



**Facultad
de
Ciencias**

**RECUPERACIÓN DE MAPAS DEL FONDO
CÓSMICO DE MICROONDAS MEDIANTE
MÉTODOS LINEALES
(Recovery of Cosmic Microwave Background
maps by linear methods)**

**Trabajo de Fin de Grado
para acceder al**

GRADO EN FÍSICA

Autor: David Fernández Corral

Director: Patricio Vielva Martínez

Junio - 2017

Índice general

Resumen/Abstract	1
1. Introducción	3
1.1. El modelo cosmológico estándar	3
1.2. El fondo cósmico de microondas (FCM)	4
1.3. Contaminantes en el cielo de microondas	6
1.3.1. Contaminantes galácticos	6
1.3.2. Contaminantes extragalácticos	8
2. Métodos de separación de componentes	11
2.1. FastICA	12
2.2. Modificación del algoritmo de FastICA	14
2.3. ILC	15
2.4. Casos estudiados en este trabajo	17
3. Estudio de simulaciones 1-D	19
3.1. Separación de 3 componentes no gaussianas	22
3.2. Separación de componentes con una señal gaussiana	23
3.3. Separación de componentes con dos gaussianas	24
3.4. Discusión	25
4. Estudio de simulaciones 2-D: CORE	27
4.1. Espectro angular de potencias	29
4.2. Análisis previo: observaciones sin ruido	31
4.3. Análisis de los datos con ruido	35
4.4. Análisis de los datos con ruido x10	37
4.5. Discusión	39
5. Análisis de datos de Planck	41
Conclusiones	45
A. Modificación del algoritmo de FastICA por Newton-Raphson	49
B. Código para señales 1-D	51
C. Código para 2-D: CORE y Planck	63

Resumen

El presente trabajo aborda el problema de la separación de componentes del cielo de microondas a través de la búsqueda y aplicación de métodos lineales. Dentro de las diversas componentes del cielo en el rango de las microondas, nos hemos centrado en la recuperación de la señal del fondo cósmico de microondas debido a su gran importancia en cosmología. En este estudio de la separación de componentes, se ha comenzado por la separación de unas simples señales unidimensionales, para posteriormente aplicar nuestros métodos lineales a datos del cielo de microondas, cuya complejidad se ha aumentado progresivamente.

Palabras clave: Cosmología, fondo cósmico de microondas, separación de componentes, análisis de componentes independientes.

Abstract

The aim of the present project is to study the component separation of the microwave sky by finding and applying linear methods. Among the many components of the microwave sky, this project focuses on the recovery of the cosmic microwave background component due to its importance in cosmology. In our study of the component separation problem, we increase the complexity of the data, starting with the separation of one-dimensional signals from a simple mix and ending with the component separation from data of the ESA's satellite Planck.

Key words: Cosmology, cosmic microwave background, component separation, independent component analysis.

Capítulo 1

Introducción

1.1. El modelo cosmológico estándar

El modelo cosmológico estándar describe la historia y formación de la estructura del Universo, originado hace 13.7×10^9 años. Este modelo se basa en el escenario del Big-Bang caliente [1], que establece que, en su origen, el Universo se encontraba en una fase extremadamente caliente y densa a partir de la cual comenzó a expandirse. Este proceso de expansión y enfriamiento del Universo ha continuado hasta alcanzar finalmente el estado frío y poco denso actual.

El paradigma actual más aceptado indica que la expansión estuvo desencadenada por un fenómeno de gran importancia conocido como inflación cósmica. En los primeros 10^{-32} s de su historia, el Universo experimentó una expansión cuasi-exponencial donde aumentó su tamaño en unos 26 órdenes de magnitud. Este breve periodo de rápida expansión explicaría por qué vivimos en un Universo plano, pues eliminó cualquier posible curvatura inicial, y espacialmente homogéneo e isótropo (hoy en día esto se manifiesta a escalas por encima de unos cientos de Mpc^1). Además, la inflación jugó un papel muy importante en la formación de la estructura del Universo, convirtiendo las fluctuaciones cuánticas primordiales en fluctuaciones (prácticamente gaussianas) en la densidad de energía. Éstas serían las semillas que más tarde, por inestabilidad gravitacional, darían lugar a las galaxias que hoy observamos en el cielo.

Además de a la inflación cósmica, el modelo cosmológico estándar atribuye la formación de la estructura observada en el Universo a la materia oscura (hipotética materia débilmente interaccionante cuya manifestación principal es a través de la gravedad). El fondo cósmico de microondas o el efecto lente gravitatorio son dos pruebas cosmológicas que no pueden ser explicadas sin una contribución significativa de esta materia. Otro ingrediente del modelo cosmológico estándar es la energía oscura, responsable de la expansión acelerada actual del Universo. A pesar de no que aún no ha sido entendida a nivel fundamental, la energía oscura se comporta como un fluido con una presión negativa y las medidas actuales la hacen compatible con la constante cosmológica Λ .

Por último mencionar que la expansión del Universo está gobernada por las ecuaciones de Friedmann, que son solución de las ecuaciones de campo de Einstein al considerar un Universo homogéneo e isótropo. Dichas ecuaciones dependen de algunos parámetros cosmológicos como el parámetro de Hubble H_0 , la curvatura k del Universo o el contenido energético del mismo. Así, las densidades de materia Ω_m , radiación Ω_r y energía oscura Ω_Λ determinan la historia del Universo en cada época. En la actualidad se estima que la curvatura espacial de nuestro

¹1pc \approx 3.26 años-luz

Un universo es muy próximo a cero ($|\Omega_K| < 0.05$) y la densidad de materia es $\Omega_m = 0.308 \pm 0.012$ [2], de la cual aproximadamente el 80 % es materia oscura y el 20 % restante materia bariónica.

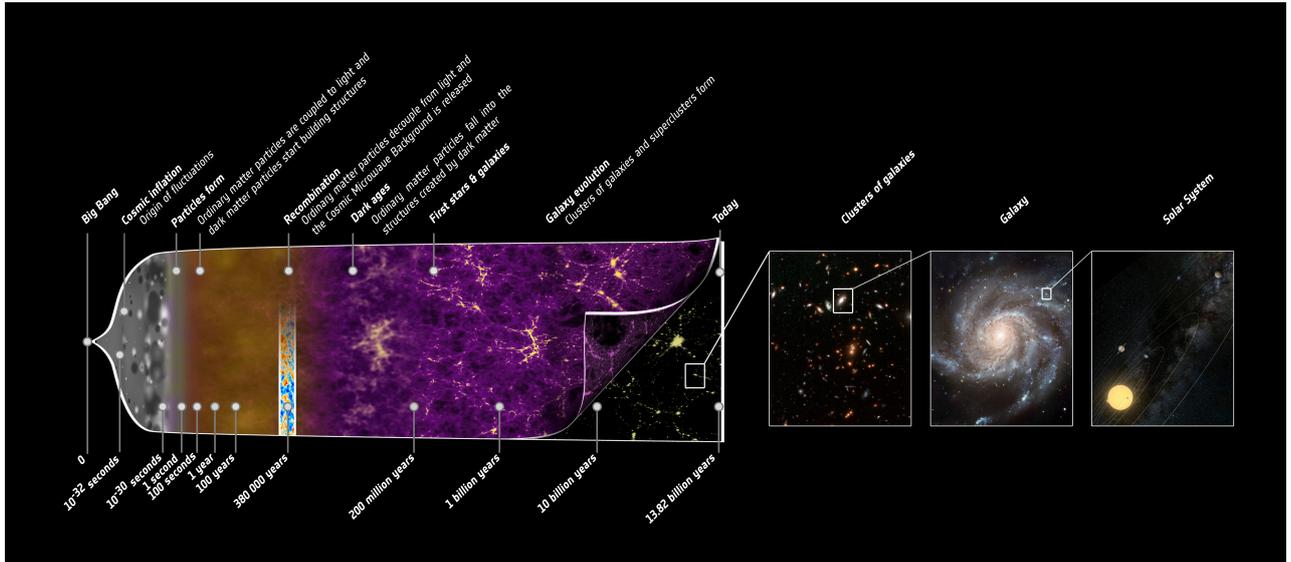


Figura 1.1: Historia de la formación de la estructura del Universo. Fuente: www.esa.int/.

1.2. El fondo cósmico de microondas (FCM)

El Universo primitivo estaba formado por un fluido de partículas y fotones en equilibrio térmico denominado plasma [3]. La radiación interactuaba de manera continua con la materia mediante scattering Thompson, teniendo lugar con mayor intensidad en aquellas zonas en las que la densidad de materia era mayor. Cuando el Universo tenía una edad de 380 000 años aproximadamente, la expansión había disminuido tanto la temperatura (~ 3000 K) que la interacción materia-radiación casi disminuyó por completo. Los electrones comenzaron a combinarse con núcleos de elementos ligeros (principalmente hidrógeno) formando los primeros átomos neutros. Este periodo es conocido como la época de la recombinación. Hasta ese momento, el Universo era opaco debido a que los electrones dispersaban los fotones impidiendo que la luz recorriera grandes distancias. El desacoplamiento entre materia y radiación volvió transparente el Universo y los fotones comenzaron a propagarse en todas direcciones. Esta radiación remanente del Big-Bang que llena todo el Universo se conoce como fondo cósmico de microondas (FCM). El corrimiento al rojo cosmológico de los fotones del FCM derivado de la expansión del Universo hace que su pico de emisión se encuentre actualmente en el rango de las microondas.

El FCM fue descubierto en 1965, de forma accidental, por los ingenieros Arno Penzias y Robert Wilson en los Laboratorios Bell. En 1992 el satélite de la NASA, COBE (en concreto, el experimento *FIRAS*) caracterizó el espectro de emisión del FCM como el de un cuerpo negro casi perfecto, correspondiente a una temperatura de 2.725 K. Así mismo, COBE (en concreto, el experimento *DMR*) obtuvo el primer mapa de las anisotropías presentes en el FCM a gran escala. Estas anisotropías son del orden de 10^{-5} y tienen su origen en la época de la recombinación, como se ha mencionado anteriormente, cuando los núcleos atraparon electrones libres para formar átomos neutros. Además de las anisotropías primarias, existe otro tipo de anisotropías secundarias debidas a interacciones electromagnéticas y gravitacionales de los fotones en su recorrido hasta nosotros. En 2001 el satélite WMAP exploró el

cielo de microondas aportando información sobre las fluctuaciones en temperatura y polarización [4]. Posteriormente el satélite de la ESA Planck se encargó de obtener imágenes de las anisotropías en temperatura y polarización del FCM con una sensibilidad y resolución sin precedentes [5]. Junto con estos satélites, el campo también se ha visto beneficiado de la existencia de varios experimentos tanto terrestres como en globos estratosféricos: BOOMERang, MAXIMA, VSA, BICEP [6].

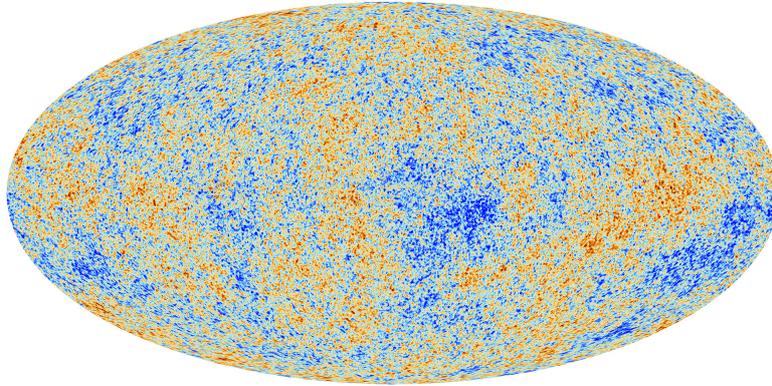


Figura 1.2: Imagen del FCM obtenido por el satélite de la ESA Planck. Fuente: www.esa.int/.

La importancia del FCM radica en que se trata de la radiación electromagnética más antigua del Universo y que, por tanto, no sólo nos proporciona información sobre el Universo temprano, sino que también proporciona estas...su evolución. Hay que notar que cuando miramos a las estrellas del cielo nocturno estamos mirando al pasado. De igual forma cuando observamos el FCM estamos viendo cómo era el Universo cuando las estrellas y las galaxias actuales aún no se habían formado. Las fluctuaciones de los fotones del FCM contienen información importante sobre el Universo en sus primeras etapas. En particular, a través de los mapas de las fluctuaciones del FCM y de su espectro angular de potencias se pueden determinar los parámetros cosmológicos, de los cuales dependen la dinámica y el contenido del Universo. Con esta información podríamos distinguir entre las diversas teorías sobre la formación de la estructura en el Universo, entender cómo funciona el mecanismo que está detrás de la inflación cósmica o aumentar el conocimiento que se tiene sobre la materia y la energía oscuras.

En la actualidad, los mayores esfuerzos de cara al estudio del FCM se centra en las anisotropías en polarización. Durante la época de la recombinación, la interacción de la radiación con la materia por dispersión Thompson generó polarización lineal en los fotones del FCM. Cuando la radiación incide sobre una partícula cargada, ésta vibra en la dirección del campo eléctrico, perpendicular a la dirección de propagación de la onda, emitiendo radiación. Sobre los electrones del plasma primordial incidía radiación en todas direcciones. Si la intensidad de la radiación hubiese sido igual en todas ellas, la luz dispersada por los electrones era despolariada, ya que los electrones vibraban con la misma intensidad en todas direcciones. Si, por el contrario, la radiación era más intensa en alguna dirección, los electrones vibraban más en un plano que en los demás, dando como resultado la dispersión de luz linealmente polarizada. Las fluctuaciones de radiación que pueden producir esta polarización son de tipo cuadrupolar.

En la recombinación hay tres fuentes que inducen anisotropía cuadrupolar; las perturbaciones de densidad, la vorticidad y las ondas gravitacionales. Las perturbaciones de densidad hacen que el movimiento de los fotones de las zonas más calientes a las más frías produzca un momento cuadrupolar. La anisotropía cuadrupolar originada por la vorticidad en el plasma

es diferente a la anterior, es debida al efecto Doppler y se anula. Por último, la expansión acelerada del Universo durante la inflación habría originado perturbaciones en el espacio-tiempo en forma de ondas gravitacionales que habrían dejado su huella en la polarización de los fotones del FCM. Es por ello que a través del estudio de las anisotropías en polarización del FCM se podría inferir información importante sobre la inflación.

1.3. Contaminantes en el cielo de microondas

Como se ha mencionado, numerosos experimentos desde bases terrestres, globos estratosféricos y satélites han proporcionado una visión completa del cielo en el rango de las microondas. El cielo de microondas no sólo contiene información sobre el FCM sino también sobre algunos procesos astrofísicos que tienen lugar en la Vía Láctea y fuera de ella. El problema fundamental que plantea la estimación del FCM es la eliminación de las emisiones contaminantes o foregrounds, de los que se puede encontrar un breve resumen en [7].

La importancia relativa de las componentes del cielo de microondas depende de varios factores, como son la frecuencia de observación, la latitud y la escala angular. Así, a latitudes bajas, a escalas angulares grandes ($> 1^\circ$) y a frecuencias bajas y altas, los contaminantes galácticos dominan sobre el FCM. Esto se debe a que la materia del medio interestelar se concentra en el plano galáctico. En cambio, a latitudes altas y frecuencias intermedias es el FCM quien domina sobre los contaminantes galácticos, cuya intensidad disminuye con la latitud.

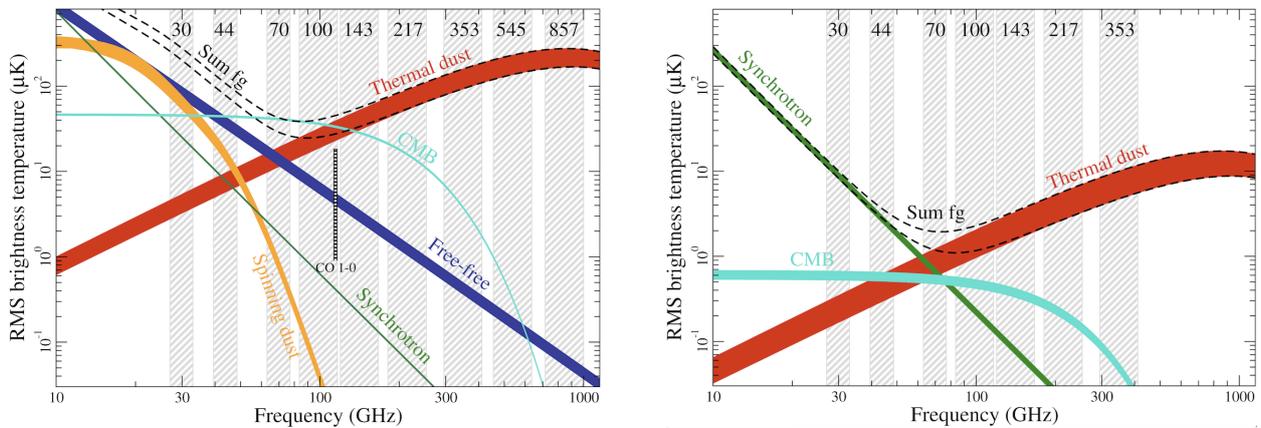


Figura 1.3: Potencia de las fluctuaciones en temperatura (izda) y polarización (dcha) de las componentes del cielo de microondas (temperatura de antena en μK). Fuente: www.esa.int/.

En cuanto a las fuentes extragalácticas, éstas son importantes a pequeña escala y están distribuidas por todo el cielo, siendo independientes de la latitud. Así, los datos recogidos por los expertos son una mezcla del FCM original, ruido instrumental y fuentes contaminantes, por lo que su eliminación se hace indispensable para el estudio del FCM.

1.3.1. Contaminantes galácticos

A escalas angulares grandes las emisiones contaminantes que surgen del medio interestelar de la galaxia son dominantes, existiendo, además, emisiones de fuentes compactas. Algunas de estas fuentes compactas son nubes moleculares, remanentes de supernova, regiones compactas

de HII o fuentes de radio galácticas. Las emisiones difusas contienen información importante sobre los procesos físicos que tienen lugar en la galaxia, como los procesos de formación estelar y el papel que juega en el enriquecimiento del medio interestelar. A continuación se describen las principales emisiones galácticas difusas.

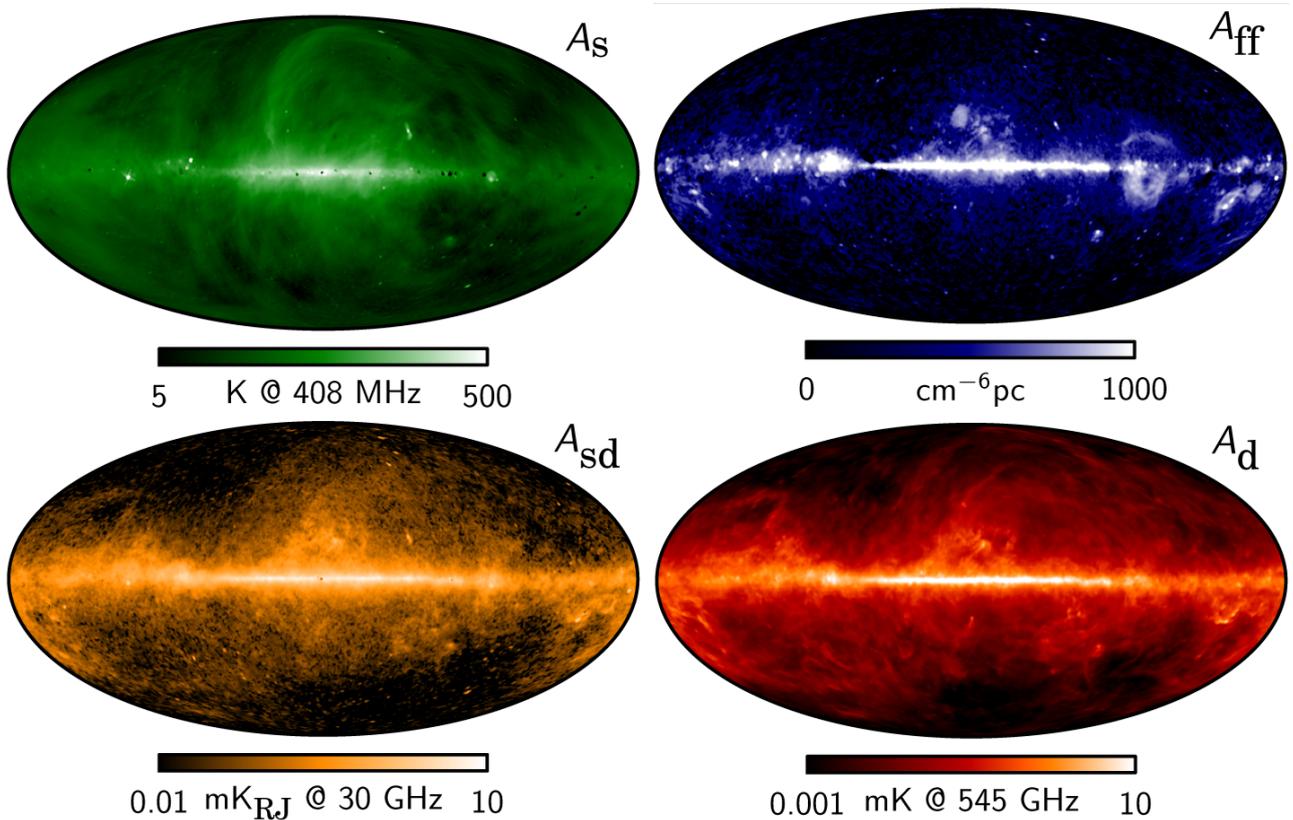


Figura 1.4: Mapas de temperatura de los contaminantes galácticos del cielo de microondas. De izquierda a derecha y de arriba a abajo: sincrotrón a 408 MHz, free-free, spinning dust a 30 GHz y polvo (thermal dust) a 545 GHz. El sincrotrón y el polvo tienen unidades de temperatura termodinámica, en K y mK respectivamente. El free-free está expresado en sus unidades típicas, densidad número de electrones al cuadrado (cm^{-6}) por distancia recorrida (en pc), y el spinning dust en mK (temperatura de antena). Fuente: www.esa.int/.

Emisión sincrotrón La emisión sincrotrón es la fuente de contaminación dominante en el rango bajo de las microondas ($\nu \lesssim 40$ GHz). Es un tipo de radiación no térmica de origen galáctico emitida por electrones acelerados en campos magnéticos. La emisión depende de la fuerza del campo magnético, del número de electrones y de su espectro de energía. Una característica de la emisión sincrotrón es que los fotones están polarizados perpendiculares a las líneas de campo. Esta polarización puede ser utilizada para localizar las fuentes de emisión sincrotrón y eliminarlas.

Emisión libre-libre Esta emisión, también denominada radiación de frenado o bremsstrahlung, proviene de dispersiones electrón-ión en el plasma interestelar. Esta radiación es emitida cuando una partícula cargada es decelerada tras ser desviada por otra partícula cargada. La emisión libre-libre tiene lugar en regiones galácticas ricas en hidrógeno ionizado, siendo posible su identificación gracias a las líneas de emisión $\text{H}\alpha$. A diferencia del sincrotrón, la emisión libre-libre se concentra más que éste en el plano galáctico. En estas regiones los electrones son desviados al pasar cerca de los núcleos hidrógeno y radían al ser decelerados.

La emisión de este contaminante galáctico es más intensa en el rango de frecuencias de 40 a 70 GHz. Los fotones de la emisión libre-libre no presentan una polarización apreciable.

Spinning dust En el rango de frecuencias entre los 20 y los 60 GHz aparece otro contaminante además de la emisión libre-libre, el spinning dust. En un principio se confundió a esta radiación con la emisión libre-libre debido a que aparece en el mismo rango de frecuencias que ésta. Sin embargo, la idea fue desechada pronto a consecuencia de la ausencia de líneas $H\alpha$. La distribución espacial de esta emisión es un tanto desconocida aún, pero se cree que está menos presente en la galaxia que el sincrotrón. Esta radiación es emitida por granos de polvo y moléculas nanométricas presentes en el medio interestelar, que pueden rotar a frecuencias de GHz.

Emisión térmica del polvo Por encima de los 70 GHz, la contaminación del FCM es mayoritariamente debida a la emisión térmica por granos de polvo del medio interestelar. La emisión térmica del polvo presenta un espectro de cuerpo negro modificado. Los granos de polvo más grandes pueden ser alineados por campos magnéticos galácticos originando polarización. La emisión térmica del polvo dificulta la observación del FCM a frecuencias $\gtrsim 150$ GHz.

Los mapas de componentes contaminantes de la Figura 1.4 fueron obtenidos por Planck. Para más información se puede consultar [8] donde se explica el modelo seguido. Con objeto de ilustrar la polarización de los contaminantes galácticos de la que también se ha hablado, en la Figura 1.5 se muestran los mapas de Planck de la polarización del sincrotrón y del polvo.

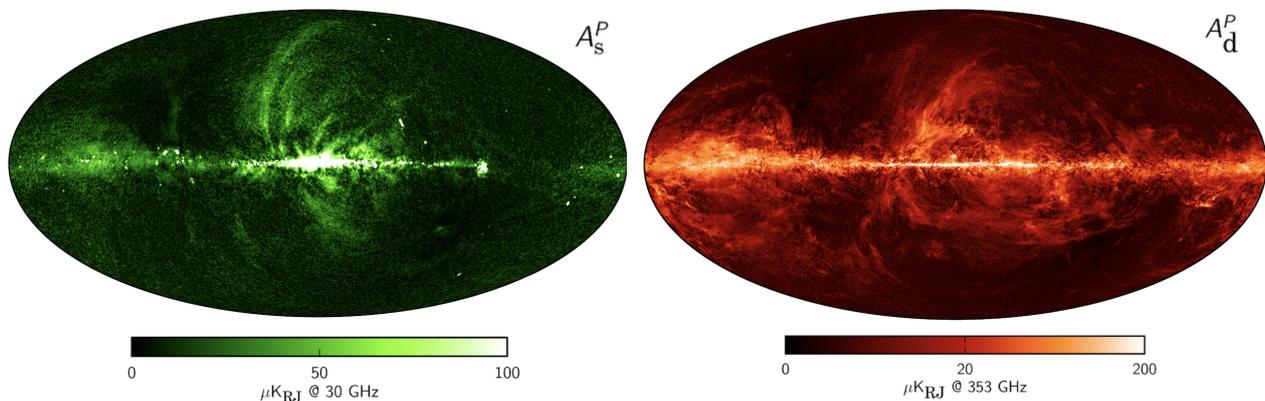


Figura 1.5: Mapas de polarización del sincrotrón a 30 GHz y del polvo a 353 GHz (temperatura de antena, en μK). Fuente: www.esa.int/.

1.3.2. Contaminantes extragalácticos

Los principales contaminantes del cielo de microondas son de origen galáctico, como el polvo o la emisión sincrotrón de las que ya se ha hablado. Sin embargo, también existen fuentes extragalácticas emisoras de radiación contaminante en el rango de las microondas. Estas emisiones extragalácticas provienen de un gran número de radio-galaxias, galaxias infrarrojas y cúmulos de galaxias.

Fuentes puntuales En el rango de frecuencias de observación comprendidas entre los 10 y los 100 GHz las fuentes puntuales contaminantes en el cielo de microondas son principalmente galaxias con núcleo activo (*AGN*), incluyendo radio-galaxias, cuásares y blázares. A frecuencias más altas, en el rango de 300-900 GHz, las fuentes puntuales dominantes son galaxias infrarrojas.

Cúmulos de galaxias Los cúmulos de galaxias contienen un gas de electrones caliente atrapado en el pozo de potencial del cúmulo. Este gas de electrones interactúa por efecto Compton inverso con la radiación del FCM produciendo una distorsión en su espectro de frecuencias. A este fenómeno se le denomina efecto Sunyaev-Zel'dovich y su observación ha servido para la detección de cúmulos de galaxias. Además de este efecto térmico, existe otro de origen cinético. El primero es más intenso y tiene una dependencia frecuencial conocida y bien determinada. El segundo es unas 10 veces más débil (en promedio) y su emisión tiene la misma dependencia frecuencial que el FCM, lo que dificulta su eliminación. En cualquier caso, es subdominante a escalas por encima de unas decenas de minutos de arco.

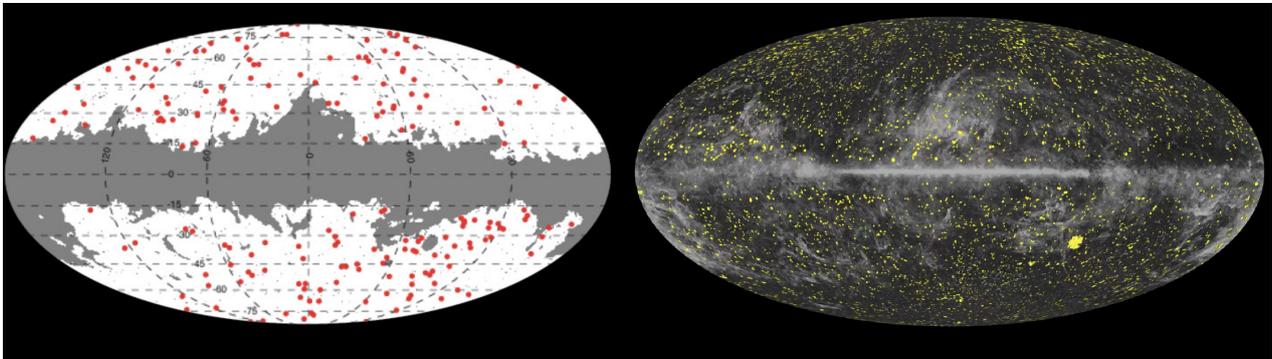


Figura 1.6: Mapa de las fuentes de contaminación extragaláctica del cielo de microondas. A la izquierda se muestran los cúmulos de galaxias detectados por Planck por efecto Sunyaev-Zel'dovich térmico y a la derecha las fuentes puntuales tanto de radio como de infrarrojo. Fuente: www.esa.int/.

Capítulo 2

Métodos de separación de componentes

Para acceder a la gran cantidad de información que contiene el cielo de microondas es necesario recurrir a la separación de componentes. La separación de componentes del cielo de microondas es un caso particular del problema inverso, en el que se infieren las componentes que forman un conjunto de datos, a partir de dicho conjunto. La separación de componentes tiene como objetivo final, en el marco del estudio del FCM, el eliminar las emisiones galácticas y extragalácticas para producir mapas en los que sólo aparezca el FCM. Además, algunos métodos de separación de componentes pueden proporcionar mapas de las emisiones difusas provenientes de la galaxia.

Según esto, en la separación de componentes se diferencian dos clases de métodos, aquellos que buscan eliminar las emisiones contaminantes del mapa, obteniendo un FCM con un nivel de contaminación lo más bajo posible, y aquellos que tratan de separar y extraer las señales componente a componente. En este trabajo se han implementado dos métodos lineales de separación de componentes que representan estas dos formas alternativas de abordar el problema; el ILC (*Internal Linear Combination*) [9], que minimiza la varianza de los datos con el objetivo de eliminar las emisiones contaminantes del mapa del FCM, y FastICA (*Fast Independent Component Analysis*) [10], un algoritmo que maximiza la no gaussianidad de las componentes para obtener, una a una, las emisiones del cielo de microondas. Otros métodos de eliminación de los contaminantes de los mapas del cielo de microondas son SEVEM (*Spectral Estimation Via Expectation Maximization*), que utiliza plantillas de los contaminantes para cada canal de frecuencia o NILC, que es una extensión del ILC para trabajar en el espacio de wavelets. Otro ejemplo de método lineal es SMICA (*Spectral Matching Independent Component Analysis*), que es una adaptación de ICA para trabajar en el espacio armónico, y que maximiza la independencia. Por último, un ejemplo de método de separación de componentes basado en la modelización paramétrica de los contaminantes es Commander-Ruler. Se puede encontrar una breve descripción de algunas de estas metodologías en [11]. Estos 4 últimos métodos han sido usados por la colaboración Planck [8].

Matemáticamente, el problema de la separación de componentes del cielo de microondas puede formularse de la siguiente manera. La temperatura total T en una dirección \hat{r} del cielo, a una frecuencia determinada ν , está dada por la superposición de señales provenientes de N procesos físicos distintos, entre los que se encuentra la radiación del FCM. Además, teniendo en cuenta la contribución del ruido n asociado al instrumento de medida y que cada frecuencia de observación tiene un distinto poder de resolución espacial caracterizado por una

convolución denotada por $b_\nu(\hat{r})$, se tiene:

$$T_\nu(\hat{r}) = s_{FCM}(\hat{r}) \otimes b_\nu(\hat{r}) + \sum_{i=1}^{N-1} s_{\nu,i}(\hat{r}) \otimes b_\nu(\hat{r}) + n_\nu(\hat{r}). \quad (2.1)$$

En la ecuación (2.1) se manifiesta de forma expresa que la temperatura (termodinámica) del FCM es independiente de la frecuencia. Dicho de otro modo, el FCM es constante en todos los canales de observación. Para simplificar el modelo, se suelen realizar algunas suposiciones: por ejemplo, en algunas ocasiones se impone que la función del beam b_ν es la misma para todas las frecuencias. De este modo, podemos suponer que el beam es parte de la señal y la ecuación (2.1) puede ser reescrita como:

$$T_\nu(\hat{r}) = s_{FCM}(\hat{r}) + \sum_{i=1}^{N-1} s_{\nu,i}(\hat{r}) + n_\nu(\hat{r}). \quad (2.2)$$

Es común el suponer además que cada componente s_ν puede factorizarse y escribirse como el producto de una función espacial y otra frecuencial, obteniéndose:

$$T_\nu(\hat{r}) = s_{FCM}(\hat{r}) + \sum_{i=1}^{N-1} s_i(\hat{r}) f_i(\nu) + n_\nu(\hat{r}). \quad (2.3)$$

Por simplicidad, la ecuación (2.3) puede ser escrita en notación matricial de la siguiente forma:

$$\mathbf{T} = \mathbf{AS} + \mathbf{N}, \quad (2.4)$$

donde $\mathbf{T} = (T_1, T_2, \dots, T_M)$ son las observaciones del cielo de microondas, \mathbf{A} es la matriz de mezcla de las componentes, de dimensión $M \times N$ (con M el número de canales de observación y N el número de señales astrofísicas), $\mathbf{S} = (s_1, s_2, \dots, s_{N-1}, s_{FCM})$ las componentes y $\mathbf{N} = (n_1, n_2, \dots, n_M)$ el ruido instrumental. El problema que se plantea a partir de la ecuación (2.4) es encontrar una transformación (lineal en la mayoría de los casos) de los datos \mathbf{T} que proporcione la mejor estimación posible de las componentes originales $\hat{\mathbf{S}}$. Esta transformación lineal puede ser alcanzada aplicando una matriz de transformación \mathbf{W} a nuestros datos como:

$$\hat{\mathbf{S}} = \mathbf{WT} = \mathbf{W}(\mathbf{AS} + \mathbf{N}). \quad (2.5)$$

El término \mathbf{WN} debido al ruido puede afectar de manera negativa a la reconstrucción de las señales obtenidas $\hat{\mathbf{S}}$. Como se ha mencionado anteriormente, dependiendo de la dimensión de \mathbf{W} , se puede obtener sólo el FCM (dimensión $1 \times M$) o N componentes (dimensión $N \times M$).

2.1. FastICA

El análisis de componentes independientes (ICA) [12] es una técnica estadística que permite expresar un conjunto de datos como combinaciones lineales de componentes estadísticamente independientes. Además de ser independientes, las componentes obtenidas por ICA no están correlacionadas, pues la independencia implica no correlación (el recíproco no siempre se cumple).

El teorema central del límite establece que la distribución de la suma de variables aleatorias independientes tiende a una distribución gaussiana bajo ciertas condiciones. Dicho de otra manera, la suma de dos variables aleatorias independientes tiene una distribución más parecida a la gaussiana que cualquiera de las dos variables aleatorias originales. Esta idea es

el principio en el que se sustenta ICA: con el fin de hallar las componentes independientes, ICA busca combinaciones lineales de los datos cuya no gaussianidad sea máxima.

FastICA es un algoritmo iterativo de ICA para la separación de componentes en la que, como mucho, se..que una de las señales sea gaussiana. A diferencia de otros métodos de separación de componentes, FastICA no necesita información a priori sobre las propiedades de las señales.

FastICA requiere que los datos estén no correlacionados, por lo que generalmente es necesario realizar un paso de blanqueo de los mismos. También es condición necesaria que el número de canales de observación sea igual al número de componentes a extraer. Para ello se suele aplicar previamente un PCA (*Principal Component Analysis*), un método que, a través de una transformación lineal, convierte un conjunto de datos (cuyas componentes pueden estar correlacionadas) en otro en el que las variables (componentes principales) están no correlacionadas. PCA no sólo blanquea los datos sino que además reduce su dimensión al número de componentes principales. En sí mismo, PCA podría considerarse un método de separación de componentes al igual que ICA. La diferencia entre ICA y PCA es que el primero consigue componentes independientes (y por lo tanto no correlacionadas) y el segundo obtiene componentes no correlacionadas pero que pueden no ser independientes. En particular, la no correlación implica independencia si las variables son gaussianas, pero debido a que en el cielo de microondas las señales presentes son no gaussianas (salvo el FCM), la aplicación de PCA no proporciona componentes independientes.

Para hallar las componentes principales, se hallan los vectores propios de los datos y sus respectivos valores propios. Si se quiere reducir de M a N canales efectivos, se toman los N vectores propios correspondientes a los N valores propios más grandes. Se dispone así de una matriz \mathbf{V} de dimensión $M \times N$ que al ser aplicada su inversa al vector de datos \mathbf{T} los reduce a dimensión $N \times 1$. Las columnas de la matriz \mathbf{V} son los N vectores propios elegidos. Tras haber aplicado esta matriz a los datos se obtienen las componentes principales sin blanquear. Como paso último y para blanquear los datos reducidos se aplica la inversa de la raíz cuadrada de la matriz diagonal $\mathbf{\Lambda}$, cuyos elementos son los valores propios de los vectores de la matriz \mathbf{V} . Por lo tanto, tras aplicar PCA a los datos se obtiene:

$$\hat{\mathbf{T}}_{PCA} = \mathbf{\Lambda}^{-1/2} \mathbf{V}^T \mathbf{T}. \quad (2.6)$$

En el caso de que se tengan tantos canales como componentes no es necesario aplicar PCA, basta con realizar un proceso más sencillo en el que solamente se blanqueen los datos. De entre las numerosas alternativas para blanquear los datos, en este trabajo se ha optado por el siguiente proceso de blanqueado de datos propuesto en [10]. Denominando \mathbf{C} y $\mathbf{\Sigma}$ a las matrices de covarianzas de nuestros datos y del ruido respectivamente, el proceso de blanqueamiento es:

$$\hat{\mathbf{\Sigma}} = (\mathbf{C} - \mathbf{\Sigma})^{-1/2} \mathbf{\Sigma} (\mathbf{C} - \mathbf{\Sigma})^{-1/2}, \quad (2.7)$$

$$\hat{\mathbf{T}} = (\mathbf{C} - \mathbf{\Sigma})^{-1/2} \mathbf{T}, \quad (2.8)$$

siendo $\hat{\mathbf{\Sigma}}$ una matriz de covarianzas modificada del ruido. De aquí en adelante, por simplicidad se denomina $\mathbf{Z} = (\mathbf{C} - \mathbf{\Sigma})^{-1/2}$. FastICA estima la matriz ortogonal \mathbf{W} fila a fila que al ser aplicada a los datos blanqueados $\hat{\mathbf{T}}$ proporciona una estimación $\hat{\mathbf{S}}$, que se relaciona con las componentes originales como:

$$\hat{\mathbf{S}} = \mathbf{W} \hat{\mathbf{T}} = \mathbf{WZ}(\mathbf{AS} + \mathbf{N}). \quad (2.9)$$

Si el método es insesgado, se ha de obtener que $\mathbf{WZ} = \mathbf{A}^{-1}$. Como se ha mencionado anteriormente, FastICA separa las componentes no gaussianas y hasta una componente gaussiana

de la mezcla. Esto significa que cuando se tiene más de una componente con una distribución gaussiana FastICA no es óptimo para separarlas. El ruido instrumental suele ser gaussiano por lo que en presencia de ruido la estimación de las componentes gaussianas va a empeorar considerablemente. Ahora el problema reside en encontrar tanto las señales originales \mathbf{S} como la matriz de mezcla \mathbf{A} a partir de los datos observados \mathbf{T} . Otro punto débil del método, es que las componentes independientes obtenidas por FastICA están normalizadas, de modo que el método recupera las señales salvo por una constante de normalización. La escala correcta, en la práctica, se ha de obtener a posteriori, en base a la física de las señales.

Una vez que los datos han sido blanqueados y se ha reducido su dimensión (en el caso de que fuera necesario), FastICA estima la matriz \mathbf{W} columna a columna, es decir, recupera las componentes de la mezcla una a una. A continuación se presenta el algoritmo de FastICA:

1. Se toma un vector aleatorio w_0 de norma unidad.
2. Se actualiza el vector como:

$$w = E \left\{ \hat{T} g(u) \right\} - (\mathbf{I} + \hat{\Sigma}) E \{ g'(u) \}, \quad (2.10)$$

donde E es el valor de espectación sobre todas las observaciones (en nuestro caso sobre una muestra de N_{pix} píxeles), $g(u)$ es una función que mide la no gaussianidad y $g'(u)$ es su derivada. La función $g(u)$ utilizada en este trabajo ha sido u^3 con $u = w_0^T \hat{T}$. Otras funciones de no gaussianidad alternativas propuestas por Hyvärinen son $\tanh(u)$ ó e^{-u^2} .

3. A continuación se realiza un paso de ortogonalización para asegurarse de que las componentes estimadas en cada proceso son diferentes. Para ello se busca en el subespacio ortogonal formado por las filas de la matriz \mathbf{W} halladas anteriormente. En nuestro algoritmo se ha implementado un paso de ortogonalización de Gram-Schmidt de la siguiente forma:

$$w = w - \sum_{i=1}^{j-1} \frac{\langle w, W_i \rangle}{\langle W_i, W_i \rangle} W_i, \quad (2.11)$$

siendo W_i es la i -ésima fila de la matriz \mathbf{W} , j el número de filas de la misma y donde $\langle w, W_i \rangle$ y $\langle W_i, W_i \rangle$ son productos escalares. El segundo término de la ecuación (2.11) representa el sumatorio de las proyecciones del vector w sobre las filas halladas de la matriz \mathbf{W} .

4. Se divide el vector resultante por su norma de modo que sea de módulo unidad:

$$w = \frac{w}{\|w\|}. \quad (2.12)$$

5. Finalmente se comprueba si ha convergido. Si el valor $\| \langle w^T, w_0 \rangle \|$ se aproxima a la unidad, el vector w resulta ser una fila de la matriz \mathbf{W} y se inicia otro proceso. Si no, se toma el vector w del paso 4 como vector inicial w_0 y se vuelve al paso 2.

Para estimar N componentes se repite el algoritmo anterior N veces.

2.2. Modificación del algoritmo de FastICA

Como se ha mencionado anteriormente, FastICA no es capaz de separar 2 componentes gaussianas presentes en la mezcla. Motivado por esta limitación y explorando si la inclusión

de información extra en el algoritmo podría mejorarlo, se ha probado a implementar una modificación en el algoritmo inicial de FastICA. La información extra que se aporta es la misma de la que se hace uso en el método de ILC (y que se explica en la siguiente sección), esto es, que para una de las componentes (típicamente, el FCM), se conoce su dependencia frecuencial.

En la práctica, dicha modificación implica conocer un vector columna de la matriz \mathbf{A} . Desarrollando la ecuación 2.9 (en la que, por simplicidad, se supone que no hay ruido), se obtiene la estimación de la componente i -ésima como:

$$\hat{s}_i = \sum_{m=1}^N \left[\sum_{l=1}^N W_{il} \sum_{k=1}^N Z_{lk} A_{km} \right] s_m, \text{ con } A_{k1} = 1, \forall k. \quad (2.13)$$

Para que \hat{s}_i sea un estimador insesgado de s_i , de la ecuación anterior se infiere:

$$\sum_{l=1}^N W_{il} \sum_{k=1}^N Z_{lk} = 1. \quad (2.14)$$

Por tanto, en la nueva implementación, para estimar la componente de forma más precisa, una vez que FastICA haya estimado la primera fila de la matriz \mathbf{W} se busca modificar los elementos de dicha fila para que se aproxime todavía más a la solución. Para conseguirlo se ha tratado de optimizar la siguiente función sujeta a dos restricciones:

$$f(W'_{i1}, \dots, W'_{iN}, \alpha, \beta) = \sum_{l=1}^N (W_{il} - W'_{il})^2 + \alpha \left[\sum_{l=1}^N W_{il}^2 - 1 \right] + \beta \left[\sum_{l=1}^N W'_{il} \sum_{k=1}^N Z_{lk} - 1 \right], \quad (2.15)$$

donde los W'_{il} son los nuevos elementos de la primera fila de la matriz \mathbf{W} . Derivando la función anterior con respecto a cada una de las incógnitas e igualando a cero se obtiene un sistema de $N + 2$ ecuaciones no lineales con $N + 2$ incógnitas. Para su resolución se ha implementado el método de Newton-Raphson.

2.3. ILC

El método de combinación lineal interna (ILC en inglés) es el método más sencillo para inferir una señal de la cual se conoce su dependencia frecuencial de un conjunto de observaciones. Este método permite extraer la señal sin hacer uso de información adicional externa (como, por ejemplo, parametrizaciones de las emisiones de los contaminantes). Además de conocer la dependencia frecuencial de la señal, se asuma que ésta y el resto de componentes no estén correlacionadas a la hora de obtener un estimador insesgado. ILC busca una combinación lineal de los datos que preserve la componente que se desea recuperar minimizando su varianza.

El incompleto conocimiento de algunas componentes astrofísicas así como de las propiedades del ruido instrumental puede llegar a suponer un problema a la hora de estimar las componentes de una mezcla. A este respecto, ILC presenta una ventaja frente a otros métodos de separación de componentes y es que no requiere información a priori y, lo que es más importante, sólo se centra en una componente. Por este motivo ILC ha sido empleado en numerosas ocasiones en experimentos del FCM. En concreto, el método de ILC ha sido utilizado para hallar el FCM a partir de los datos recogidos por el satélite WMAP de la NASA [9] y una extensión del mismo (NILC), para trabajar en el espacio de wavelets [8], ha

sido aplicado a los datos de Planck de la ESA, para trabajar en el espacio de wavelets.

En el caso del FCM se sabe que presenta un espectro de cuerpo negro, siendo su temperatura independiente de la frecuencia. Además, la señal del FCM y los contaminantes no están estadísticamente correlacionados de modo que el modelo de ILC puede aplicarse directamente a los mapas para obtener la componente del FCM de la siguiente forma:

$$\hat{s}_{fcm} = \sum_{i=1}^M \omega_i T_i, \quad (2.16)$$

donde ω_i son los pesos de los mapas que han de cumplir la condición de que su suma sea igual a la unidad:

$$\sum_{i=1}^M \omega_i = 1. \quad (2.17)$$

Para obtener los pesos de cada mapa, ILC busca que la varianza de (2.16) sea mínima bajo la restricción (2.17). Aplicando el método de los multiplicadores de Lagrange se minimiza la siguiente función:

$$f = \frac{1}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} s_{FCM}^2(\hat{r}) + \lambda \left[\sum_{i=1}^M \omega_i - 1 \right], \quad (2.18)$$

donde se ha tenido en cuenta que la media de s_{FCM} es cero. Sustituyendo (2.16) en la ecuación anterior se obtiene:

$$f(\omega_1, \dots, \omega_M, \lambda) = \frac{1}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} \left[\sum_{i=1}^M \omega_i \hat{T}_i \right]^2 + \lambda \left[\sum_{i=1}^M \omega_i - 1 \right]. \quad (2.19)$$

Derivando la función f con respecto a cada una de las $M + 1$ variables e igualando a cero se obtienen $M + 1$ ecuaciones. Finalmente, para hallar los pesos hay que resolver el sistema de ecuaciones lineales resultante, que expresado en forma matricial es:

$$\begin{pmatrix} \frac{2}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} T_1^2 & \cdots & \frac{2}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} T_1 T_M & 1 \\ \frac{2}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} T_2 T_1 & \cdots & \frac{2}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} T_2 T_M & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \frac{2}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} T_M T_1 & \cdots & \frac{2}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} T_M^2 & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_M \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}. \quad (2.20)$$

Así, es inmediato que los pesos son:

$$\begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_M \\ \lambda \end{pmatrix} = \begin{pmatrix} \frac{2}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} T_1^2 & \cdots & \frac{2}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} T_1 T_M & 1 \\ \frac{2}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} T_2 T_1 & \cdots & \frac{2}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} T_2 T_M & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \frac{2}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} T_M T_1 & \cdots & \frac{2}{N_{pix}} \sum_{\hat{r}}^{N_{pix}} T_M^2 & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}. \quad (2.21)$$

Una vez se han hallado los pesos correspondientes a cada mapa se utilizan para formar la combinación lineal que minimiza las componentes contaminantes y el ruido, obteniéndose el mapa estimación del FCM.

Este método no puede usarse para estimar los contaminantes galácticos por el desconocimiento que se tiene sobre la dependencia frecuencial. Consecuentemente el ILC no es un método de separación de componentes sino un método de reducción de las fuentes contaminantes. En el caso particular de la separación de componentes del cielo de microondas, lo que busca ILC es minimizar las componentes astrofísicas y el ruido gaussiano inherente a los instrumentos de medida que contaminan la señal del FCM, dejándola lo más limpia posible. Otra ventaja de ILC es que puede aplicarse a cualquier tipo de señal independientemente de cuál sea su distribución. Algunos métodos no pueden ser empleados en situaciones en las que las componentes presenten cierto tipo de distribuciones, como es el caso de FastICA, descrito con anterioridad.

2.4. Casos estudiados en este trabajo

En este trabajo se ha abordado la separación de componentes en 3 casos en los que se ha ido aumentando progresivamente la complejidad del problema. El primer caso tratado, y el más sencillo, ha sido el de simulaciones unidimensionales. Se ha tratado de recuperar señales en 1-D por los métodos lineales expuestos anteriormente, FastICA e ILC, en función de algunos parámetros como son las propiedades estadísticas de las componentes, el dominio de las mismas en los canales de observación, o la existencia o no de ruido instrumental.

En la segunda parte se ha aplicado la separación de componentes a un caso más realista donde los datos han sido simulaciones del cielo de microondas en 2-D para las características instaladas del experimento CORE [13]. Este caso es más complejo con respecto al 1-D por varios motivos: en primer lugar los contaminantes han sido simulados de una forma realista, razón por la cual algunas de las suposiciones que se hacían en el caso 1-D no pueden tenerse en cuenta. Además en el caso unidimensional $M = N$ y en este caso se dispone de más canales de observación que componentes, siendo necesaria una reducción de los datos en el caso particular de FastICA.

Para finalizar, se ha estudiado un caso de datos reales, el del cielo de microondas obtenido por Planck [5]. En este caso los datos utilizados han sido los mapas de los 9 canales de frecuencia proporcionados por Planck del cielo de microondas. Se ha recuperado la señal del FCM por los dos métodos lineales de separación de componentes implementados. En el caso de FastICA, se ha estudiado la bondad de los resultados en función del número de componentes.

Capítulo 3

Estudio de simulaciones 1-D

En este capítulo se presentan los resultados obtenidos tras estudiar de forma detallada la recuperación de señales en una dimensión mediante los dos métodos lineales de separación de componentes descritos anteriormente, FastICA e ILC. En las simulaciones se ha utilizado el mismo número de canales de frecuencia que de componentes ($N = M = 3$), por lo que en el caso de FastICA no ha sido necesario aplicar un PCA para reducir la dimensión de los datos. No obstante, sí ha sido necesario realizar el blanqueamiento de los datos explicado en el capítulo anterior.

Para comparar los dos métodos, la separación de componentes por ILC y FastICA se ha llevado a cabo en diferentes situaciones. La primera parte de este estudio se ha basado en las propiedades estadísticas de las componentes. En particular, se han considerado las mezclas de 3 señales no gaussianas, 1 gaussiana y 2 no gaussianas y, por último, de 2 gaussianas y 1 no gaussiana. Las componentes unidimensionales simuladas han sido generadas como:

$$s(t) = s_g(t) + \varepsilon s_g^2(t), \quad (3.1)$$

donde $s(t)$ es la señal generada a partir de una señal $s_g(t)$ con distribución gaussiana, de media cero y dispersión unidad. Si $\varepsilon = 0$ la señal generada es gaussiana, $s(t) = s_g(t)$, mientras que si $\varepsilon \neq 0$, la componente tiene una distribución no gaussiana $s(t) = s_g(t) + \varepsilon s_g^2(t) \equiv s_{ng}(t)$, dependiente del valor de ε . En este trabajo hemos considerado señales no gaussianas con valores: $\varepsilon_1 = 3$, $\varepsilon_2 = 2$ y $\varepsilon_3 = 5$. Cada señal generada es una muestra de 10.000 puntos y píxeles.

Con objeto de ilustrar las observaciones generadas en este trabajo según la ecuación (3.1), en la Figura 3.1 se muestra una componente no gaussiana y otra gaussiana. Así mismo, se muestran los histogramas de ambas señales. El histograma de la componente gaussiana muestra la distribución normal característica de ese tipo de señales, mientras que la no gaussiana tiene una clara asimetría positiva.

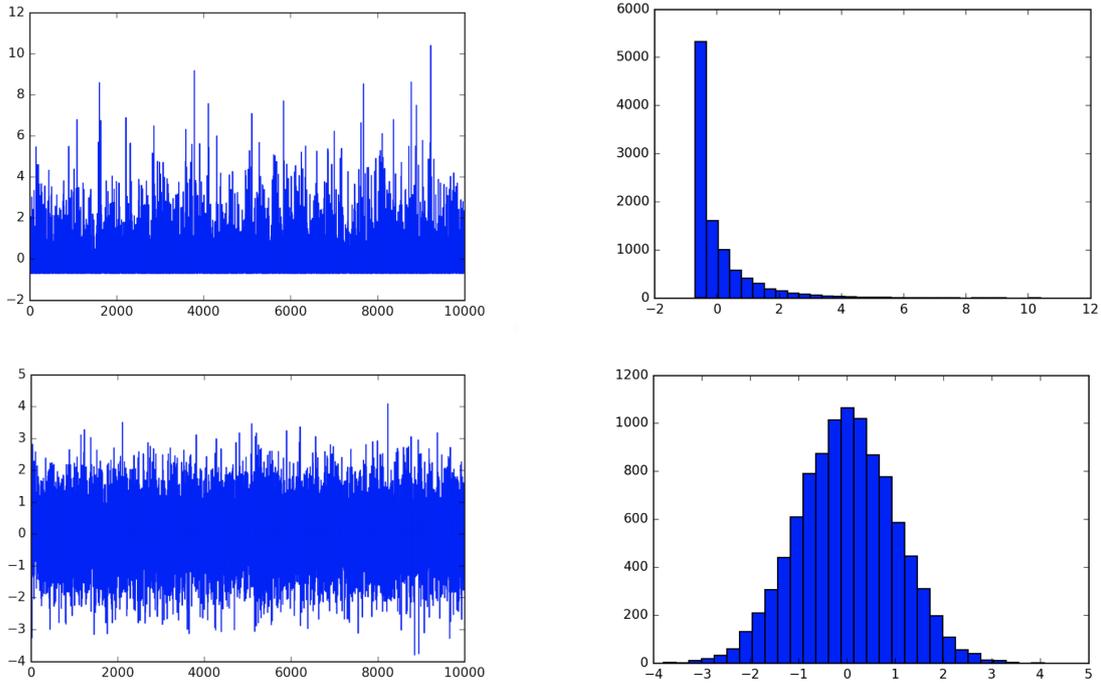


Figura 3.1: De izquierda a derecha: simulación de una componente no gaussiana para $\varepsilon = 3$ y su histograma; señal gaussiana ($\varepsilon = 0$) y su histograma.

Así mismo, los dos métodos se han aplicado tanto en presencia como en ausencia de ruido instrumental. En los casos en los que se ha introducido ruido en las simulaciones éste ha sido generado como una señal gaussiana de media cero y dispersión $\sigma(n_\nu) = \eta^{1/2}\sigma(T'_\nu)$, donde η es una fracción y T'_ν es la observación sin ruido. En este estudio se ha tomado el valor $\eta = 0.05$. Además de las propiedades estadísticas de las componentes y del ruido instrumental, se ha estudiado cómo afecta a la separación de componentes la mezcla de las señales. Para ello se han utilizado dos matrices de mezcla, una en la que no hay dominio de las señales en los canales de observación, y otra en la que en cada uno de los canales hay una componente que domina sobre el resto. Las matrices utilizadas han sido:

$$\mathbf{A}_{\text{ND}} = \begin{pmatrix} 1 & 2 & 1 \\ 3 & 2 & 1 \\ 2 & 3 & 1 \end{pmatrix} \quad (3.2)$$

y

$$\mathbf{A}_{\text{D}} = \begin{pmatrix} 2 & 0.6 & 1 \\ 0.6 & 2 & 1 \\ 0.2 & 0.2 & 1 \end{pmatrix} \quad (3.3)$$

En la matriz de no dominio en (3.2), se observa que la tercera componente no domina en ningún canal de observación. Las señales 1 y 2 dominan ligeramente en los canales 2 y 3 respectivamente, pero se encuentran en una proporción similar. En cambio, en la matriz de dominio (3.3) se tiene que la primera componente domina en el canal de frecuencia 1, la segunda en el canal 2 y la tercera en el canal 3. En ambas matrices, la tercera columna representa una dependencia similar a la del FCM en el caso del cielo de microondas.

Para ilustrar el estudio de la separación de componentes unidimensionales que tiene lugar a continuación, la siguiente figura muestra un ejemplo visual de un caso sencillo. En la Figura 3.2 aparece a modo de esquema cómo son simuladas las observaciones. Se observan

3 componentes, dos gaussianas y una tercera no gaussiana, mezcladas a través de la matriz \mathbf{A}_D y se ha añadido ruido. A la izquierda aparecen las observaciones, en el centro las señales originales y a la derecha el ruido añadido. Las soluciones aparecen en la Figura 3.3.

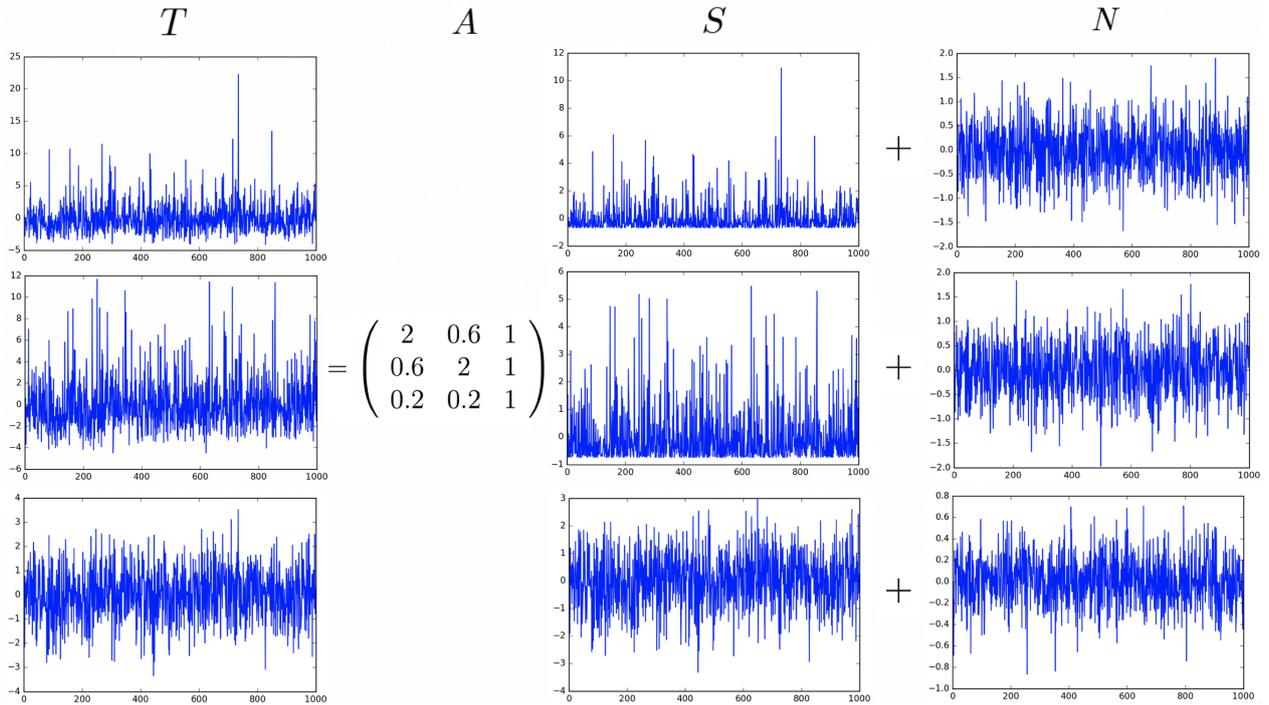


Figura 3.2: Esquema visual de la observación. A la izquierda aparecen 3 observaciones, producto de la aplicación de la matriz de mezcla a las componentes originales más el ruido.

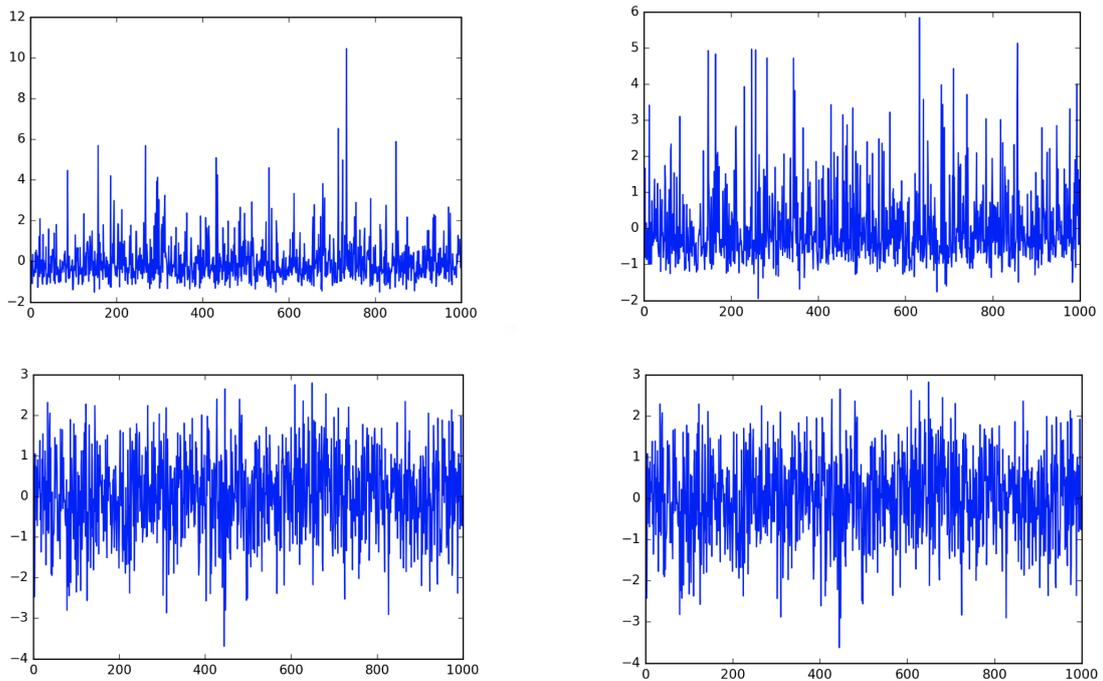


Figura 3.3: De izquierda a derecha: componentes 1 y 2 recuperadas por FastICA. Abajo la tercera componente por FastICA e ILC, respectivamente.

Para cuantificar la bondad de las componentes estimadas y comparar los resultados obtenidos mediante ambos métodos, se han calculado los coeficientes de correlación ρ entre las señales estimadas y las originales. También se ha calculado los valores de la skewness (momento de orden 3) de las componentes estimadas y originales, así como la diferencia entre ellas. En lo que sigue, los resultados de FastICA se ofrecen en dos casos: la implementación original y la modificación propuesta en este trabajo. Para cada caso se han generado un conjunto de 100 simulaciones, que nos ayudan a estimar los valores medios y la desviación estándar de estos estadísticos.

3.1. Separación de 3 componentes no gaussianas

En la separación de componentes no gaussianas, los resultados obtenidos en las simulaciones son óptimos en los casos sin ruido mientras que cuando se añade ruido a las observaciones la estimación de las componentes originales mediante los algoritmos de FastICA e ILC empeoran, notablemente en el caso de no dominio. En la Tabla 3.1 aparecen tabulados los coeficientes de correlación ρ :

R	D	$\langle \rho_1 \rangle$	$\langle \rho_2 \rangle$	$\langle \rho_3 \rangle$	$\langle \rho_{ILC} \rangle$
No	No	0.9996 ± 0.0004 0.9998 ± 0.0004	0.9996 ± 0.0006 0.9998 ± 0.0003	0.9996 ± 0.0006 0.9998 ± 0.0002	0.9999 ± 0.0001
	Sí	0.9997 ± 0.0003 0.9997 ± 0.0004	0.9997 ± 0.0004 0.9998 ± 0.0003	0.9997 ± 0.0004 0.9998 ± 0.0003	0.9999 ± 0.0001
Sí	No	0.8902 ± 0.0084 0.8512 ± 0.0816	0.7157 ± 0.0527 0.6488 ± 0.0877	0.4131 ± 0.0154 0.4249 ± 0.0186	0.5112 ± 0.0073
	Sí	0.9501 ± 0.0012 0.9471 ± 0.0098	0.9501 ± 0.0013 0.9473 ± 0.0130	0.9620 ± 0.0010 0.9581 ± 0.0180	0.9636 ± 0.0005

Tabla 3.1: Valor medio de los coeficientes de correlación ρ para el caso de 3 señales no gaussianas. R y D hacen referencia a si se trata de un caso con ruido o no, y con dominio o sin él en la mezcla, respectivamente.

En el caso sin ruido la separación de componentes consigue muy buenos resultados, siendo los coeficientes de correlación muy próximos a la unidad. En ausencia de ruido instrumental el uso de la matriz con dominio (y del algoritmo modificado en el caso de FastICA) conduce a resultados análogos al caso anterior. Cabe destacar que la componente recuperada por el ILC resulta una mejor estimación que la de FastICA, tal y como reflejan los valores de ρ .

Los resultados obtenidos con la introducción de ruido son bien distintos. Sin dominio, tanto FastICA como ILC no son capaces de dar una correcta estimación de las componentes. Los valores de ρ muestran la descorrelación entre componentes. De las tres componentes la peor estimación corresponde a la señal que es constante en todos los canales de frecuencia. El hecho de que esta señal no domine en ningún canal y que se añada ruido provoca que ni FastICA ni ILC recuperen la señal. Que la estimación de ILC sea ligeramente mejor que la de FastICA es inherente al tipo de método. Mientras que el primero limpia de señales y ruido en la medida de lo posible la componente a extraer, el segundo tan sólo realiza una separación de las componentes, no siendo posible desprenderse del ruido. El uso de la matriz con dominio mejora las soluciones de ILC y FastICA. Por el contrario, el método de FastICA modificado no aporta ninguna mejora significativa en estos casos con ruido.

En la Tabla 3.2 aparecen las diferencias entre la skewness de las componentes recuperadas y las iniciales, ΔS :

R	D	$\langle \Delta S_1 \rangle$	$\langle \Delta S_2 \rangle$	$\langle \Delta S_3 \rangle$	$\langle \Delta S_{ILC} \rangle$
No	No	-0.0015 ± 0.0023	-0.0021 ± 0.0049	-0.0024 ± 0.0036	-0.0008 ± 0.0013
		-0.0015 ± 0.0034	-0.0012 ± 0.0023	-0.0017 ± 0.0028	
No	Sí	-0.0021 ± 0.0031	-0.0021 ± 0.0037	-0.0008 ± 0.0016	-0.0009 ± 0.0013
		-0.0021 ± 0.0033	-0.0012 ± 0.0025	-0.0016 ± 0.0025	
Sí	No	-0.8217 ± 0.0719	-1.7626 ± 0.1953	-2.6122 ± 0.1453	-2.0421 ± 0.1248
		-1.0465 ± 0.3982	-2.0135 ± 0.3860	-2.5933 ± 0.1281	
Sí	Sí	-0.3976 ± 0.0262	-0.3979 ± 0.0327	-0.3090 ± 0.0267	-0.2986 ± 0.0250
		-0.4186 ± 0.0600	-0.4221 ± 0.1103	-0.3322 ± 0.1265	

Tabla 3.2: Promedio de las diferencias de la skewness de las componentes recuperadas y originales.

Al igual que ocurría con los coeficientes de correlación, los valores de ΔS empeoran (se acercan menos a cero) en los casos con ruido. Sin ruido, ΔS tiende a cero sin que existan grandes diferencias entre los casos sin dominio y con dominio. El método de FastICA modificado tampoco mejora los resultados disminuyendo el valor ΔS . Si a las observaciones se añade ruido ΔS aumenta con respecto al caso sin ruido, siendo menor cuando hay dominio de las componentes.

3.2. Separación de componentes con una señal gaussiana

En el caso en el que una de las señales es gaussiana los resultados de la separación de componentes son similares al caso en el que todas las señales son no gaussianas. Esto es de esperar ya que, como se ha explicado, de un lado, FastICA está diseñado para recuperar hasta como mucho una señal gaussiana y, por otro lado, ILC no depende de las propiedades estadísticas individuales de cada señal. Se ha tomado $\varepsilon_3 = 0$, es decir, la señal gaussiana es la que es constante en los distintos canales (también por analogía con el FCM). En la Tabla 3.3 aparece el promedio de los coeficientes de correlación. Si en los datos no se ha introducido ruido, tanto FastICA como ILC consiguen separar las componentes de la mezcla obteniendo una correlación entre componentes prácticamente igual a la unidad. La presencia de ruido instrumental en las observaciones dificulta la separación de componentes.

R	D	$\langle \rho_1 \rangle$	$\langle \rho_2 \rangle$	$\langle \rho_3 \rangle$	$\langle \rho_{ILC} \rangle$
No	No	0.9997 ± 0.0004	0.9996 ± 0.0004	0.9995 ± 0.0005	0.9999 ± 0.0001
		0.9997 ± 0.0004	0.9997 ± 0.0004	0.9998 ± 0.0004	
No	Sí	0.9997 ± 0.0003	0.9996 ± 0.0005	0.9995 ± 0.0005	0.9999 ± 0.0001
		0.9998 ± 0.0002	0.9998 ± 0.0002	0.9998 ± 0.0003	
Sí	No	0.8864 ± 0.0149	0.7121 ± 0.0462	0.4136 ± 0.0125	0.5111 ± 0.0062
		0.8405 ± 0.0770	0.6587 ± 0.0868	0.4249 ± 0.0213	
Sí	Sí	0.9500 ± 0.0013	0.9498 ± 0.0012	0.9620 ± 0.0013	0.9636 ± 0.0006
		0.9417 ± 0.0345	0.9396 ± 0.0377	0.9545 ± 0.0276	

Tabla 3.3: Valor medio de los coeficientes de correlación ρ para el caso de 1 gaussiana.

Comparando los valores de los coeficientes de la Tabla 3.1 y la Tabla 3.3, se puede afirmar que los resultados obtenidos en la separación de componentes cuando se cuenta con una señal de tipo gaussiana es prácticamente igual al caso en el que todas las componentes son no gaussianas.

R	D	$\langle \Delta S_1 \rangle$	$\langle \Delta S_2 \rangle$	$\langle \Delta S_3 \rangle$	$\langle \Delta S_{ILC} \rangle$
No	No	-0.0011 ± 0.0026 -0.0018 ± 0.0037	-0.0008 ± 0.0027 -0.0020 ± 0.0033	-0.0001 ± 0.0012 -0.00006 ± 0.00079	0.00003 ± 0.00062
	Sí	-0.0007 ± 0.0015 -0.0015 ± 0.0027	-0.0015 ± 0.0034 -0.0012 ± 0.0019	0.0001 ± 0.0013 0.00008 ± 0.00090	0.00002 ± 0.00062
Sí	No	-0.8584 ± 0.1108 -1.1041 ± 0.3946	-1.7909 ± 0.2037 -1.9458 ± 0.3542	0.0009 ± 0.0317 0.0816 ± 0.3226	0.3948 ± 0.0406
	Sí	-0.4060 ± 0.0324 -0.4667 ± 0.2762	-0.4016 ± 0.0274 -0.4579 ± 0.1934	0.0013 ± 0.0129 0.0141 ± 0.0739	0.0015 ± 0.0125

Tabla 3.4: Promedio de las diferencias de la skewness de las componentes recuperadas y originales.

Similares conclusiones se pueden derivar atendiendo a la skewness (Tabla 3.2 y 3.4).

3.3. Separación de componentes con dos gaussianas

Por último, el caso en el que se cuenta con dos componentes con distribuciones gaussianas, es el que peores resultados ha proporcionado desde el punto de vista de FastICA, como era de esperar. Las señales han sido generadas tomando $\varepsilon_2 = \varepsilon_3 = 0$.

Los resultados obtenidos por ILC han sido similares a los de los dos casos anteriores. Las señales recuperadas por ILC son buenas estimaciones de las originales excepto en los casos en los que se añade ruido.

Respecto a FastICA, los resultados difieren de los dos casos estudiados con anterioridad. Como se explicó en la sección de separación de componentes, FastICA sólo puede estimar componentes de mezclas en las que como mucho haya una señal gaussiana. Por ello, al disponer en este caso de 2 componentes gaussianas FastICA no es capaz de separarlas. La Tabla 3.5 muestra los coeficientes de correlación ρ :

R	D	$\langle \rho_1 \rangle$	$\langle \rho_2 \rangle$	$\langle \rho_3 \rangle$	$\langle \rho_{ILC} \rangle$
No	No	0.9997 ± 0.0003 0.9997 ± 0.0003	0.8963 ± 0.0931 0.9997 ± 0.0005	0.8959 ± 0.0938 0.9998 ± 0.0003	0.9999 ± 0.0001
	Sí	0.9996 ± 0.0007 0.9997 ± 0.0002	0.9042 ± 0.0854 0.9997 ± 0.0003	0.9042 ± 0.0855 0.9998 ± 0.0003	0.9999 ± 0.0001
Sí	No	0.8819 ± 0.0488 0.8337 ± 0.0850	0.7384 ± 0.1473 0.7081 ± 0.0740	0.4255 ± 0.0486 0.4147 ± 0.0215	0.5114 ± 0.0067
	Sí	0.9500 ± 0.0013 0.9439 ± 0.0202	0.8570 ± 0.0838 0.9441 ± 0.0256	0.8659 ± 0.0869 0.9535 ± 0.0269	0.9635 ± 0.0005

Tabla 3.5: Valor medio de los coeficientes de correlación ρ para el caso de 2 gaussianas.

Centrándonos en los casos sin ruido se observa que la primera componente (no gaussiana) es estimada por FastICA de manera satisfactoria. Sin embargo, FastICA es incapaz de proporcionar una buena estimación de las dos componentes gaussianas porque no puede separarlas, tanto en el caso sin dominio (donde resulta más evidente) como en el caso con dominio. Ahora, si en lugar del método original de FastICA explicado en la sección 1.4.1 se utiliza el algoritmo modificado, se observa que FastICA sí que es capaz de separar las componentes gaussianas, rompiendo así la degeneración de las mismas. En esta situación los coeficientes ρ muestran una correlación casi perfecta, para los casos usuales.

La tabla que aparece a continuación muestra los valores de ΔS :

R	D	$\langle \Delta S_1 \rangle$	$\langle \Delta S_2 \rangle$	$\langle \Delta S_3 \rangle$	$\langle \Delta S_{ILC} \rangle$
No	No	-0.0003 ± 0.0006	-0.00007 ± 0.01588	-0.0009 ± 0.0161	-0.0001 ± 0.0006
		-0.0014 ± 0.0018	-0.0002 ± 0.0013	-0.0001 ± 0.0010	
No	Sí	-0.0008 ± 0.0045	-0.0004 ± 0.0150	-0.0016 ± 0.0184	-0.000005 ± 0.000691
		-0.0011 ± 0.0015	-0.0002 ± 0.0011	-0.0001 ± 0.0008	
Sí	No	-0.8663 ± 0.2066	-0.0010 ± 0.0234	-0.00007 ± 0.03588	0.0049 ± 0.0342
		-1.1385 ± 0.4052	-0.0003 ± 0.0272	0.0016 ± 0.0349	
Sí	Sí	-0.3929 ± 0.0296	-0.0013 ± 0.0208	0.0011 ± 0.0173	0.0007 ± 0.0105
		-0.4425 ± 0.1477	-0.0021 ± 0.0258	0.0078 ± 0.0522	

Tabla 3.6: Promedio de las diferencias de la skewness de las componentes recuperadas y originales.

Si nos fijamos en la componente constante en los casos sin ruido, tanto sin dominio como con él, FastICA consigue mejorar los valores de ΔS . Cuando no hay dominio ΔS mejora en un factor 9 y en el caso con dominio en un factor 16.

3.4. Discusión

Como se esperaba, los resultados de la separación de componentes 1-D por FastICA ha dependido de las propiedades estadísticas de las componentes, siendo imposible separar las componentes gaussianas cuando hay al menos 2 de ellas en la mezcla. En cambio, la recuperación de la componente de tipo FCM por parte de ILC ha proporcionado resultados similares en los 3 casos estudiados en este capítulo, independientemente de si las componentes de la mezcla eran o no gaussianas. Aunque en menor medida, los resultados obtenidos también han dependido de la matriz de mezcla \mathbf{A} y del ruido de los distintos canales de observación, mejorando en aquellos casos en los que las componentes dominan en alguno de los canales y empeorando con el ruido.

En relación al caso con 2 gaussianas y a la imposibilidad de separarlas por FastICA, es necesario aclarar un par de cosas. La primera con respecto al ruido gaussiano generado para las simulaciones, del cual se podría pensar que debido a que es gaussiano puede jugar un papel similar al de otra componente gaussiana, lo cual no es cierto. El ejemplo más claro es el caso de la separación de componentes con una señal gaussiana, que en los casos con ruido proporcionaba los mismos resultados que el caso de 3 componentes no gaussianas. Queda claro, por tanto, que el ruido aunque también es gaussiano no juega el mismo rol que otra componente gaussiana, y la razón es la falta de coherencia del ruido entre canales de observación. En segundo lugar, hacer especial énfasis en que la modificación del algoritmo original

de FastICA permite separar componentes gaussianas de la mezcla al romper la degeneración de las mismas.

Para finalizar, establecer que los dos métodos sirven a propósitos distintos, de manera que si se quiere recuperar una única componente de la cual se conoce su dependencia con la frecuencia (como en el caso real del FCM) es lógico que se utilice el ILC, mientras que por su parte, FastICA puede ser de utilidad cuando lo que se busca es separar múltiples componentes de una mezcla.

Capítulo 4

Estudio de simulaciones 2-D: CORE

Los métodos de separación de componentes descritos en el capítulo 2 han sido aplicados a mapas simulados del cielo de microondas. Esta configuración refleja una situación similar a la propuesta de la misión CORE [13], hecha a la ESA en el contexto de su programa de misiones de tamaño intermedio. Nótese que, si bien cada uno de los canales de CORE tiene su propia resolución espacial (de los 17.5 arcmin a 60 GHz, a los 2 arcmin a 600 GHz), sin embargo, debido a que, como se mencionó en la sección 2, tanto FastICA como ILC necesitan trabajar con datos a la misma resolución, aquí se ha adoptado la de 17.5' para todos ellos. Además de la señal del FCM, se han incluido los contaminantes galácticos y extragalácticos más relevantes (sincrotrón, polvo y fuentes puntuales, respectivamente), mostrados a continuación en la Figura 4.1 (donde, a modo de ilustración, se representan a la frecuencia de 100 GHz). Así mismo, en la Figura 4.2, se muestra la emisión total de los 19 canales de la misión.

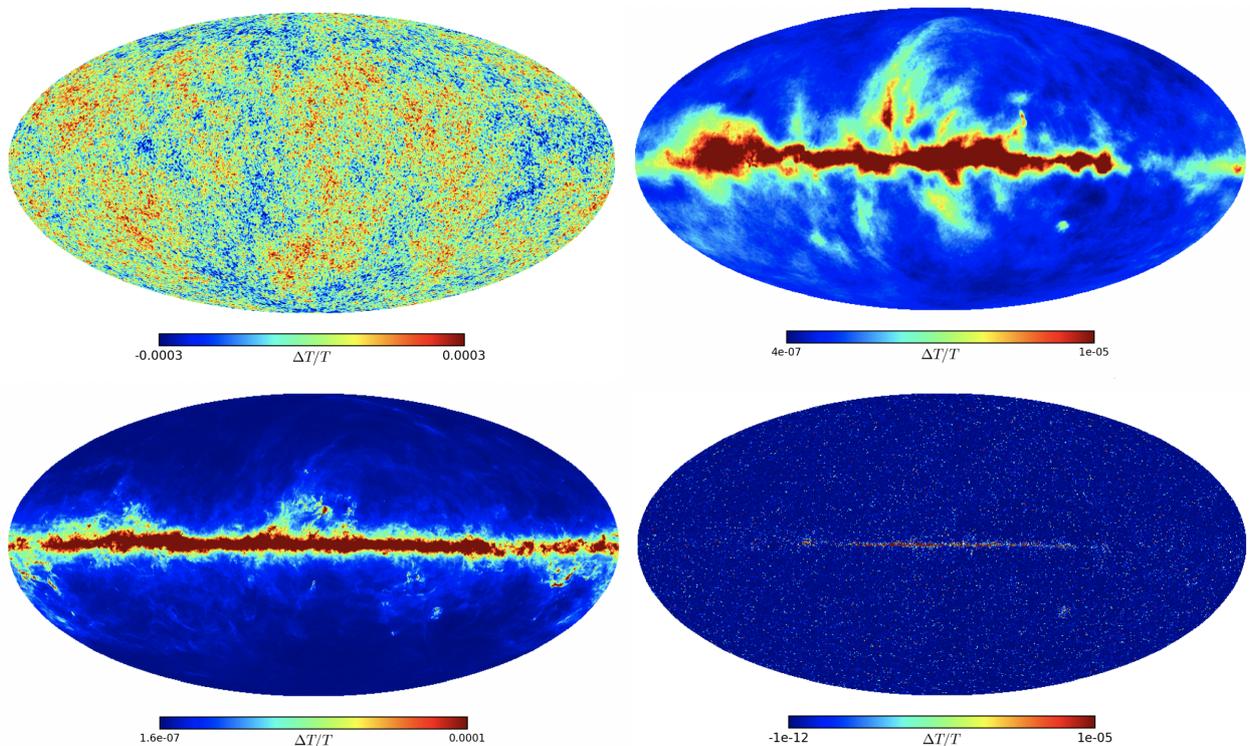


Figura 4.1: Componentes del cielo de microondas simulados a la frecuencia de observación $\nu = 100$ GHz. De izquierda a derecha y de arriba a abajo: FCM, sincrotrón, polvo y fuentes puntuales.

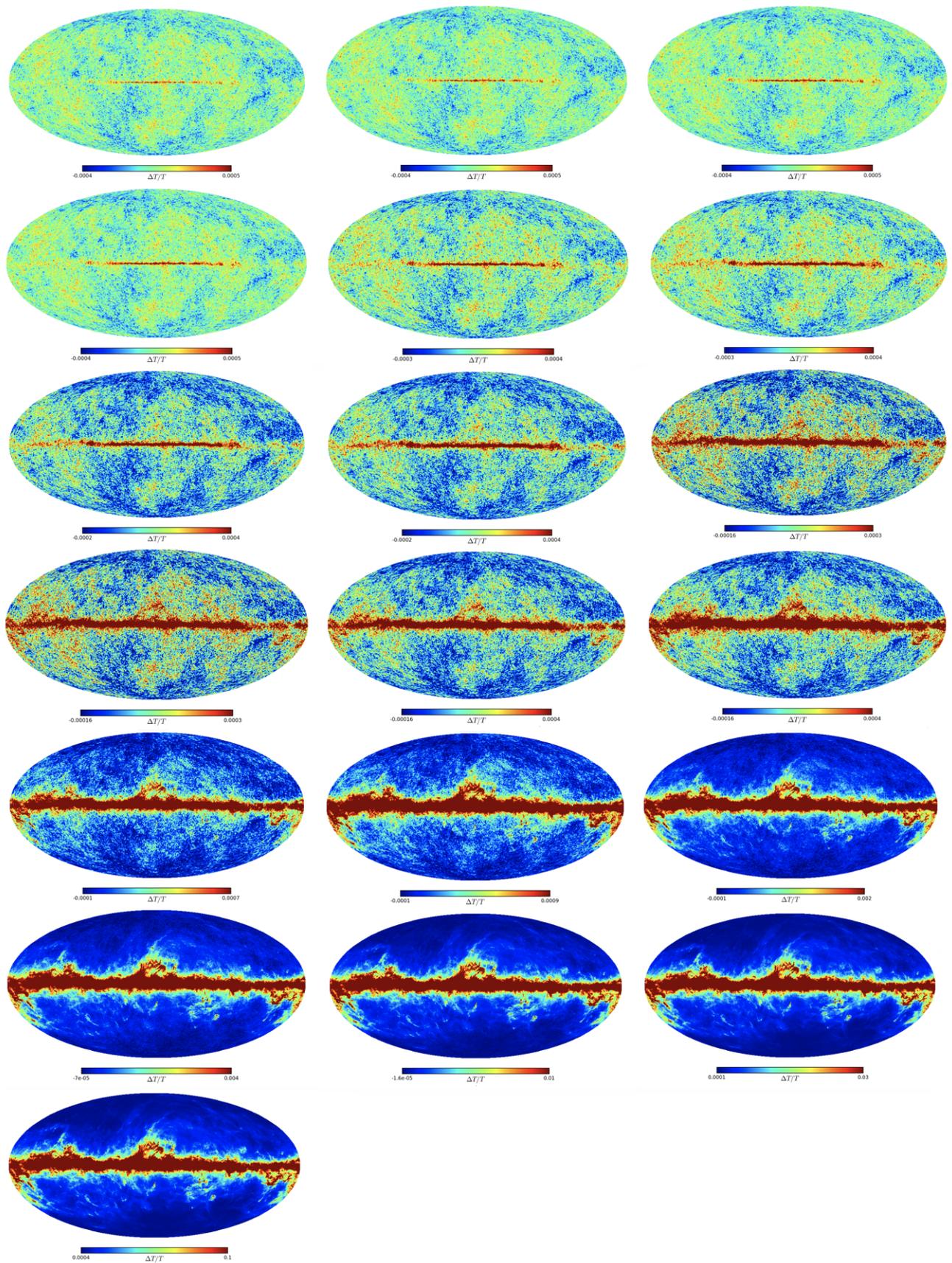


Figura 4.2: Simulaciones del cielo de microondas a las frecuencias de observación. De arriba a abajo y de izquierda a derecha: 60, 70, 80, 90, 100, 115, 130, 145, 160, 175, 190, 220, 255, 295, 340, 390, 450, 520 y 600 GHz.

Las simulaciones, basadas en el Planck Sky Model [14], cuentan con unos contaminantes realistas, donde, en particular, $s_{\nu,i}(\hat{r}) \neq s_i(\hat{r})f_i(\nu)$; de modo que la ecuación (2.3) no se cumple estrictamente, aumentando así la complejidad de la separación de componentes con respecto al capítulo anterior. Otro grado extra de cara a la complejidad de las simulaciones es que se ha contado con más canales de observación que de componentes, por lo que en el caso de FastICA, que recordemos sólo puede aplicarse en el caso $M = N$, se ha introducido un PCA para reducir la dimensión de los datos a N canales efectivos y blanquearlos. Como se ha dicho anteriormente, PCA se puede entender como un método de separación de componentes, pero es menos eficaz que FastICA, ya que falla a la hora de separar componentes que estén correlacionadas. Sobre este asunto se volverá más adelante.

Como estimadores de la bondad de los resultados se ha utilizado el coeficiente de correlación (ρ) y las diferencias de la skewness y la kurtosis de la señal recuperada y la original (ΔS y ΔK , respectivamente). Los estimadores anteriores se aplican en dos casos, uno correspondiente a la señal del FCM recuperada a todo el cielo, y otro correspondiente a la señal de FCM enmascarada. El hecho de que se haya enmascarado el FCM se debe a que, a pesar de haber aplicado la separación de componentes, existe una importante contaminación en algunas regiones del cielo, como, por ejemplo, el plano galáctico, de modo que la aplicación de una máscara permite conocer cómo de buena es la señal del FCM recuperada en aquellas zonas donde la contaminación ha sido minimizada por los métodos de separación de componentes. En estos casos los estadísticos han sido calculados teniendo en cuenta tan sólo aquellos píxeles de los mapas sobre los que no actúa la máscara, es decir, sobre mapas con un número reducido de píxeles con respecto a los recuperados. El uso de mapas enmascarados es una práctica común en el campo de la separación de componentes. En la Figura 4.3 se muestra la máscara utilizada en las simulaciones realizadas en este capítulo. Además, se han utilizado otros estadísticos que se basan en el espectro angular de potencias, que se introduce a continuación.

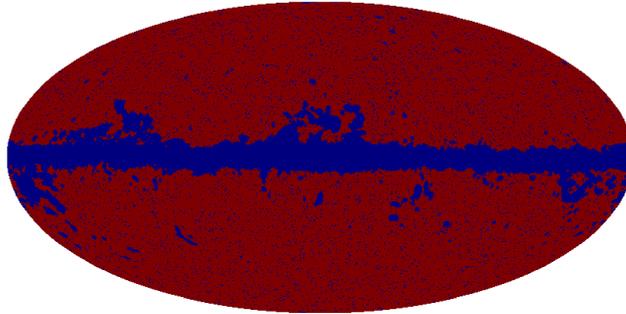


Figura 4.3: Máscara utilizada en las simulaciones de CORE.

4.1. Espectro angular de potencias

El espectro angular de potencias, C_ℓ , es el estadístico más usado en el campo del FCM, y es equivalente a la función de correlación de 2 puntos (de hecho, se trata de la transformada de Fourier de éste), con la ventaja de que, para señales isótropas, sus componentes son independientes. En particular, una función $T(\theta, \varphi)$ sobre la esfera se puede expresar en función de los armónicos esféricos $Y_{\ell m}(\theta, \varphi)$, que son funciones definidas sobre una esfera:

$$T(\theta, \varphi) = \sum_{\ell=0}^{\ell_{max}} \sum_{m=-\ell}^{\ell} a_{\ell m} Y_{\ell m}(\theta, \varphi), \quad (4.1)$$

donde $a_{\ell m}$ son los coeficientes de los armónicos esféricos, y que a su vez pueden calcularse a través de la transformación inversa:

$$a_{\ell m} = \int d\Omega T(\theta, \varphi) Y_{\ell m}^*(\theta, \varphi). \quad (4.2)$$

En la práctica, el sumatorio en ℓ en (4.1) va hasta un cierto ℓ_{max} , asociado a la escala más pequeña que permite la imagen pixelada. En nuestro caso, los mapas sobre la esfera están representados en la pixelización HEALPIX [15], que es la más usada en el campo del FCM. Nosotros estamos trabajando con mapas cuyos píxeles tienen un tamaño de, aproximadamente, 17.5 arcmin, lo que equivale a un $\ell_{max} \approx 600$. Los armónicos esféricos sirven para estimar el espectro angular de potencias:

$$C_\ell = \frac{1}{2\ell + 1} \sum a_{\ell m} a_{\ell m}^* = \frac{1}{2\ell + 1} \sum |a_{\ell m}|^2. \quad (4.3)$$

Es costumbre definir la cantidad

$$D_\ell = \frac{\ell(\ell + 1)}{2\pi} C_\ell, \quad (4.4)$$

que tiene la ventaja de que, para el FCM, ofrece una medida de la potencia más constante con la escala. Es posible definir el espectro de correlación normalizado como:

$$\rho_\ell = \frac{X_\ell}{\sqrt{C_{\ell_a} C_{\ell_b}}}, \quad (4.5)$$

donde C_{ℓ_a} y C_{ℓ_b} son los autoespectros de las señales. Otro estadístico relevante para nuestra caracterización es el espectro de potencias cruzado, que estima la correlación de dos señales a partir de sus respectivos coeficientes armónicos, $a_{\ell m}$ y $b_{\ell m}$:

$$X_\ell = \frac{1}{2\ell + 1} \sum a_{\ell m} b_{\ell m}^*. \quad (4.6)$$

El espectro angular de potencias contiene la información relevante a todas las escalas, en función del multipolo ℓ , que es inversamente proporcional a la escala angular: $\ell \approx 200/\theta$ (con θ en grados). El espectro de potencias del FCM tiene una forma muy característica como se muestra en la Figura 4.4, donde se representan las señales a una resolución de 17.5'. En la figura se observa cómo las emisiones de sincrotrón y del polvo se hacen subdominantes a escalas angulares pequeñas (valores de ℓ grandes) mientras que con las fuentes puntuales y el ruido ocurre el efecto contrario, aumenta su importancia a medida que nos desplazamos a escalas menores. El espectro del FCM presenta una estructura que no tiene el resto de componentes del cielo de microondas, y que es característica a escalas ($> 1^\circ$) de las propiedades estadísticas de las fluctuaciones primordiales de densidad, y, por debajo, de los procesos físicos sufridos por el plasma de fotones y electrones entorno a la recombinación.

Además, hay que notar que todas las señales físicas, a partir de $\ell \approx 800$, sufren un amortiguamiento exponencial, que es debido a la resolución del instrumento. Nótese que, claro está, este suavizado no afecta al ruido instrumental.

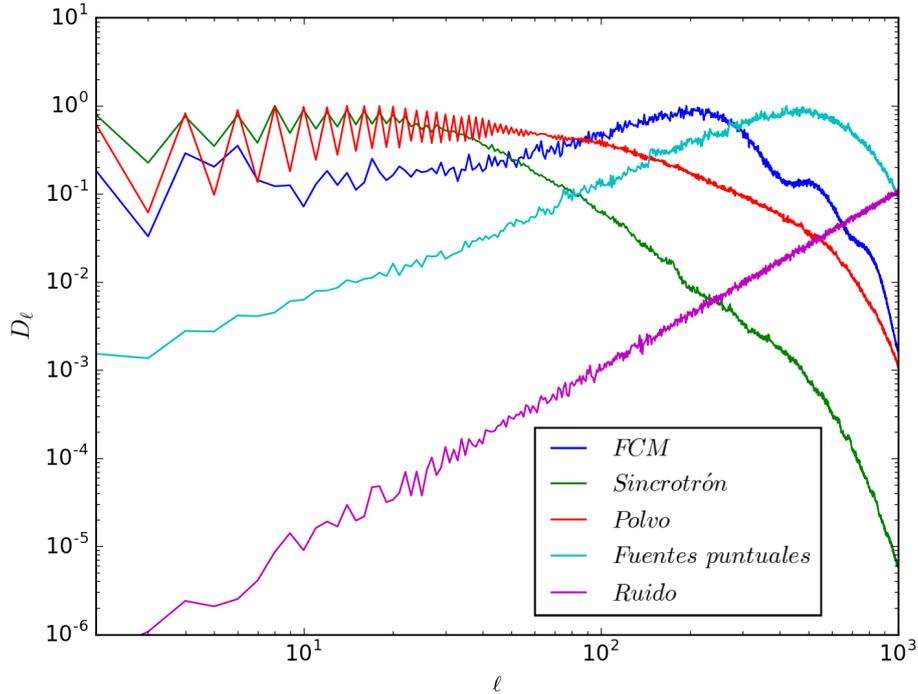


Figura 4.4: Espectro angular de potencias de las componentes del cielo de microondas a 100 GHz.

4.2. Análisis previo: observaciones sin ruido

En esta primera sección del estudio de las simulaciones de CORE se presentan los resultados obtenidos de la recuperación del FCM por FastICA en función del número de componentes. Se han explorado los casos en los que se recuperan desde 3 hasta 7 componentes, siempre sin añadir ruido a las observaciones. También se compara con la estimación obtenida por ILC. Además de proporcionar tablas de los estimadores ρ , ΔS y ΔK se han representado las cantidades $|\Delta D_\ell/D_\ell|$ y ρ_ℓ :

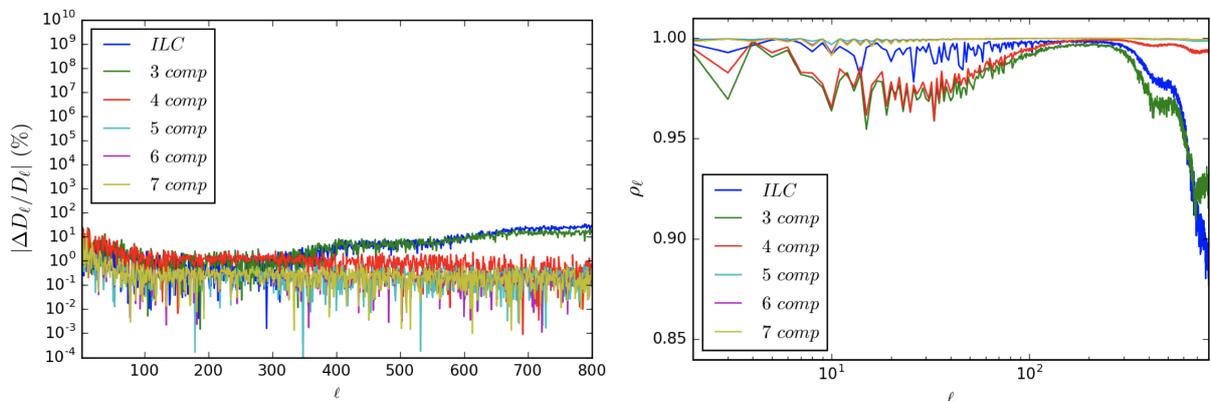


Figura 4.5: Diferencia entre el espectro de potencias del FCM recuperado y el original (izda) y espectro angular de potencias cruzado normalizado (dcha).

En la Figura 4.5 se observa cómo aumentando el número de componentes en FastICA, el error relativo se va haciendo cada vez más pequeño. Este error es mayor a pequeña escala en el caso en el que FastICA recupera 3 componentes y en el de ILC.

Centrándonos ahora en los espectros de potencias cruzados, se hace evidente que a medida que se aumenta el número de componentes mejoran los resultados de la recuperación del FCM. Si nos fijamos en las señales de 3 y 4 componentes, la señal del FCM recuperada es muy similar en ambos casos a gran escala pero mejora considerablemente a pequeña escala en el caso de 4 componentes. Por encima de 4 componentes la señal del FCM hallada por FastICA resulta una muy buena estimación de la componente original, siendo su correlación con ésta muy próxima a la unidad. En la Tabla 4.1 se encuentran tabulados los estimadores ρ , ΔS y ΔK para cada caso:

N	Cielo completo			Cielo enmascarado		
	ρ	ΔS	ΔK	ρ	ΔS	ΔK
3	0.98727	-0.02812	-0.04307	0.99992	0.00011	-0.02728
4	0.99192	0.00241	-0.01254	0.99981	0.00008	-0.02730
5	0.99957	0.00190	-0.01305	0.99998	0.00061	-0.02678
6	0.99932	0.00321	-0.01174	0.99994	0.00113	-0.02626
7	0.99922	0.00347	-0.01149	0.99993	0.00122	-0.02617
ILC	0.99294	-0.05843	-0.07338	0.99730	0.00082	-0.02657

Tabla 4.1: Estadísticos. N es el número de componentes recuperadas por FastICA.

En base a los datos de la Tabla 4.1, se hace evidente que la recuperación del FCM de FastICA en el caso del cielo completo es mejor que la de ILC a partir de 5 componentes. La señal del FCM obtenida por ILC empeora notablemente a pequeña escala (valores grandes de ℓ), como se observa en el espectro de la Figura 4.5. Al enmascarar las zonas más contaminadas del cielo, los resultados mejoran si los comparamos con los obtenidos para el cielo completo. A diferencia del caso unidimensional, no es posible realizar varias simulaciones que nos permitan obtener una estimación de las incertidumbres de los estadísticos, ya que ejecutar el problema de la separación de componentes más de una vez es computacionalmente muy costoso.

A continuación, en la Figura 4.6, se muestran las componentes halladas por FastICA para el caso en el que se impone que se recuperen 3 componentes, en ausencia de ruido instrumental. La primera componente corresponde inequívocamente a la señal del FCM, pues la estructura que presenta la misma no deja lugar a dudas. En cuanto a las dos componentes restantes, si nos basamos en su morfología y las comparamos con las señales de la Figura 4.1, en ambas se distingue la componente galáctica del polvo. La tercera señal es la componente del polvo, y en la segunda la contaminación debida al polvo aparece en los extremos del plano galáctico. En este caso, la señal de sincrotrón no se distingue en las componentes recuperadas por FastICA, y su contribución vendrá distribuida entre las 3 señales recuperadas. Un aspecto a comentar sobre las componentes obtenidas por FastICA, es que, como se dijo con anterioridad, son de dispersión unidad, de modo que la escala de las mismas no es la original. Por ello, las unidades de las señales no son $\Delta T/T$ sino una cantidad α veces $\Delta T/T$. En cualquier caso, la estimación de esta constante de normalización puede realizarse de diferentes maneras (ver, por ejemplo, [10]).

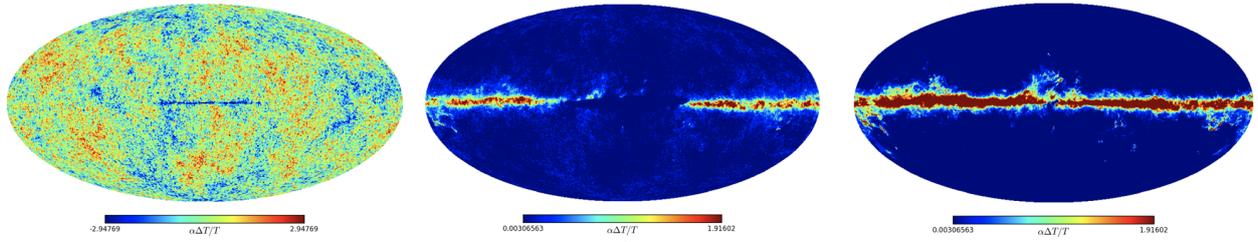


Figura 4.6: Mapas recuperados por FastICA (3 componentes).

En la Figura 4.7 se muestran los mapas obtenidos para el caso de 7 componentes, así como el FCM proporcionado por ILC.

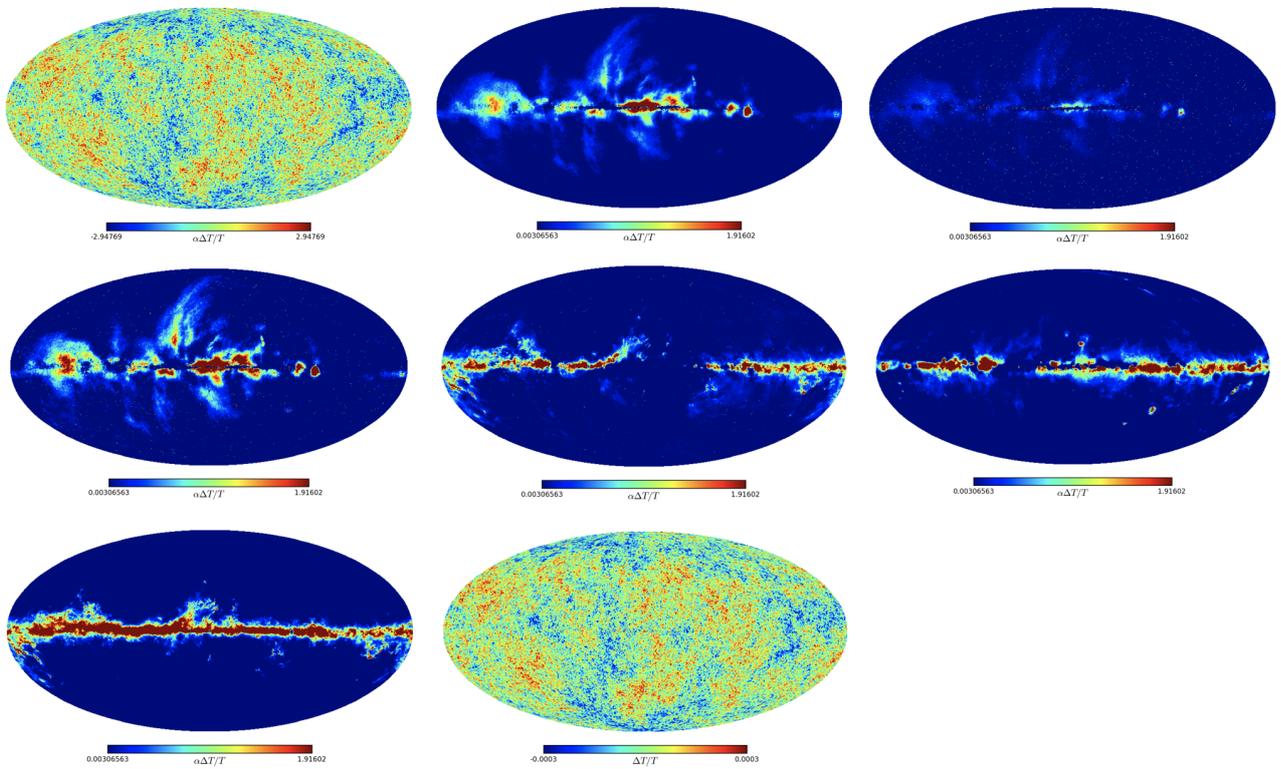


Figura 4.7: Mapas recuperados por FastICA (7 componentes) e ILC en el caso sin ruido. Los 7 primeros mapas son las 7 componentes recuperadas por FastICA y el último el FCM obtenido por ILC.

Nuevamente, la primera señal corresponde al FCM de FastICA. La última componente es el FCM obtenido por ILC, que a diferencia de FastICA, sí recupera el fondo cósmico de microondas con su normalización original. Atendiendo a la morfología de las señales restantes, se observa que en este caso sí aparece la componente de sincrotrón. La contaminación de sincrotrón es evidente en las componente 2 y 4 y en menor medida en la componente 3, donde también se distingue contaminación debida a fuentes puntuales. Las señales 5 y 6 presentan contaminación debida al polvo en el plano galáctico y la última componente recuperada por FastICA es la señal del polvo, semejante a la hallada para 3 componentes. Nótese que la contaminación que mostraba el FCM de FastICA para 3 componentes ha desaparecido al imponer que recupere 7 componentes en lugar de 3. Esto es debido a que la componente de

FCM hallada por FastICA para 3 componentes no sólo contiene esta señal, si no que también contiene las contribuciones debidas a otras componentes que FastICA no ha separado, como pueden ser la componente de sincrotrón o las fuentes puntuales.

En la sección 1.4 se habló de los métodos de separación de componentes, estableciéndose que PCA además de un método para blanquear los datos y reducir su dimensión podía ser considerado un método de separación de componentes. PCA busca combinaciones lineales de los datos que proporcionen las componentes principales (componentes no correlacionadas). Por lo tanto, aplicando PCA a los mapas del cielo de microondas se puede encontrar una primera estimación de las componentes originales. En esta sección se ha aplicado PCA a los mapas con el fin de blanquear los datos, reducir su dimensión y que de este modo FastICA pueda ser aplicado. Además de comparar los resultados obtenidos entre la señal del FCM estimada por FastICA e ILC se comparan los resultados obtenidos por PCA y FastICA.

Antes de aplicar FastICA se debe decidir cuántas componentes va a separar FastICA. Imponer un número de componentes inferior al real provocaría que las componentes no extraídas aparecieran en las señales recuperadas contaminándolas. Como una primera aproximación se ha impuesto recuperar 3 componentes (Figura 4.8), para a posteriori recuperar 6 componentes (Figura 4.9).

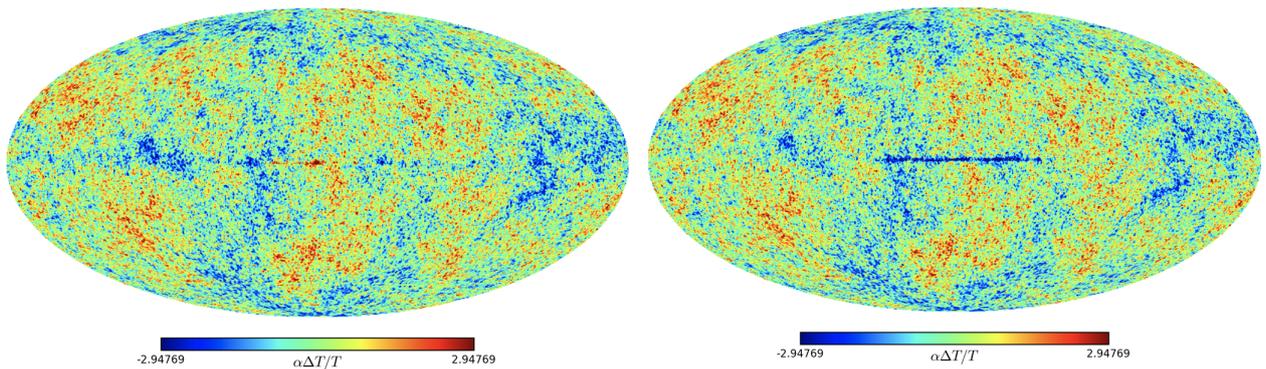


Figura 4.8: Se muestra el FCM recuperado por PCA (izda) y FastICA (dcha) para el caso de 3 componentes.

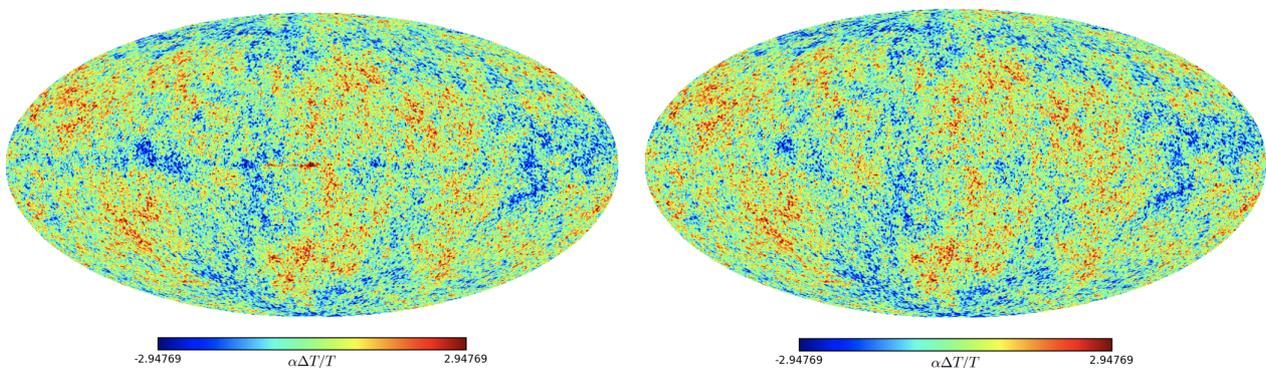


Figura 4.9: Se muestra el FCM recuperado por PCA (izda) y FastICA (dcha) para el caso de 6 componentes.

En la Figura 4.8 se aprecia cierta contaminación en el plano galáctico, siendo más claro en el caso de FastICA. Comparando las componentes del FCM recuperadas por FastICA para 3 y 6 componentes, se observa que la contaminación que aparece en el caso de 3 componentes desaparece si aumentamos hasta 6 componentes. Esto se debe a que los mapas simulados del cielo de microondas contienen más de 3 componentes, de modo que si imponemos al algoritmo de FastICA que recupere tan sólo 3 componentes, dichas componentes van a contener contribuciones de las demás señales. Esa es la razón por la que la recuperación de la señal del FCM mejora pasando de imponer que FastICA recupere 3 a 6 componentes. En el caso de PCA la componente del FCM no mejora tanto con respecto al caso de 3 componentes debido a que PCA obtiene componentes que no están correlacionadas entre sí pero que pueden no ser independientes (debido a la naturaleza no gaussiana de las emisiones galácticas y de fuentes puntuales). FastICA, como se dijo en la sección 1.4, no sólo obtiene componentes no correlacionadas sino que además dichas componentes son independientes entre sí.

4.3. Análisis de los datos con ruido

Después del análisis sin ruido se ha decidido ir un paso más allá y explorar cómo varían los resultados con la introducción de ruido en las observaciones. Teniendo en cuenta las propiedades estadísticas del ruido instrumental, es de esperar que la señal del FCM recuperada empeore a medida que ℓ crece, ya que es a pequeña escala donde se va a manifestar el ruido.

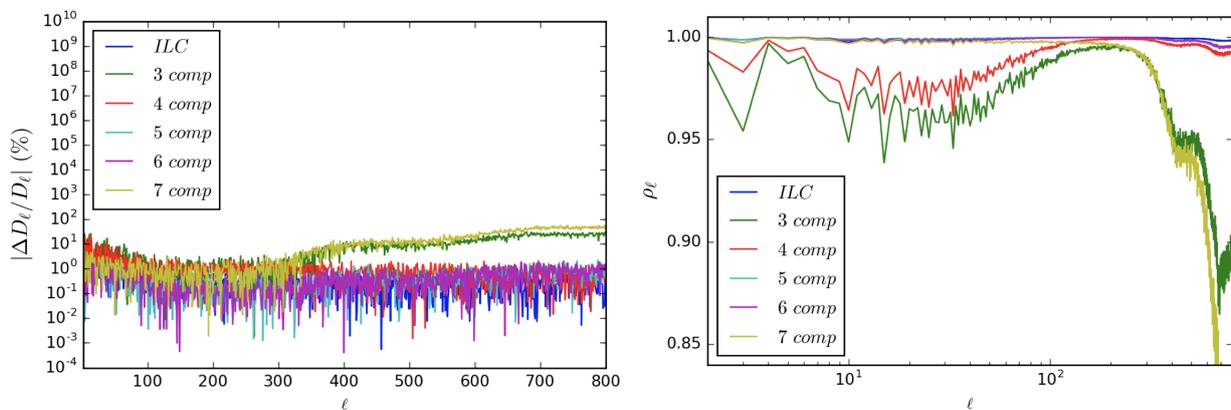


Figura 4.10: Estadísticos, pero incluyendo el ruido instrumental. N es el número de componentes recuperadas por FastICA.

Observando los espectros de potencias cruzados de la Figura 4.10, podemos ver cómo, al igual que en el caso anterior, las señales para 3 y 4 componentes se asemejan para valores pequeños de ℓ , y a medida que ℓ crece, la correlación para la señal de 3 componentes disminuye mientras que para 4 componentes crece y se aproxima a la unidad. Al pasar de 4 componentes se produce un salto cuantitativo en la recuperación del FCM. Para 5 y 6 componentes, las señales halladas por FastICA presentan una correlación muy alta, y para más de 6 componentes, la correlación de las señales cae a pequeña escala, como se puede comprobar en los estimadores de la Tabla 4.2.

N	Cielo completo			Cielo enmascarado		
	ρ	ΔS	ΔK	ρ	ΔS	ΔK
3	0.98152	-0.05516	-0.07012	0.99989	-0.000003	-0.02739
4	0.99209	0.00317	-0.01178	0.99980	-0.00021	-0.02760
5	0.99953	0.00114	-0.01382	0.99998	-0.00019	-0.02757
6	0.99933	-0.00083	-0.01578	0.99998	-0.00013	-0.02751
7	0.99242	-0.00144	-0.01639	0.99324	-0.00133	-0.02872
ILC	0.99935	-0.00095	-0.01591	0.99996	-0.00001	-0.02740

Tabla 4.2: Estadísticos. N es el número de componentes recuperadas por FastICA.

De esta forma, con estas condiciones de ruido, se tiene que al aumentar el número de componentes, la recuperación del fondo cósmico de microondas por FastICA mejora hasta alcanzar su máximo en 5 componentes, para después caer ligeramente. Comparando el FCM recuperado por FastICA con el de ILC, se tiene que este último proporciona una mejor estimación en todos los casos a excepción del de 5 componentes, con el que es muy similar.

Curiosamente, en el caso con ruido, ILC es capaz de estimar el FCM con mayor precisión que en el caso sin ruido. Si miramos el espectro de potencias cruzado de ILC se aprecia que las pequeñas oscilaciones para valores intermedios del multipolo ℓ que aparecían en el caso sin ruido han desaparecido y que la correlación a pequeña escala ha mejorado de manera notable. Del mismo modo, su error relativo disminuye con respecto al caso anterior.

Las señales del cielo de microondas recuperadas por FastICA para 3 componentes se representan en la Figura 4.11. Al igual que en el caso sin ruido, se ha recuperado el fondo cósmico de microondas (primera señal en la figura anterior, mostrando contaminación en el plano galáctico), y dos señales en las que se observa contribución debida a la componente del polvo.

Para el caso de 7 componentes, las señales obtenidas se muestran en la Figura 4.12. Cualitativamente se observa que éstas señales recuperadas son más ruidosas que las halladas en el caso anterior. Si comparamos las señales obtenidas en ambos casos, vemos que la señal de sincrotrón que antes se distinguía hasta en 3 componentes, ahora tan sólo se aprecia mínimamente en la tercera componente. Las señales del FCM de FastICA e ILC (primera y última componentes de la Figura 4.12, respectivamente) son fácilmente reconocibles. La contaminación del FCM de 3 componentes ha desaparecido al imponer la recuperación de 7 componentes. Las componentes 4, 5 y 6 presentan contaminación en el plano galáctico que se corresponde con el polvo. En este caso, además aparece una componente ruidosa donde FastICA parece introducir contribuciones de ruido y fuentes puntuales.

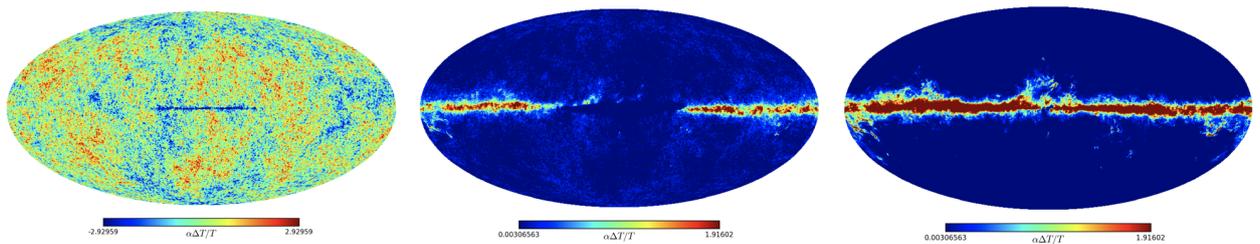


Figura 4.11: Mapas recuperados por FastICA (3 componentes).

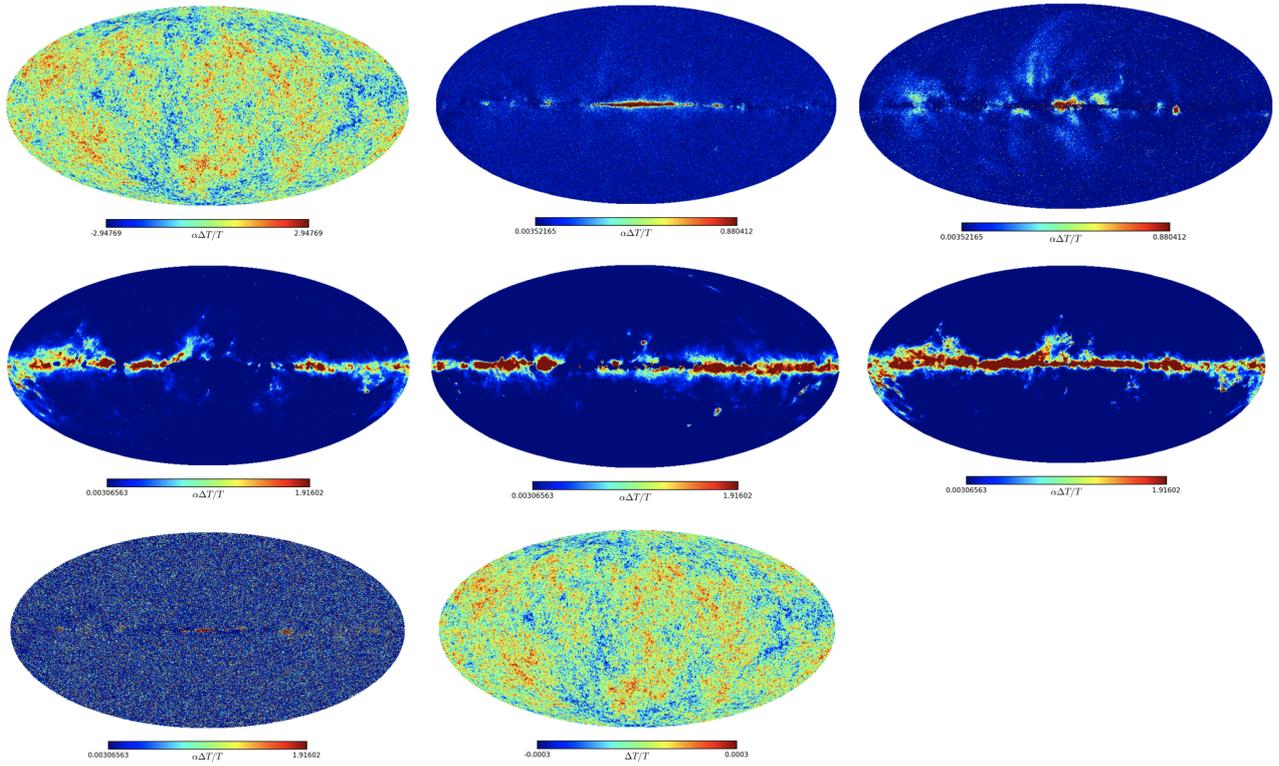


Figura 4.12: Mapas recuperados por FastICA (7 componentes) e ILC en el caso con ruido.

4.4. Análisis de los datos con ruido x10

Como último caso de las simulaciones 2-D, se ha recuperado la señal del FCM a partir de las observaciones del cielo de microondas en las que se ha añadido un ruido 10 veces superior al normal. La razón es que este incremento del ruido es más acorde a situaciones experimentales más actuales, como puede ser el caso del satélite de Planck.

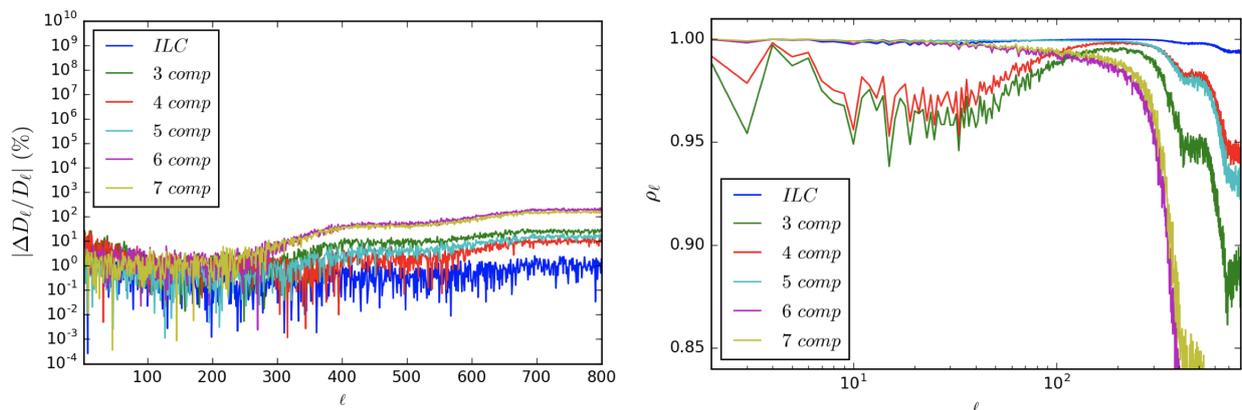


Figura 4.13: Caso con ruido x10.

Si nos fijamos en los errores relativos y en los espectros de potencias cruzados de la Figura 4.13, vemos que el ILC proporciona una reconstrucción de la señal de FCM superior a las de FastICA. El error relativo para ILC es el más bajo y su correlación con la señal original es

cercana a la unidad tanto a pequeña como a gran escala. Por su parte, FastICA, a medida que se aumenta el número de componentes a recuperar, mejora la correlación a gran escala pero empeora a escalas pequeñas, donde también aumenta su error relativo. Esto era de esperar, ya que se ha introducido un ruido diez veces superior al caso con ruido, y el ruido cobra importancia a pequeña escala, para valores de ℓ grandes. Nuevamente, FastICA obtiene la mejor reconstrucción de la señal de FCM para el caso de 5 componentes, como ocurría en los casos sin ruido y con ruido anteriores.

N	Cielo completo			Cielo enmascarado		
	ρ	ΔS	ΔK	ρ	ΔS	ΔK
3	0.98141	-0.05504	-0.07000	0.99976	0.00020	-0.02718
4	0.98874	0.00652	-0.00843	0.99826	0.00048	-0.02691
5	0.99740	-0.00017	-0.01512	0.99785	-0.00094	-0.02832
6	0.97125	-0.00413	-0.01908	0.97200	-0.00288	-0.03027
7	0.97729	0.00215	-0.01280	0.97790	0.00172	-0.02567
ILC	0.99917	-0.00102	-0.01597	0.78231	-0.00001	-0.02740

Tabla 4.3: Estadísticos. N es el número de componentes recuperadas por FastICA.

Tal y como sucedía con el caso anterior de ruido, la correlación de las señales estimadas por FastICA alcanza su valor máximo para 5 componentes y después comienza a disminuir al crecer el número de componentes. En este caso en el que el ruido es 10 veces mayor que en el caso anterior, ésto se hace más evidente todavía. La razón de que esto ocurra es que la solución que proporciona FastICA es una solución global, es decir, que FastICA busca una solución que maximice la no gaussianidad de todas las componentes, sin importar si para ello alguna de las componentes empeora de manera individual. Esto es, al aumentar el número de componentes por encima de 5, FastICA está introduciendo el ruido en la componente del FCM con el objetivo de conseguir una solución global que reproduzca mejor el cielo de microondas.

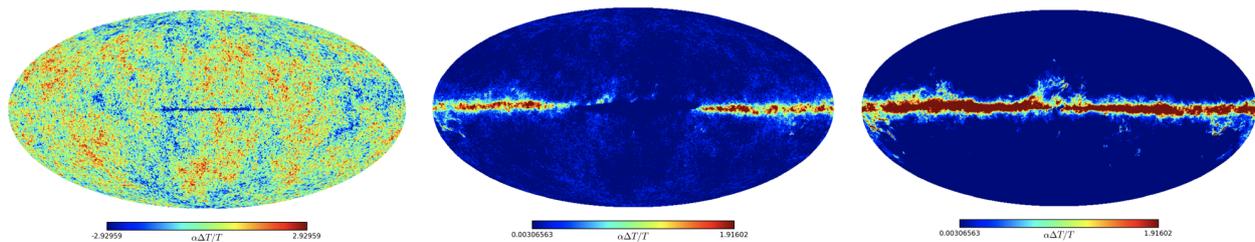


Figura 4.14: Mapas recuperados por FastICA (3 componentes).

Las componentes halladas que aparecen en la figura anterior son análogas al caso sin ruido y con ruido comentados anteriormente.

Atendiendo a la morfología de las señales de la Figura 4.15, es posible distinguir la señal de FCM y el polvo de FastICA (componentes 1 y 6, respectivamente), al igual que en los dos casos anteriores. En este último caso, el aumento de ruido ha hecho desaparecer prácticamente la contribución de la componente de sincrotrón que, aunque ligeramente, podía discernirse en los casos anteriores. Las componentes 4 y 5 contienen contribución debida al polvo galáctico, y las componentes 2 y 7 son tan ruidosas que no es posible identificar ninguna componente galáctica.

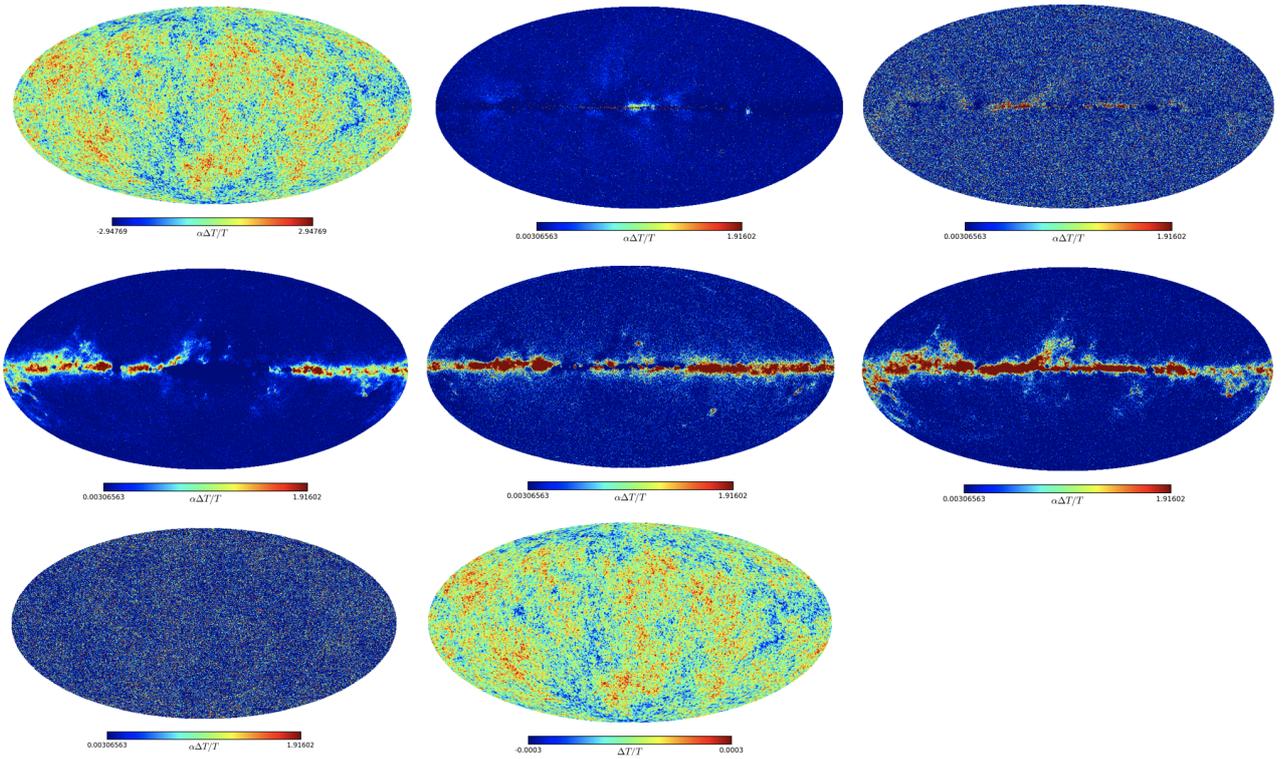


Figura 4.15: Mapas recuperados por FastICA (7 componentes) e ILC en el caso con ruido $\times 10$.

4.5. Discusión

Los resultados obtenidos en la separación de componentes del cielo de microondas para simulaciones de CORE se comportan como cabría esperar. Cuando se considera el cielo completo, el aumento progresivo del ruido en las observaciones conduce a una peor estimación de la señal del FCM, manifestándose de manera más clara a pequeñas escalas, haciéndose patente un aumento del error relativo y la descorrelación. Del mismo modo que el ruido dificulta la separación de componentes del cielo de microondas, en el caso de FastICA, el aumento progresivo del número de componentes da como resultado mejores estimaciones de la señal hasta un máximo.

Enmascarando el cielo se consigue disminuir la complejidad del mismo, obteniéndose que la señal recuperada por FastICA e ILC se aproxime más a la original en aquellas zonas del cielo menos contaminadas.

Capítulo 5

Análisis de datos de Planck

Para finalizar este trabajo, se ha aplicado la separación de componentes al caso real del cielo de microondas de Planck. El satélite de la ESA Planck ha producido observaciones detalladas de todo el cielo en 9 canales de frecuencia comprendidos entre los 30 y los 857 GHz [8]. La complejidad de este caso es mayor con respecto al caso anterior de las simulaciones del estilo de CORE, pues se trata de datos reales y tan sólo se cuenta con 9 canales de observación, de manera que la información de la que se dispone sobre el cielo de microondas es menor. A partir de los mapas del cielo de microondas proporcionados por Planck se ha buscado recuperar el FCM mediante FastICA e ILC, y se ha comparado con la señal obtenida por el método de separación Commander-Ruler, uno de los cuatro oficialmente publicados por la ESA (los otros tres son SEVEM, SMICA y NILC). Commander-Ruler se trata de un método paramétrico. Ver [16] para más detalles. A continuación, en la Figura 5.1, se muestran la señal del fondo cósmico de microondas de Commander-Ruler y la máscara del cielo utilizada para calcular los estadísticos:

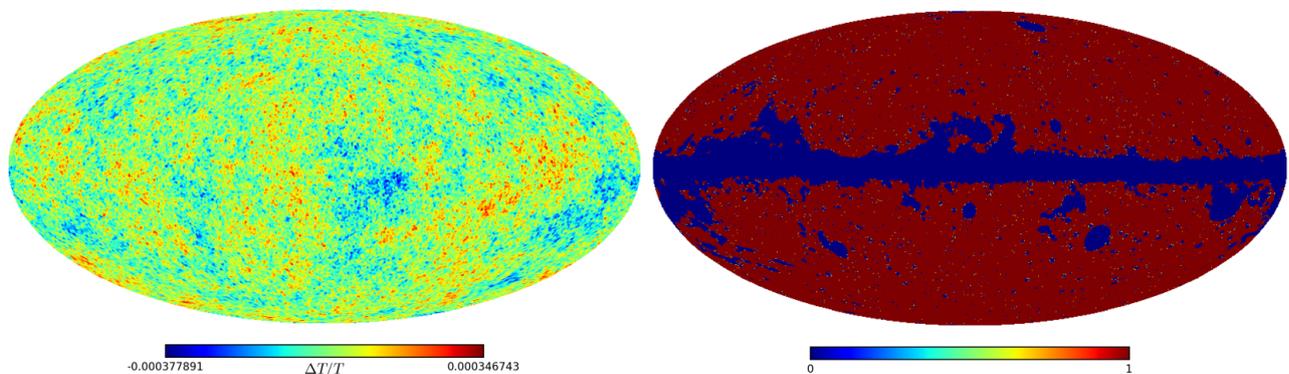


Figura 5.1: Componente del FCM obtenida mediante Commander-Ruler a partir de los datos de Planck (izda). A la derecha se muestra la máscara utilizada en este trabajo para enmascarar las zonas más contaminadas de los mapas obtenidos mediante FastICA e ILC. Se trata de la máscara proporcionada por la propia misión Planck.

En la Figura 5.2 se representan las cantidades $|\Delta D_\ell/D_\ell|$ y ρ_ℓ en función del multipolo ℓ para los distintos casos de la separación de componentes de los datos de Planck. A medida que se aumenta el número de componentes en FastICA, el valor ρ_ℓ aumenta hasta llegar a conseguirse una correlación aceptable para 7 componentes, siendo muy parecida a la correlación proporcionada por ILC. Para el ILC y FastICA en el caso de 7 componentes, la correlación es bastante buena a gran escala, disminuyendo ligera y paulatinamente al aumentar ℓ . Si nos centramos en los errores relativos, los valores más bajos también corresponden al caso de ILC

y FastICA con 7 componentes, y todos ellos aumentan al crecer ℓ . Todas estas observaciones están de acuerdo con los valores de los estadísticos expuestos en la Tabla 5.1.

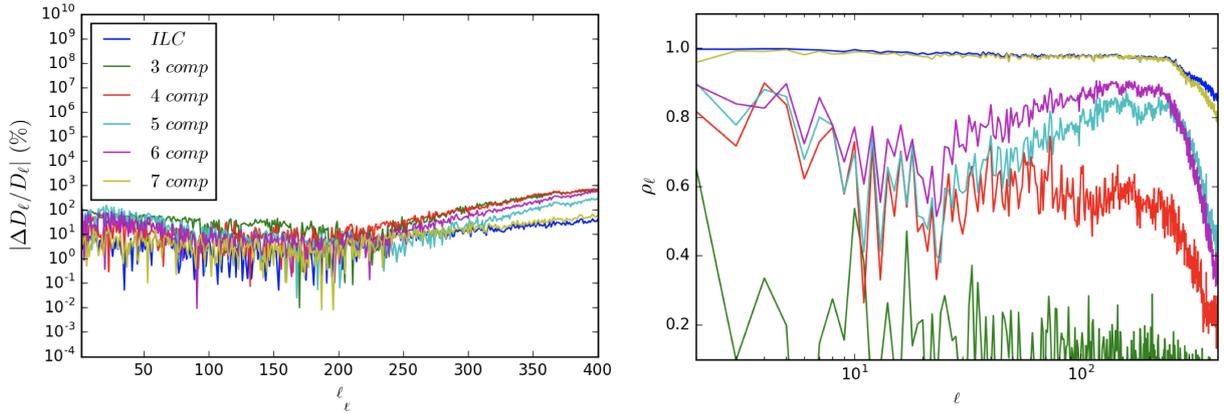


Figura 5.2: Planck.

N	Cielo completo			Cielo enmascarado		
	ρ	ΔS	ΔK	ρ	ΔS	ΔK
3	0.1017	-51.4344	-51.3706	0.8411	-0.3752	-0.3454
4	0.5675	39.8450	39.9089	0.9910	0.0073	0.0371
5	0.7235	2.6565	2.7203	0.9929	0.0104	0.0403
6	0.7865	2.2564	2.3203	0.9873	0.0125	0.0424
7	0.9723	0.0668	0.1306	0.9984	0.0012	0.0311
ILC	0.9779	-0.0407	0.0231	0.9991	-0.0010	0.0289

Tabla 5.1: Estadísticos. N es el número de componentes recuperadas por FastICA.

Si nos fijamos en los resultados obtenidos al enmascarar aquellas zonas del cielo que presentan una mayor contaminación (la región del plano galáctico, principalmente), se hace patente que si sólo tenemos en cuenta los píxeles de la máscara, la correlación de las componentes aumenta. Esto ocurre como cabría esperar, ya que al reducir los mapas a aquellas zonas del cielo que están menos contaminadas, estamos eliminando la descorrelación introducida por dichas regiones.

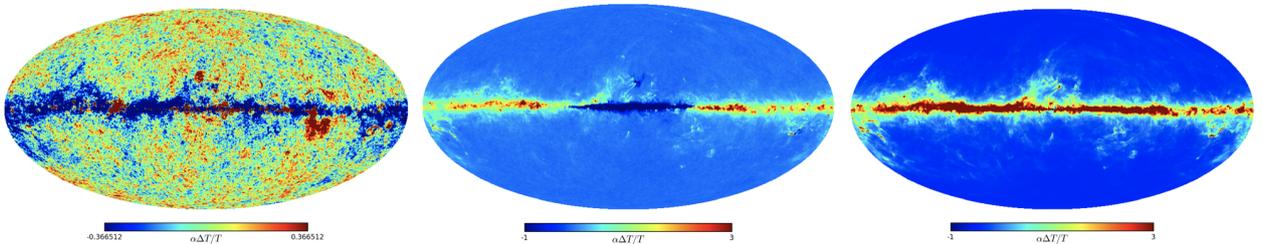


Figura 5.3: Mapas recuperados por FastICA (3 componentes) para los datos obtenidos por el satélite Planck.

En la Figura 5.3, la primera componente corresponde a la señal del FCM. Si bien esta componente presenta una gran contaminación en la región correspondiente al plano galáctico,

cuando nos desplazamos a las regiones más alejadas del mismo, observamos la estructura propia del fondo cósmico de microondas. Las dos componentes restantes contienen contribución debida al polvo galáctico.

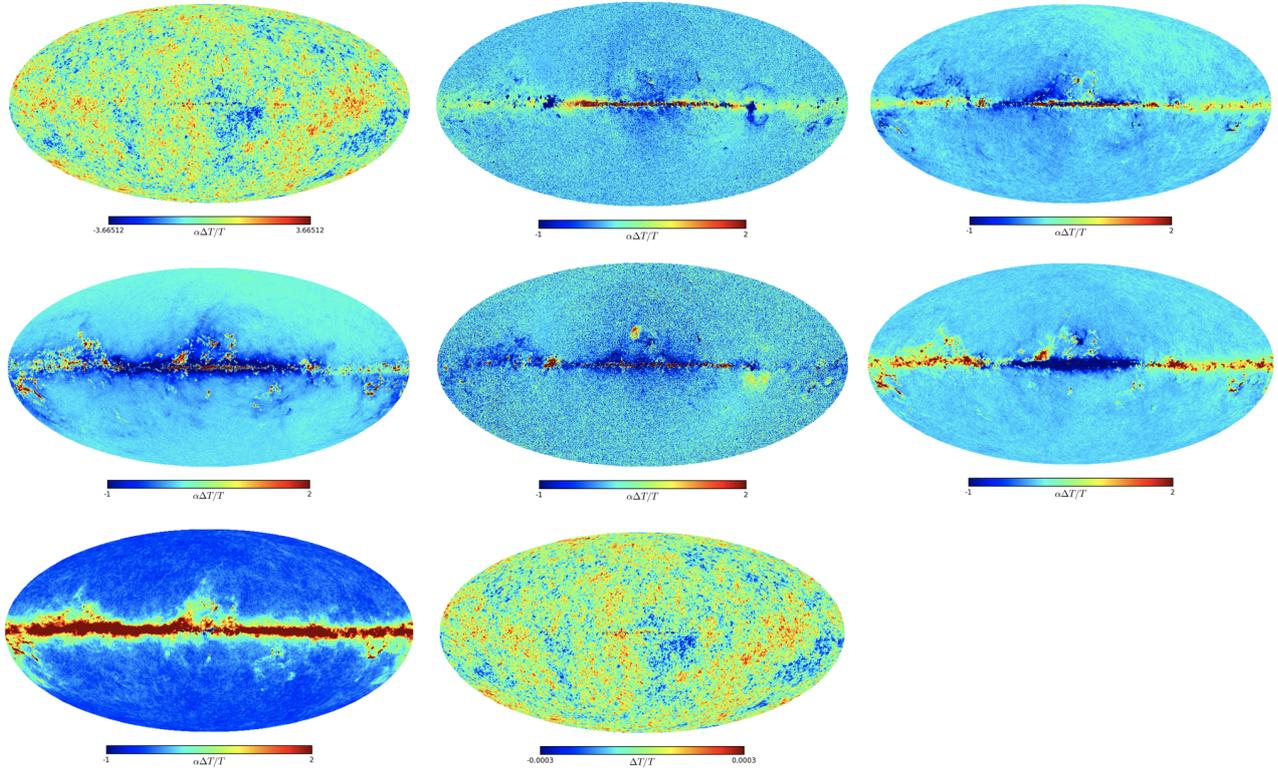


Figura 5.4: Mapas recuperados por FastICA (7 componentes) para los datos obtenidos por el satélite Planck.

Comparando la señal de FCM de FastICA para 7 componentes (primera componente) con la obtenida para 3 componentes de la Figura 5.3, se evidencia la mejora conseguida, habiéndose eliminado en gran medida las contribuciones debidas a las componentes contaminantes en el plano galáctico. Morfológicamente, se observa la componente de sincrotrón en la componente 4, y la componente del polvo en las componentes 3, 6 y 7 (esta última corresponde al polvo).

Conclusiones

Concluir, en primer lugar, que el problema de la separación de componentes encierra una gran complejidad por los múltiples factores que intervienen en el mismo (propiedades estadísticas de las componentes a separar, posibles relaciones de dependencia y correlación de las mismas, importancia del ruido en la separación, etc.), y de los que va a depender en gran medida la elección de uno u otro método de separación de componentes. La elección de los métodos también va a depender del objetivo de la separación, pues, por ejemplo, puede carecer de sentido aplicar un algoritmo como FastICA, capaz de recuperar N componentes, si tan sólo se pretende encontrar una, como en el caso de los mapas del fondo cósmico de microondas. Si por el contrario, de lo que se trata es de hallar diversas componentes, y disponer así de una mayor información sobre el conjunto y las señales que lo componen, un método como ILC no resulta de utilidad.

Los dos métodos lineales utilizados en este trabajo son bien distintos. Si bien FastICA maximiza la no gaussianidad de las componentes para así encontrar señales independientes, ILC minimiza la varianza. En estas diferencias radican las ventajas y desventajas de ambos métodos. Para minimizar la varianza de las observaciones, ILC necesita conocer cierta información sobre la componente que se desea recuperar, como lo es su dependencia con la frecuencia, mientras que FastICA no requiere de información adicional sobre las componentes. Por su parte, FastICA, para maximizar la no gaussianidad, se ve limitado por las propiedades estadísticas de las componentes, no pudiendo aplicarse en aquellos casos en los que existe más de una componente que tenga una distribución gaussiana. En el caso de la separación de componentes del cielo de microondas eso supone un problema, pues tanto el fondo cósmico de microondas como el ruido instrumental presentan distribuciones gaussianas. Es por este motivo, que cuando el ruido cobra importancia en las simulaciones, como ocurría en el capítulo 4 en el caso con ruido y con ruido $\times 10$, las estimaciones de la señal del FCM aportados por FastICA no eran óptimas. A este respecto, ILC consigue buenos resultados ya que es muy eficiente eliminando el ruido. Como se dijo con anterioridad, FastICA es un método de separación de componentes mientras que ILC es un método que reduce las fuentes contaminantes y el ruido, limpiando de esta forma la señal del FCM. La conclusión que se extrae de todo lo mencionado anteriormente, y como ha quedado demostrado en el estudio exhaustivo que se ha realizado en este trabajo, es que a medida que aumenta la complejidad de los datos, ILC es una mejor opción que FastICA para recuperar la componente del fondo cósmico de microondas. FastICA tan sólo consigue resultados parecidos a los de ILC o incluso superiores en casos en los que la complejidad no era grande, como en el caso de las simulaciones de CORE sin ruido. Sin embargo, a la hora de tratar datos de mayor complejidad como las simulaciones de CORE en las que el ruido cobraba importancia o el caso de los datos reales de Planck, ILC superaba en resultados a FastICA. Recordar también, que FastICA tiene el hándicap de preocuparse de obtener diversas componentes mientras que ILC sólo recupera una señal.

Por último, hacer mención a la modificación del algoritmo de FastICA llevada a cabo

en este trabajo en el capítulo 3, para el caso 1-D con 2 señales gaussianas, en la que se trataba de romper la degeneración y estimar así más de una componente gaussiana a través de FastICA. Para ello, además de maximizar la no gaussianidad de las componentes, se imponía que la primera componente obtenida por nuestro algoritmo fuera constante con la frecuencia. Con esta modificación, se buscaba proporcionar a FastICA la misma información de la que disponía ILC, es decir, conocer la dependencia frecuencial de la componente del FCM. El resultado obtenido de la aplicación de nuestro algoritmo modificado fue positivo, consiguiendo estimar las dos componentes gaussianas.

Bibliografía

- [1] Steven Weinberg. *Cosmology*. Oxford University Press, 2008.
- [2] Planck Collaboration, P. A. R. Ade, N. Aghanim, M. Arnaud, M. Ashdown, J. Aumont, C. Baccigalupi, A. J. Banday, R. B. Barreiro, J. G. Bartlett, N. Bartolo, E. Battaner, R. Battye, K. Benabed, A. Benoit, A. Benoit-Levy, J.-P. Bernard, M. Bersanelli, P. Bielewicz, J. J. Bock, A. Bonaldi, L. Bonavera, J. R. Bond, J. Borrill, F. R. Bouchet, F. Boulanger, M. Bucher, C. Burigana, R. C. Butler, E. Calabrese, J.-F. Cardoso, A. Catalano, A. Challinor, A. Chamballu, R.-R. Chary, H. C. Chiang, J. Chluba, P. R. Christensen, S. Church, D. L. Clements, S. Colombi, L. P. L. Colombo, C. Combet, A. Coulais, B. P. Crill, A. Curto, F. Cuttaia, L. Danese, R. D. Davies, R. J. Davis, P. de Bernardis, A. de Rosa, G. de Zotti, J. Delabrouille, F.-X. Desert, E. Di Valentino, C. Dickinson, J. M. Diego, K. Dolag, H. Dole, and et al. (202 additional authors not shown). Planck 2015 results. xiii. cosmological parameters. *A&A*, 02 2015.
- [3] Anthony Challinor. Cmb anisotropy science: a review. *Astro-Ph*, 10 2012.
- [4] Spergel, D. N. and Verde, L. and Peiris, H. V. and Komatsu, E. and Nolta, M. R. and Bennett, C. L. and Halpern, M. and Hinshaw, G. and Jarosik, N. and Kogut, A. and Limon, M. and Meyer, S. S. and Page, L. and Tucker, G. S. and Weiland, J. L. and Wollack, E. and Wright, E. L. First-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Determination of Cosmological Parameters. *APJS*, 148:175–194, September 2003.
- [5] Planck Collaboration and Adam, R. and Ade, P. A. R. and Aghanim, N. and Akrami, Y. and Alves, M. I. R. and Argüeso, F. and Arnaud, M. and Arroja, F. and Ashdown, M. and et al. Planck 2015 results. I. Overview of products and scientific results. *A&A*, 594:A1, September 2016.
- [6] D. Scott and G. F. Smoot. Cosmic Microwave Background Mini-review. *The Review of Particle Physics: PDG*, 2016.
- [7] C. Dickinson. CMB foregrounds - A brief review. *Astro-Ph*, June 2016.
- [8] Planck Collaboration and Adam, R. and Ade, P. A. R. and Aghanim, N. and Alves, M. I. R. and Arnaud, M. and Ashdown, M. and Aumont, J. and Baccigalupi, C. and Banday, A. J. and et al. Planck 2015 results. X. Diffuse component separation: Foreground maps. *A&A*, 594:A10, September 2016.
- [9] Bennett, C. L. and Hill, R. S. and Hinshaw, G. and Nolta, M. R. and Odegard, N. and Page, L. and Spergel, D. N. and Weiland, J. L. and Wright, E. L. and Halpern, M. and Jarosik, N. and Kogut, A. and Limon, M. and Meyer, S. S. and Tucker, G. S. and Wollack, E. First-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Foreground Emission. *Astro-Ph*, 148:97–117, September 2003.

- [10] D. Maino, A. Farusi, C. Baccigalupi, F. Perrotta, A. J. Banday, L. Bedini, C. Burigana, G. De Zotti, K. M. Górski, and E. Salerno. All-sky astrophysical component separation with Fast Independent Component Analysis (FASTICA). *MNRAS*, 334:53–68, July 2002.
- [11] K. Ichiki. CMB foreground: A concise review. *Progress of Theoretical and Experimental Physics*, 2014(6):06B109, June 2014.
- [12] A. Hyvärinen. Independent Component Analysis. *Neural Computation*, 9:1483–1492, 1997.
- [13] CORE Collaboration and Finelli, F. and Bucher, M. and Achúcarro, A. and Ballardini, M. and Bartolo, N. and Baumann, D. and Clesse, S. and Errard, J. and Handley, W. and Hindmarsh, M. and Kiiveri, K. and Kunz, M. and Lasenby, A. and Liguori, M. and Paoletti, D. and Ringeval, C. and Väiliviita, J. and van Tent, B. and Vennin, V. and Arroja, F. and Ashdown, M. and Banday, A. J. and Banerji, R. and Baselmans, J. and Bartlett, J. G. and de Bernardis, P. and Bersanelli, M. and Bonaldi, A. and Borril, J. and Bouchet, F. R. and Boulanger, F. and Brinckmann, T. and Cai, Z.-Y. and Calvo, M. and Challinor, A. and Chluba, J. and D’Amico, G. and Delabrouille, J. and María Diego, J. and De Zotti, G. and Desjacques, V. and Di Valentino, E. and Feeney, S. and Fergusson, J. R. and Ferraro, S. and Forastieri, F. and Galli, S. and García-Bellido, J. and Génova-Santos, R. T. and Gerbino, M. and González-Nuevo, J. and Grandis, S. and Greenslade, J. and Hagstotz, S. and Hanany, S. and Hazra, D. K. and Hernández-Monteagudo, C. and Hivon, E. and Hu, B. and Kovetz, E. D. and Kurki-Suonio, H. and Lattanzi, M. and Lesgourgues, J. and Lizarraga, J. and López-Caniego, M. and Luzzi, G. and Maffei, B. and Martins, C. J. A. P. and Martínez-González, E. and McCarthy, D. and Matarrese, S. and Melchiorri, A. and Melin, J.-B. and Monfardini, A. and Natoli, P. and Negrello, M. and Oppizzi, F. and Pajer, E. and Patil, S. P. and Piat, M. and Pisano, G. and Poulin, V. and Ravenni, A. and Remazeilles, M. and Renzi, A. and Roest, D. and Salvati, L. and Tartari, A. and Tasinato, G. and Torrado, J. and Trappe, N. and Tucci, M. and Urrestilla, J. and Vielva, P. and Van de Weygaert, R. Exploring Cosmic Origins with CORE: Inflation. *Atro-Ph*, December 2016.
- [14] Delabrouille, J. and Betoule, M. and Melin, J.-B. and Miville-Deschênes, M.-A. and Gonzalez-Nuevo, J. and Le Jeune, M. and Castex, G. and de Zotti, G. and Basak, S. and Ashdown, M. and Aumont, J. and Baccigalupi, C. and Banday, A. J. and Bernard, J.-P. and Bouchet, F. R. and Clements, D. L. and da Silva, A. and Dickinson, C. and Dodu, F. and Dolag, K. and Elsner, F. and Fauvet, L. and Fajÿ, G. and Giardino, G. and Leach, S. and Lesgourgues, J. and Liguori, M. and Macías-Pérez, J. F. and Massardi, M. and Matarrese, S. and Mazzotta, P. and Montier, L. and Mottet, S. and Paladini, R. and Partridge, B. and Piffaretti, R. and Prezeau, G. and Prunet, S. and Ricciardi, S. and Roman, M. and Schaefer, B. and Toffolatti, L. The pre-launch Planck Sky Model: a model of sky emission at submillimetre to centimetre wavelengths. *A&A*, 553:A96, May 2013.
- [15] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann. HEALPix: A Framework for High-Resolution Discretization and Fast Analysis of Data Distributed on the Sphere. *Astro-Ph*, 622:759–771, April 2005.
- [16] Planck Collaboration, P. A. R. Ade, N. Aghanim, C. Armitage-Caplan, M. Arnaud, M. Ashdown, F. Atrio-Barandela, J. Aumont, C. Baccigalupi, A. J. Banday, and et al. Planck 2013 results. XII. Diffuse component separation. *A&A*, 571:A12, November 2014.

Apéndice A

Modificación del algoritmo de FastICA por Newton-Raphson

El método de Newton-Raphson se obtiene del desarrollo en serie de Taylor para una función $f(x)$:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \frac{f'''(x_0)}{3!}(x - x_0)^3 + \dots \quad (\text{A.1})$$

Tomando los dos primeros términos e igualando a cero para hallar las raíces se tiene

$$f(x) = 0 \approx f(x_0) + f'(x_0)(x - x_0). \quad (\text{A.2})$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}. \quad (\text{A.3})$$

El argumento anterior se puede hacer extensible a $N+2$ funciones con $N+2$ incógnitas:

$$\mathbf{F} = \begin{pmatrix} f_1(x_1, x_2, \dots, x_{N+2}) \\ f_2(x_1, x_2, \dots, x_{N+2}) \\ \vdots \\ f_{N+2}(x_1, x_2, \dots, x_{N+2}) \end{pmatrix}. \quad (\text{A.4})$$

y su jacobiano es

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_{N+2}} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_{N+2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_{N+2}}{\partial x_1} & \frac{\partial f_{N+2}}{\partial x_2} & \dots & \frac{\partial f_{N+2}}{\partial x_{N+2}} \end{pmatrix}. \quad (\text{A.5})$$

Nuestra solución es entonces

$$x_{n+1} = x_n - \mathbf{J}^{-1}\mathbf{F}(x_n). \quad (\text{A.6})$$

Partiendo de la función (2.15) y siguiendo el procedimiento descrito anteriormente se tiene que el jacobiano es:

$$\mathbf{J} = \begin{pmatrix} 2 + 2\beta & 0 & 0 & \dots & \sum_{k=1}^N Z_{1k} & 2W_{i1} \\ 0 & 2 + 2\beta & 0 & \dots & \sum_{k=1}^N Z_{2k} & 2W_{i2} \\ 0 & 0 & 2 + 2\beta & \dots & \sum_{k=1}^N Z_{3k} & 2W_{i3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \sum_{k=1}^N Z_{1k} & \sum_{k=1}^N Z_{2k} & \sum_{k=1}^N Z_{3k} & \dots & 0 & 0 \\ 2W_{i1} & 2W_{i2} & 2W_{i3} & \dots & 0 & 0 \end{pmatrix}. \quad (\text{A.7})$$

FastICA halla la matriz \mathbf{W} fila a fila pero no estrictamente en orden, de manera que se ha decidido utilizar el método de Newton-Raphson para estimar en primer lugar la componente del FCM. Tras estimar la primera fila de la matriz \mathbf{W} , el algoritmo de FastICA halla el número restante de filas de la matriz de forma estándar maximizando la no gaussianidad.

Apéndice B

Código para señales 1-D

```
1 # -*- coding: utf-8 -*-
2
3 ## en este código las tres componentes a extraer son no
4 ## gaussianas. la tercera componente es constante en
5 ## todos los canales
6
7 import numpy as np; import matplotlib.pyplot as plt;
8 import scipy.linalg; import scipy.stats; import pylab as pl
9
10 # directorio
11 dire='/Users/david/Documents/TFG/metodos_lineales/newton_raphson/'
12 N=3 # número de componentes
13 Nc=3 # número de canales
14 N_sim=100 # número de simulaciones
15 t=10000 # puntos
16 z=np.arange(t)
17 M=np.matrix(np.zeros((N_sim,28))) # matriz con todos los valores
18                               # de las figuras de mérito
19
20 # varianza, skewness y kurtosis de las señales
21 sigma=np.matrix(np.zeros((N_sim,2*N+1)))
22 delta_sigma=np.matrix(np.zeros((N_sim,N+1)))
23 abs_delta_sigma=np.matrix(np.zeros((N_sim,N+1)))
24 skew=np.matrix(np.zeros((N_sim,2*N+1)))
25 delta_skew=np.matrix(np.zeros((N_sim,N+1)))
26 abs_delta_skew=np.matrix(np.zeros((N_sim,N+1)))
27 kurt=np.matrix(np.zeros((N_sim,2*N+1)))
28 delta_kurt=np.matrix(np.zeros((N_sim,N+1)))
29 abs_delta_kurt=np.matrix(np.zeros((N_sim,N+1)))
30
31 ##
32 ## MATRICES DE
33 ## MEZCLA
34 ##
35 # A=np.matrix('1 2 1; 3 2 1; 2 3 1') # matriz sin dominio
36 A=np.matrix('2 0.6 1; 0.6 2 1; 0.2 0.2 1') # matriz con dominio
37
38 ##
39 ## FRACCIONES
```

```

40 ## RUIDO
41 ##
42 frac=np.matrix('3. 2. 5.')
43 frac_ruido=.0 # si es 0 no hay ruido
44
45 ##
46 ## MEDIDAS DE NO
47 ## GAUSSIANIDAD
48 ##
49 # función
50 def g(u):
51     return u**3
52     # return np.tanh(u)
53     # return np.e**(-u**2)
54 # derivada de la función
55 def g_2(u):
56     return 3*u**2
57     # return (1/np.cosh(u))**2
58     # return -2*u*np.e**(-u**2)
59
60 # Newton-Raphson
61 newton='si'
62
63 numbin=10 # número de bins del histograma de los coef. de corr
64 rancor=[0.999,1] # rango del histograma coeficientes correlación
65
66 # número de iteraciones por simulación
67 itera=np.zeros((100,3))
68
69 ##
70 ## COMIENZO SIMULACIONES
71 ## SEPARACIÓN DE COMPONENTES
72 ##
73 for p in range(N_sim):
74     ##
75     ## GENERACIÓN DE
76     ## LAS SEÑALES
77     ##
78     S=np.zeros((N,t))
79     for i in range(N):
80         S[i]=np.random.normal(0,1,t)
81         S[i]=S[i]+frac[0,i]*np.std(S[i])*S[i]**2
82         S[i]=S[i]-np.mean(S[i])
83         S[i]=S[i]/np.std(S[i])
84         # plt.hold(False)
85         # plt.plot(z,S[i])
86         # plt.title('S$_{'+str(i+1)+'}$',fontsize=12)
87         # pl.savefig(dire+'s'+str(i+1)+'.png',dpi=199)
88         # plt.hist(S[i],30)
89         # pl.savefig(dire+'hist'+str(i+1)+'.png',dpi=199)
90     var=np.zeros((N,1)) # varianza ruido
91     n=np.zeros((Nc,t)) # ruido
92     X=A*S

```

```

93  if frac_ruido!=.0:
94      for i in range(Nc):
95          var[i]=frac_ruido*np.var(X[i])
96      for i in range(Nc):
97          n[i]=np.random.normal(0,np.sqrt(var[i]),t)
98          # plt.plot(z,n[i])
99          # pl.savefig(dire+'/ruido'+str(i+1)+'.png',dpi=199)
100     n=np.matrix(n)
101     X=A*S+n
102     ##
103     ## GRAFICAS
104     ## OBSERVACIONES
105     ##
106     # X=X.A
107     # for i in range(Nc):
108         # plt.plot(z,X[i])
109         # pl.savefig(dire+'/observacion'+str(i+1)+'.png',dpi=199)
110     # X=np.matrix(X)
111     X_ilc=X
112
113     if N!=Nc:
114         ##
115         ## PCA (Principal Component Analysis)
116         ##
117         for i in range(Nc):
118             X_white[i]=X[i]-np.mean(X[i]) # sustracción de la media
119         cov_X_white=np.cov(X_white) # matriz de covarianzas de los datos
120         V_prop=np.linalg.eig(cov_X_white) # valores y vectores propios de la
121                                     # matriz de covarianzas de los datos
122         val_prop=(V_prop)[0]
123         vec_prop=np.matrix((V_prop)[1])
124         arg_val=np.flipud(np.argsort(val_prop)) # posición de los elementos
125                                             # de mayor a menor (numpy.roll()
126                                             # voltea el vector en orden
127                                             # ascendente)
128         val_prop_2=np.zeros((N)) # valores propios en orden descendente
129         V=np.matrix(np.zeros((Nc,N))) # matriz vectores propios en orden
130                                     # descendente
131
132         # sólo nos vamos a quedar con los N valores propios más altos y sus
133         # correspondientes
134         # vectores propios, desechando el resto. Así conseguimos reducir la
135         # dimensión de
136         # nuestros datos al número de componentes N y podemos aplicar FastICA
137         for i in range(N):
138             val_prop_2[i]=val_prop[arg_val[i]]
139             V[:,i]=vec_prop[:,arg_val[i]]
140
141         # datos finales tras PCA. si se calcula la matriz de covarianzas de los
142         # datos
143         # obtenidos después del PCA se tiene una matriz diagonal cuyos elementos
144         # son

```

```

139 # los valores propios correspondientes a los vectores propios elegidos.
      se ha
140 # añadido un blanqueo de los datos (whitening). a los datos de PCA se les
      ha
141 # ha dividido por la raíz cuadrada de una matriz diagonal (diag) cuyos
      elementos
142 # son los valores propios. de esta forma las covarianzas son cero (ya lo
      hacía PCA)
143 # y además ahora las varianzas son cero (paso whitening)
144 # X=V.T*X # paso final PCA
145 diag=np.matrix(np.identity(N)*val_prop_2)
146 X_white=np.linalg.inv(np.matrix(scipy.linalg.sqrtm(diag)))*V.T*X #
      paso final tras whitening
147 X_white=X_white.A
148
149 ##
150 ## ORDENAR FILAS PCA
151 ##
152 X_pca=np.zeros((N,t))
153 for i in range(N):
154     coeficiente=.0
155     for j in range(N):
156         coeficiente_2=(np.corrcoef(X_white[j],S[i]))[0][1]
157         if coeficiente_2>coeficiente:
158             X_pca[i]=X_white[j]
159             coeficiente=coeficiente_2
160         elif np.abs(coeficiente_2)>coeficiente:
161             X_pca[i]=-X_white[j]
162             coeficiente=np.abs(coeficiente_2)
163
164 ##
165 ## PINTAR SEÑALES
166 ## PCA
167 ##
168 # for i in range(N):
169     # plt.plot(z,X_pca[i])
170     # plt.title('PCA$_{' +str(i+1)+'}$',fontsize=12)
171     # pl.savefig(dire+'/pca'+str(i+1)+'.png',dpi=199)
172 X_white=np.matrix(X_white)
173 C_n_m=np.matrix(np.zeros((N,N)))
174 else:
175     ##
176     ## BLANQUEAMIENTO DATOS
177     ##
178     C_n=np.cov(n) # matriz covarianzas de n
179     C_X=np.cov(X) # matriz covarianzas de X
180     Z=np.matrix(scipy.linalg.sqrtm(C_X-C_n)) # raíz cuadrada de la diferencia
      de las
181
      # matrices de covarianzas C_X y C_n
182     C_n_m=Z.I*C_n*Z.I # matriz de covarianzas de n modificada
183     # print 'Matriz blanqueo datos:'
184     # print raiz.I
185     X_white=Z.I*X # datos blanqueados X_white

```

```

186     # print np.cov(X_white) # matriz de covarianzas de los datos blanqueados
        X_white
187                                     # que ha de ser igual a la matriz identidad I
188
189     ##
190     ## ALGORITMO FASTICA
191     ##
192     W=np.matrix(np.zeros((N,N))) # matriz mezcla a determinar
193     for i in range(N):
194         ite=0
195         for j in range(15):
196             if j==0:
197                 w0=np.random.rand(N,1)           # w0 vector columna inicial
198                 w0=np.matrix(w0/np.linalg.norm(w0)) # y aleatorio de norma unidad
199                 # cálculo de los valores de espectación
200                 E=(X_white*(np.matrix(g((w0.T*X_white).A)).T))/float(t)
201                 E_2=np.sum(np.matrix(g_2((w0.T*X_white).A)))/float(t)
202                 I=np.matrix(np.identity(N))
203                 w=E-(I+C_n_m)*w0*E_2 # nuevo vector columna w
204                 ##
205                 ## GRAM-SCHMIDT
206                 ## (ORTOGONALIZACIÓN)
207                 ##
208                 suma=np.matrix(np.zeros((N,1)))
209                 if i!=0:
210                     for m in range(i):
211                         ar=(W[m]).T*((W.T*(W[m])).T)/(np.linalg.norm((W[m]).T)**2)
212                         suma=suma+ar
213                 w=w-suma
214                 w=np.matrix(w.A/np.linalg.norm(w.A))
215                 ##
216                 ## NEWTON-RAPHSON
217                 ##
218                 if newton=='si' and i==0:
219                     epsilon=1e-2 # valor de tolerancia
220                     # solución inicial (input del método)
221                     newton_sol=np.matrix(np.random.rand(N+2,1))
222                     w_new=newton_sol[0:N]
223                     Lambda=newton_sol[N:N+2]
224                     # este bucle itera el algoritmo de Newton tantas veces
225                     # como se indique
226                     for k in range(100):
227                         funcion=np.matrix(np.zeros((N+2,1))) # vector columna que va a
                            # contener
228                                     # los valores de las funciones
229                                     # evaluadas
230                         J=np.matrix(np.zeros((N+2,N+2))) # jacobiano
231                         # el siguiente bucle for evalúa las funciones y el jacobiano
232                         # con los valores del vector w_new
233                         for l in range(N):
234                             funcion[l]=-2*(w[l]-w_new[l])+Lambda[0]*np.sum((Z.I)[: ,l])
235                             +2*Lambda[1]*w_new[l] # funciones 1-N

```

```

236         funcion[N]=funcion[N]+w_new[1]*np.sum((Z.I)[: ,1]) # funcion
           N+1
237         funcion[N+1]=funcion[N+1]+w_new[1]**2 # funcion N+2
238         J[1,1]=2+2*Lambda[1]
239         J[1,N]=np.sum((Z.I)[: ,1])
240         J[N,1]=J[1,N]
241         J[1,N+1]=2*w_new[1]
242         J[N+1,1]=J[1,N+1]
243         funcion[N]=funcion[N]-1
244         funcion[N+1]=funcion[N+1]-1
245         # nueva solución
246         newton_sol2=newton_sol-J.I*funcion
247         # condición de convergencia Newton-Raphson
248         if np.linalg.norm(newton_sol2[0:N]-newton_sol[0:N])<epsilon:
249             break
250         distancia=np.linalg.norm(newton_sol2[0:N]-newton_sol[0:N])
251         w_new=newton_sol2[0:N]
252         Lambda=newton_sol2[N:N+2]
253         newton_sol=newton_sol2
254         # fin Newton-Raphson
255         print 'Fin Newton-Raphson:',k,'iteraciones'
256         # print 'Distancia:',distancia
257         w=w_new
258         W[0]=w.T
259         break
260     ##
261     ## CONDICIÓN DE
262     ## CONVERGENCIA
263     ## FASTICA
264     ##
265     if np.abs(w.T*w0)>0.9999:
266         break
267     else:
268         ite=ite+1
269         w0=w
270     if j>0:
271         print 'Iteraciones W_'+str(i+1)+'':',ite
272         itera[p,i]=ite
273     W[i]=w.T
274     Y=W*X_white # componentes recuperadas por FastICA
275
276     ##
277     ## ORDENAR FILAS
278     ##
279     # se ordenan las filas de tal modo que la componente
280     # i-ésima de Y corresponda a la componente i-ésima de S.
281     # Así se pueden calcular los coeficientes correlación de
282     # las componentes extraídas por FastICA con las originales
283     Y_2=np.matrix(np.zeros((N,t)))
284     for i in range(N):
285         coeficiente=.0
286         for j in range(N):
287             coeficiente_2=(np.corrcoef(Y[j],S[i]))[0][1]

```

```

288     if coeficiente_2>coeficiente:
289         Y_2[i]=Y[j]
290         M[p,i]=coeficiente_2 # coeficientes de correlación rho (FastICA)
291         coeficiente=coeficiente_2
292     elif np.abs(coeficiente_2)>coeficiente:
293         Y_2[i]=-Y[j]
294         M[p,i]=np.abs(coeficiente_2)
295         coeficiente=np.abs(coeficiente_2)
296 W_2=Y_2*X_white.I
297
298 ##
299 ## ORDEN FILAS
300 ##
301 orden=np.zeros((N))
302 for i in range(N):
303     for j in range(N):
304         if np.abs((np.corrcoef(W_2[i],W[j]))[0][1])>0.9:
305             orden[i]=j
306         else:
307             pass
308
309 ##
310 ## RECUPERACIÓN NORMA
311 ## SEÑALES OBTENIDAS
312 ## POR FASTICA
313 ##
314 # se recupera la norma de la señal constante
315 # (denominada alpha), y a partir de ella se
316 # recupera la norma del resto de señales
317 alpha=float(S[N-1]*(Y_2[N-1]).T/(Y_2[N-1]*(Y_2[N-1]).T))
318
319 ##
320 ## ALGORITMO ILC:
321 ##
322 C=np.matrix(np.ones((Nc+1,Nc+1)))
323 C[Nc,Nc]=.0
324 for i in range(Nc):
325     X_ilc[i]=X_ilc[i]-np.mean(X_ilc[i]) # hacemos 0 la media de los datos
326     for j in range(i,Nc):
327         X_ilc[j]=X_ilc[j]-np.mean(X_ilc[j])
328         C[i,j]=2*X_ilc[i]*X_ilc[j].T/float(t)
329         C[j,i]=C[i,j]
330
331 C_2=C.I
332 # print 'Matriz inversa de C:'
333 # print C_2
334 x=C_2[:,Nc] # vector pesos y lambda
335 # print x
336 # np.savetxt(dire+'/pesos_'+str(Nc)+'canales.txt',x)
337 # print 'Suma pesos = ',np.sum(x[0:N]) # suma de los pesos, que ha de ser
    igual a la unidad
338 S_ilc=.0
339 for i in range(Nc):

```

```

340     S_ilc=S_ilc+x[i]*X_ilc[i]
341
342     ##
343     ## GRÁFICAS SEÑALES RECUPERADAS
344     ## POR FASTICA E ILC
345     ##
346     Y_2=Y_2.A
347     # for i in range(N):
348         # plt.plot(z,Y_2[i])
349         # plt.title('FastICA$_{'+str(i+1)+'}$',fontsize=12)
350         # pl.savefig(dire+'/sol_fastICA'+str(i+1)+'.png',dpi=199)
351     S_ilc=S_ilc.A
352     # plt.plot(z,S_ilc[0])
353     # plt.title('ILC',fontsize=12)
354     # pl.savefig(dire+'/sol_ILC.png',dpi=199)
355
356     ##
357     ## VARIANZA Y MOMENTOS
358     ## DE ORDEN 3 Y 4
359     ## (SKEWNESS Y KURTOSIS)
360     ## SEÑALES RECUPERADAS
361     ##
362     for i in range(N):
363         sigma[p,i]=np.var(S[i])
364         sigma[p,N+i]=np.var(Y_2[i])
365         # print 'Varianza señal',str(i+1),'=',sigma[p,i]
366         # print 'Varianza FastICA',str(i+1),'=',sigma[p,N+i]
367         delta_sigma[p,i]=sigma[p,N+i]-sigma[p,i]
368         abs_delta_sigma[p,i]=np.abs(sigma[p,N+i]-sigma[p,i])
369     sigma[p,2*N]=np.var(S_ilc[0])
370     # print 'Varianza ILC =',sigma[p,2*N]
371     delta_sigma[p,N]=sigma[p,2*N]-sigma[p,2]
372     abs_delta_sigma[p,N]=np.abs(sigma[p,2*N]-sigma[p,2])
373     for i in range(N):
374         skew[p,i]=scipy.stats.skew(S[i])
375         skew[p,N+i]=scipy.stats.skew(Y_2[i])
376         # print 'Skewness señal',str(i+1),'=',skew[p,i]
377         # print 'Skewness FastICA',str(i+1),'=',skew[p,N+i]
378         delta_skew[p,i]=skew[p,N+i]-skew[p,i]
379         abs_delta_skew[p,i]=np.abs(skew[p,N+i]-skew[p,i])
380     skew[p,2*N]=scipy.stats.skew(S_ilc[0])
381     # print 'Skewness ILC =',skew[p,2*N]
382     delta_skew[p,N]=skew[p,2*N]-skew[p,2]
383     abs_delta_skew[p,N]=np.abs(skew[p,2*N]-skew[p,2])
384     for i in range(N):
385         kurt[p,i]=scipy.stats.kurtosis(S[i])
386         kurt[p,N+i]=scipy.stats.kurtosis(Y_2[i])
387         # print 'Kurtosis señal',str(i+1),'=',kurt[p,i]
388         # print 'Kurtosis FastICA',str(i+1),'=',kurt[p,N+i]
389         delta_kurt[p,i]=kurt[p,N+i]-kurt[p,i]
390         abs_delta_kurt[p,i]=np.abs(kurt[p,N+i]-kurt[p,i])
391     kurt[p,2*N]=scipy.stats.kurtosis(S_ilc[0])
392     # print 'Kurtosis ILC =',kurt[p,2*N]

```

```

393 delta_kurt[p,N]=kurt[p,2*N]-kurt[p,2]
394 abs_delta_kurt[p,N]=np.abs(kurt[p,2*N]-kurt[p,2])
395
396 ##
397 ## FIGURAS DE MÉRITO
398 ##
399 for i in range(N):
400     print 'Rho',str(i+1),'=' ,M[p,i]
401 M[p,3]=np.abs((np.corrcoef(S_ilc[0],S[2]))[0][1]) # coeficiente de
    correlación rho (ILC)
402 print 'Rho ILC =' ,M[p,3]
403 # if M[p,0]<0.9 or M[p,1]<0.9 or M[p,2]<0.9:
404     # break
405
406 q=4
407 for i in range(N):
408     D=(Y_2[i]-S[i])/np.std(S[i])
409     # plt.plot(z,D[0])
410     # plt.ylabel('D'+str(i+1),fontsize=12)
411     # pl.savefig(dire+'D'+str(i+1)+'.png',dpi=199)
412     M[p,q]=np.mean(D) # <D>
413     M[p,q+1]=np.mean(np.abs(D)) # <|D|>
414     M[p,q+2]=np.std(D) # std(D)
415     q=q+3
416 D_ILC=(S_ilc[0]-S[2])/np.std(S[2])
417 # plt.plot(z,D_ILC[0])
418 # plt.ylabel('D$_{ILC}$',fontsize=12)
419 # pl.savefig(dire+'D_ILC.png',dpi=199)
420 M[p,13]=np.mean(D_ILC) # <D_ILC>
421 M[p,14]=np.mean(np.abs(D_ILC)) # <|D_ILC|>
422 M[p,15]=np.std(D_ILC) # std(D_ILC)
423 q=16
424 for i in range(N):
425     M[p,q]=np.mean(2*(Y_2[i]-S[i])/(Y_2[i]+S[i])) # <D*>
426     M[p,q+1]=np.mean(np.abs(2*(Y_2[i]-S[i])/(Y_2[i]+S[i]))) # <|D*|>
427     M[p,q+2]=np.std(M[p,q]) # std*(D*)
428     q=q+3
429 M[p,25]=np.mean(2*(S[2]-S_ilc[0])/(S[2]+S_ilc[0])) # <D*_ILC>
430 M[p,26]=np.mean(np.abs(2*(S[2]-S_ilc[0])/(S[2]+S_ilc[0]))) # <|D*_ILC|>
431 M[p,27]=np.std(M[p,25]) # std*(D*_ILC)
432
433 print 'Simulación ',str(p+1),' realizada'
434
435 ##
436 ## FICHEROS
437 ## FINALES A
438 ## GUARDAR
439 ##
440 np.savetxt(dire+'orden.txt',orden)
441
442 # varianza
443 np.savetxt(dire+'varianza.txt',sigma)
444 plt.hold(False)

```

```

445 for i in range(N):
446     plt.scatter(sigma[:,i],sigma[:,N+i])
447     plt.ylabel(r'\sigma^2_{'+str(i+1)+'}', fontsize=12)
448     plt.xlabel(r'\sigma^2_{0,'+str(i+1)+'}', fontsize=12)
449     pl.savefig(dire+'delta_varianza'+str(i+1)+'.png',dpi=199)
450 plt.scatter(sigma[:,N-1],sigma[:,2*N-1])
451 plt.ylabel(r'\sigma^2_{ILC}', fontsize=10)
452 plt.xlabel(r'\sigma^2_{0,ILC}', fontsize=12)
453 pl.savefig(dire+'delta_varianza_ilc.png',dpi=199)
454 np.savetxt(dire+'delta_varianza.txt',delta_sigma)
455 np.savetxt(dire+'abs_delta_varianza.txt',abs_delta_sigma)
456 delta_sigma_mean=np.matrix(np.zeros((1,N+1)))
457 delta_sigma_std=np.matrix(np.zeros((1,N+1)))
458 abs_delta_sigma_mean=np.matrix(np.zeros((1,N+1)))
459 abs_delta_sigma_std=np.matrix(np.zeros((1,N+1)))
460 for i in range(N+1):
461     delta_sigma_mean[0,i]=np.mean(delta_sigma[:,i])
462     delta_sigma_std[0,i]=np.std(delta_sigma[:,i])
463     abs_delta_sigma_mean[0,i]=np.mean(abs_delta_sigma[:,i])
464     abs_delta_sigma_std[0,i]=np.std(abs_delta_sigma[:,i])
465 np.savetxt(dire+'delta_varianza_mean',delta_sigma_mean,fmt='%.6e',header='Media
delta varianza:') # archivo media delta varianza
466 np.savetxt(dire+'delta_varianza_std',delta_sigma_std,fmt='%.6e',header='Desviación
estándar delta varianza:') # archivo desviación estándar delta varianza
467 np.savetxt(dire+'abs_delta_varianza_mean',abs_delta_sigma_mean,fmt='%.6e',header='Media
delta abs varianza:')
468 np.savetxt(dire+'abs_delta_varianza_std',abs_delta_sigma_std,fmt='%.6e',header='Desviación
estándar delta abs varianza:')
469
470 # skewness
471 np.savetxt(dire+'skewness.txt',skew)
472 for i in range(N):
473     plt.scatter(skew[:,i],skew[:,N+i])
474     plt.ylabel(r'\$S^'+str(i+1)+'}', fontsize=12)
475     plt.xlabel(r'\$S^'+str(i+1)+'}_{0}', fontsize=12)
476     pl.savefig(dire+'delta_skew'+str(i+1)+'.png',dpi=199)
477 plt.scatter(skew[:,N-1],skew[:,2*N-1])
478 plt.ylabel(r'\$S^{\ILC}', fontsize=12)
479 plt.xlabel(r'\$S^{\ILC}_{0}', fontsize=12)
480 pl.savefig(dire+'delta_skew_ilc.png',dpi=199)
481 np.savetxt(dire+'delta_skewness.txt',delta_skew)
482 np.savetxt(dire+'abs_delta_skewness.txt',abs_delta_skew)
483 delta_skew_mean=np.matrix(np.zeros((1,N+1)))
484 delta_skew_std=np.matrix(np.zeros((1,N+1)))
485 abs_delta_skew_mean=np.matrix(np.zeros((1,N+1)))
486 abs_delta_skew_std=np.matrix(np.zeros((1,N+1)))
487 for i in range(N+1):
488     delta_skew_mean[0,i]=np.mean(delta_skew[:,i])
489     delta_skew_std[0,i]=np.std(delta_skew[:,i])
490     abs_delta_skew_mean[0,i]=np.mean(abs_delta_skew[:,i])
491     abs_delta_skew_std[0,i]=np.std(abs_delta_skew[:,i])
492 np.savetxt(dire+'delta_skewness_mean',delta_skew_mean,fmt='%.6e',header='Media
delta skewness:') # archivo media delta skewness

```

```

493 np.savetxt(dire+'delta_skewness_std',delta_skew_std,fmt='%.6e',header='Desviación
    estándar delta skewness:') # archivo desviación estándar delta skewness
494 np.savetxt(dire+'abs_delta_skewness_mean',abs_delta_skew_mean,fmt='%.6e',header='Media
    delta abs skewness:')
495 np.savetxt(dire+'abs_delta_skewness_std',abs_delta_skew_std,fmt='%.6e',header='Desviación
    estándar delta abs skewness:')
496
497 # kurtosis
498 np.savetxt(dire+'kurtosis.txt',kurt)
499 for i in range(N):
500     plt.scatter(kurt[:,i],kurt[:,N+i])
501     plt.ylabel(r'$K^{'+str(i+1)+'}$', fontsize=12)
502     plt.xlabel(r'$K^{'+str(i+1)+'}_{0}$', fontsize=12)
503     pl.savefig(dire+'delta_kurt'+str(i+1)+'.png',dpi=199)
504 plt.scatter(kurt[:,N-1],kurt[:,2*N-1])
505 plt.ylabel(r'$K^{\{ILC\}}$', fontsize=12)
506 plt.xlabel(r'$K^{\{ILC\}}_{0}$', fontsize=12)
507 pl.savefig(dire+'delta_kurt_ilc.png',dpi=199)
508 np.savetxt(dire+'delta_kurtosis.txt',delta_kurt)
509 np.savetxt(dire+'abs_delta_kurtosis.txt',abs_delta_kurt)
510 delta_kurt_mean=np.matrix(np.zeros((1,N+1)))
511 delta_kurt_std=np.matrix(np.zeros((1,N+1)))
512 abs_delta_kurt_mean=np.matrix(np.zeros((1,N+1)))
513 abs_delta_kurt_std=np.matrix(np.zeros((1,N+1)))
514 for i in range(N+1):
515     delta_kurt_mean[0,i]=np.mean(delta_kurt[:,i])
516     delta_kurt_std[0,i]=np.std(delta_kurt[:,i])
517     abs_delta_kurt_mean[0,i]=np.mean(abs_delta_kurt[:,i])
518     abs_delta_kurt_std[0,i]=np.std(abs_delta_kurt[:,i])
519 np.savetxt(dire+'delta_kurtosis_mean',delta_kurt_mean,fmt='%.6e',header='Media
    delta kurtosis:') # archivo media delta kurtosis
520 np.savetxt(dire+'delta_kurtosis_std',delta_kurt_std,fmt='%.6e',header='Desviación
    estándar delta kurtosis:') # archivo desviación estándar delta kurtosis
521 np.savetxt(dire+'abs_delta_kurtosis_mean',abs_delta_kurt_mean,fmt='%.6e',header='Media
    delta abs kurtosis:')
522 np.savetxt(dire+'abs_delta_kurtosis_std',abs_delta_kurt_std,fmt='%.6e',header='Desviación
    estándar delta abs kurtosis:')
523 plt.close()
524
525 np.savetxt(dire+'coefcorr.txt',M[:, :4:],fmt='%.6e',header='Coeficientes de
    correlación rho:') # archivo coeficientes de correlación rho
526 rho_mean=np.matrix(np.zeros((1,N+1)))
527 rho_std=np.matrix(np.zeros((1,N+1)))
528 for i in range(N+1):
529     rho_mean[0,i]=np.mean(M[:,i])
530     rho_std[0,i]=np.std(M[:,i])
531 np.savetxt(dire+'rho_mean',rho_mean,fmt='%.6e',header='Media de los coeficientes
    de correlación rho:') # archivo media coeficientes rho
532 np.savetxt(dire+'rho_std',rho_std,fmt='%.6e',header='Desviación estándar de los
    coeficientes de correlación rho:') # archivo desviación estándar coeficientes
    rho
533
534 plt.hold(True)

```

```

535 plt.hist((M[:,2],M[:,3]),bins=numbin,range=rancor) # histograma de los
      coeficientes de correlación señal 3
536 # plt.title(u'Histograma coeficientes de correlación')
537 # plt.ylabel('Frecuencia', fontsize=14)
538 plt.xlabel(r'$\rho$', fontsize=12)
539 plt.legend((r'$\rho_{FastICA}$',r'$\rho_{ILC}$'),fontsize=12,loc='upper left')
540 pl.savefig(dire+'histo_rho3.png',dpi=199)
541 plt.close()
542
543 np.savetxt(dire+'D_datos.txt',M[:,4:16],fmt='%.4e',header='<D_1> <|D_1|>
      std(D_1) <D_2> <|D_2|> std(D_2) <D_3> <|D_3|> std(D_3)') # archivo D's
544 np.savetxt(dire+'D*_datos.txt',M[:,16:28],fmt='%.4e',header='<D*_1> <|D*_1|>
      std*(D*_1) <D*_2> <|D*_2|> std*(D*_2) <D*_3> <|D*_3|> std(D*_3)') # archivo
      D*'s
545
546 plt.hist((M[:,10],M[:,13]),10) # histograma D de la señal 3
547 # plt.title(u'Histograma <D$_{3}$>')
548 # plt.ylabel('Frecuencia')
549 plt.xlabel('<D$_{3}$>',fontsize=12)
550 plt.legend(('<D$_{3}$> FastICA', '<D$_{3}$> ILC'),fontsize=12,loc='upper left')
551 pl.savefig(dire+'histo_D3.png',dpi=199)
552 plt.close()
553
554 np.savetxt(dire+'M_datos.txt',M,fmt='%.6e',header='Matriz de datos M:') # matriz
      de datos M

```

Apéndice C

Código para 2-D: CORE y Planck

```
1 # -*- coding: utf-8 -*-
2
3 ##
4 ## en este código de aplica FastICA e ILC
5 ## a los mapas del cielo de microondas de Planck
6 ##
7
8 import numpy as np; import healpy as hp
9 import scipy.linalg; import pylab as pl
10
11 # directorios
12 directorio='/Users/david/Documents/TFG/metodos_lineales/aplicacion_datos/
13 aplicacion_planck/3comp/'
14 dire='/Users/david/Documents/TFG/mapas/mapas_planck/mapas_planck_1024_fwhm32p29/'
15
16 N=3 # número de componentes
17 Nc=9 # número de canales
18 nside=1024
19 npix=12*nside**2 # número de pixels
20
21 # frecuencias
22 f=['30','44','70','100','143','217','353','545','857']
23
24 # función
25 def g(u):
26     return u**3
27 # derivada de la función
28 def g_2(u):
29     return 3*u**2
30
31 ##
32 ## DATOS
33 ##
34 # mapas del cielo de microondas
35 X=np.matrix(np.zeros((Nc,npix)))
36 for i in range(Nc):
37     X[i]=hp.read_map(dire+'planck_'+f[i]+'GHz_1024_fwhm32p29.fits')
38
39 X_ilc=X
```

```

40
41 ##
42 ## PCA (Principal Component Analysis)
43 ##
44 for i in range(Nc):
45     X[i]=X[i]-np.mean(X[i]) # sustracción de la media
46 cov_X=np.cov(X) # matriz de covarianzas de los datos
47 V_prop=np.linalg.eig(cov_X) # valores y vectores propios de la
48                             # matriz de covarianzas de los datos
49 val_prop=(V_prop)[0]
50 vec_prop=np.matrix((V_prop)[1])
51 arg_val=np.flipud(np.argsort(val_prop)) # posición de los elementos
52                                         # de mayor a menor (numpy.roll())
53                                         # voltea el vector en orden ascendente)
54 val_prop_2=np.zeros((N)) # valores propios en orden descendente
55 V=np.matrix(np.zeros((Nc,N))) # matriz vectores propios en orden descendente
56
57 # sólo nos vamos a quedar con los N valores propios más altos y sus
58     correspondientes
59 # vectores propios, desechando el resto. Así conseguimos reducir la dimensión de
60 # nuestros datos al número de componentes N y podemos aplicar FastICA
61 for i in range(N):
62     val_prop_2[i]=val_prop[arg_val[i]]
63     V[:,i]=vec_prop[:,arg_val[i]]
64
65 # datos finales tras PCA. si se calcula la matriz de covarianzas de los datos
66 # obtenidos después del PCA se tiene una matriz diagonal cuyos elementos son
67 # los valores propios correspondientes a los vectores propios elegidos. se ha
68 # añadido un blanqueo de los datos (whitening). a los datos de PCA se les ha
69 # ha dividido por la raíz cuadrada de una matriz diagonal (Lambda) cuyos elementos
70 # son los valores propios. de esta forma las covarianzas son cero (ya lo hacía
71     PCA)
72 # y además ahora las varianzas son cero (paso whitening)
73 # X=V.T*X # paso final PCA
74 gamma=np.matrix(np.identity(N)*val_prop_2)
75 X_white=np.linalg.inv(np.matrix(scipy.linalg.sqrtm(gamma)))*V.T*X # paso final
76     tras whitening
77 C_n_m=np.zeros((N,N))
78
79 ##
80 ## ALGORITMO FASTICA
81 ##
82 W=np.matrix(np.zeros((N,N))) # matriz mezcla a determinar
83 for i in range(N):
84     for j in range(15):
85         if j==0:
86             w0=np.random.rand(N,1) # w0 vector columna inicial
87             w0=np.matrix(w0/np.linalg.norm(w0)) # y aleatorio de norma unidad
88             # cálculo de los valores de espectación
89             E=(X_white*(np.matrix(g((w0.T*X_white).A)).T))/float(npix)
90             E_2=np.sum(np.matrix(g_2((w0.T*X_white).A)))/float(npix)
91             I=np.matrix(np.identity(N))
92             w=E-(I+C_n_m)*w0*E_2 # nuevo vector columna w

```

```

90     ##
91     ## GRAM-SCHMIDT
92     ## (ORTOGONALIZACIÓN)
93     ##
94     suma=np.matrix(np.zeros((N,1)))
95     if i!=0:
96         for m in range(i):
97             ar=(W[m]).T*(w.T*(W[m]).T)/(np.linalg.norm((W[m]).T)**2)
98             suma=suma+ar
99             w=w-suma
100    w=np.matrix(w.A/np.linalg.norm(w.A))
101    ##
102    ## CONDICIÓN DE
103    ## CONVERGENCIA
104    ## FASTICA
105    ##
106    if np.abs(w.T*w0)>0.9999:
107        break
108    else:
109        w0=w
110    W[i]=w.T
111    Y=(W*X_white).A
112
113    ###
114    ### MAPAS OBTENIDOS
115    ### POR FASTICA
116    ###
117    for i in range(N):
118        hp.mollview(Y[i])
119
120    ##
121    ## ALGORITMO ILC:
122    ##
123    C=np.matrix(np.ones((Nc+1,Nc+1)))
124    C[Nc,Nc]=.0
125    for i in range(Nc):
126        X_ilc[i]=X_ilc[i]-np.mean(X_ilc[i]) # hacemos 0 la media de los datos
127        for j in range(i,Nc):
128            X_ilc[j]=X_ilc[j]-np.mean(X_ilc[j])
129            C[i,j]=2*X_ilc[i]*X_ilc[j].T/float(npix)
130            C[j,i]=C[i,j]
131
132    C_2=np.linalg.inv(C)
133    x=C_2[:,Nc] # vector pesos y lambda
134    # np.savetxt(dire+'/pesos_'+str(Nc)+'canales.txt',x)
135    # print 'Suma pesos = ',np.sum(x[0:N]) # suma de los pesos, que ha de ser igual a
    la unidad
136    S_ilc=.0
137    for i in range(Nc):
138        S_ilc=S_ilc+x[i]*X_ilc[i]
139    S_ilc=S_ilc.A
140    hp.mollview((S_ilc[0]-np.mean(S_ilc[0]))/np.std(S_ilc[0]),min=-3,max=3)
141    pl.savefig(directorio+'cmb_ilc.png',dpi=199)

```

```
142 hp.write_map(directorio+'cmb_ilc.fits',S_ilc[0])
```
