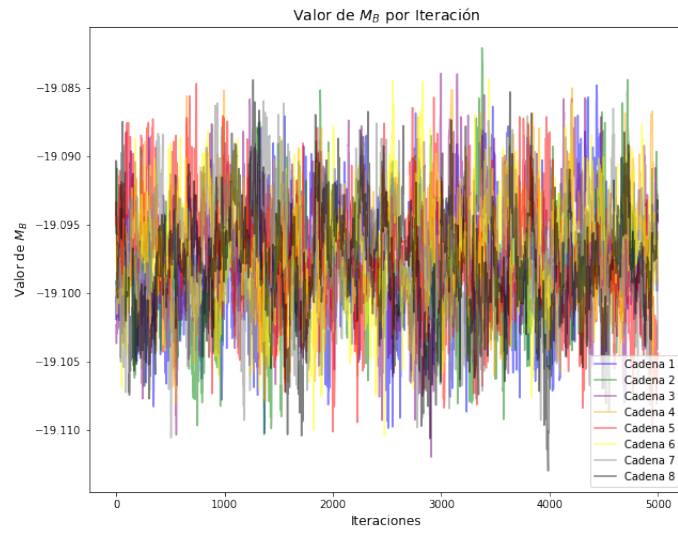


Figura 12: Convergencia iteración a iteración para M_B .

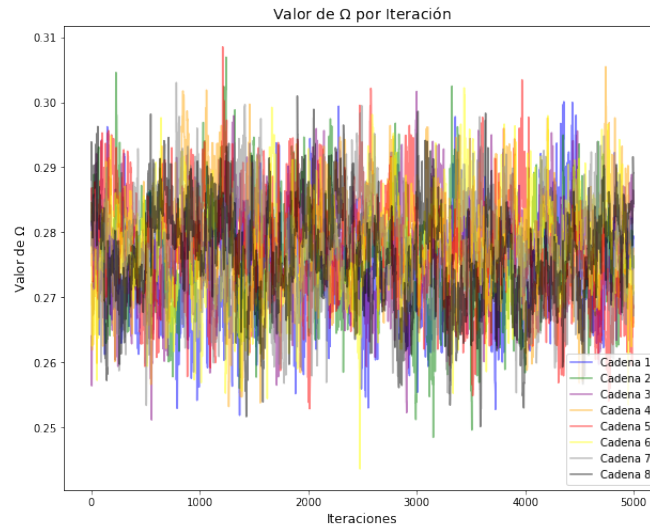


Figura 13: Convergencia iteración a iteración para Ω .

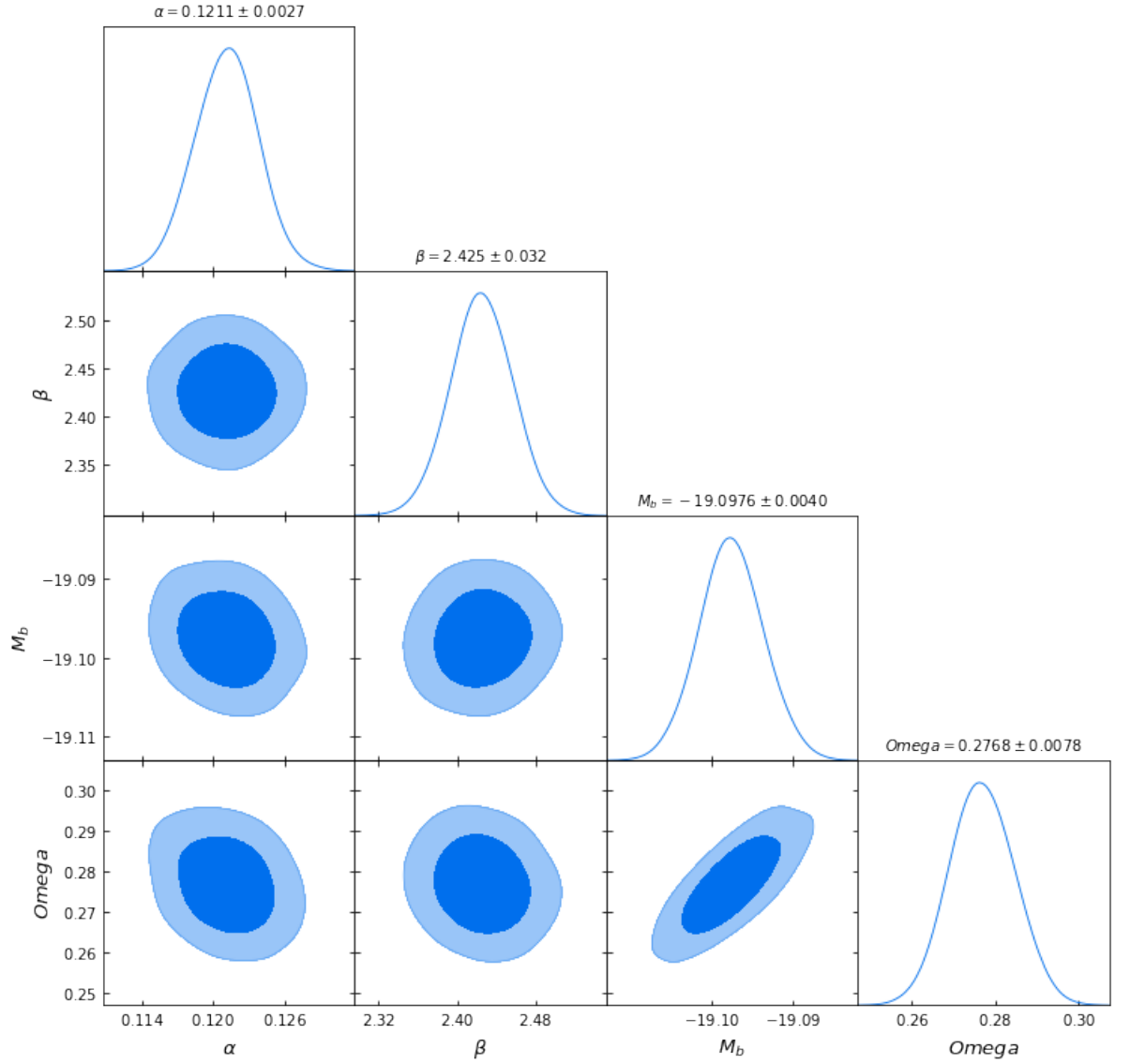


Figura 14: Triangle plot de los resultados por emcee.

5. Referencias

- *Improved cosmological constraints from a joint analysis of the SDSS-II and SNLS supernova samples.* June 5, 2014. SDSS-II & SNLS Collaborations.