Trabajo Computacional 3
Detección de comunidades en redes

CICCHINI Tomás, SZISCHIK Candela, VIDAL María Sol

Resumen

En este trabajo se estudiarán distintos métodos de detección de comunidades en una red de interacción entre delfines. Se evaluarán las estructuras dadas en cada método mediante la medida de modularidad y de *silhouette*. Se medirán observables como la precisión e información mutua para caracterizar el acuerdo de particiones. Se analizará la relación entre la estructura de las comunidades en la red y la distribución de géneros de los delfines.

Introducción

En este trabajo se analizarán distintos métodos para detectar comunidades en una red de interacción de delfines. La idea principal de hallar comunidades es buscar similaridad dentro de las mismas. En otras palabras, se buscan vértices de una red que sean semejantes según un dado criterio.

Al analizar la existencia de comunidades en los grafos se trabaja con la hipótesis de la existencia de comunidades en la propia red, es decir que la información debería estar, de alguna forma, en la matriz de adyacencia y que no sea dado únicamente por la heurística. Lo que soporta esta idea de comunidades es un grado medio de inhomogeneidad presente en la red, suficiente para alegarse de la aleatoriedad- al haber desorden veo estadísticamente entornos similares.

La noción de comunidad en el lenguaje de redes es la de una partición de los N vértices de la red en M grupos tales que nodos de la misma comunidad sean *más parecidos* entre sí que respecto a los nodos fuera de dicho grupo. Esto último implica que hay que definir dos cosas: primero un criterio de similaridad entre vértices, es decir, a qué llamo nodos parecidos y en segundo lugar una heurística para calcular comunidades a partir del criterio elegido.

La heurística puede ser, por ejemplo, por agrupamiento jerárquico. En esta se construye una matriz de similaridad de los nodos a partir de la matriz de adyacencia. Luego, iterativamente se identifican grupos de vértices de alta similaridad siguiendo alguna estrategia, ya sea aglomerativa o divisiva.

La estrategia aglomerativa consiste en partir de comunidades compuestas por un solo nodo y adosar sucesivamente nodos y comunidades de alta similaridad. En estrategia divisiva, en cambio, se empieza con todo el grafo como una sola comunidad y luego se dividen sucesivamente comunidades, removiendo enlaces que conectan nodos de baja similaridad. En este trabajo se consideran cuatro algorítmos de detección: Edge-Betweenness, Fast Greedy, Infomap y Louvain.

Evaluando particiones

Para evaluar particiones se pueden usar medidas con información externa o en ausencia de información externa (medidas internas). Dentro de estas últimas se encuentran la modularidad y *silhouette*.

La medida de silhouette evalúa la compacidad y la separación de los grupos hallados en la partición, entendiendo a las comunidades como subgrafos conexos y densos. La definición de *silhouette* para cada nodo es la siguiente:

$$s(i) = \frac{b(i) - a(i)}{max\left\{a(i), b(i)\right\}} \tag{1}$$

Donde $a(i)$ es la distancia media del nodo i con el resto de su *cluster* y $b(i)$ es el mínimo de las distancias medias del nodo i a los otros grupos.

Lo que se espera es que la distancia de un nodo i a otros clusters sea mayor que $a(i)$, por lo que espero un valor de *silhouette* positivo y grande. Los valores negativos de esta medida implican que la distancia de un nodo a otro *cluster* es menor a la distancia promedio a los del mismo grupo, por lo que ese nodo no pareciera formar parte del grupo al que fue asignado.

La modularidad es una medida del grado en que las componentes de un sistema pueden separarse y recombinarse. Asimismo es una medida del alcance en la que lo parecido se conecta con lo parecido en una red. La modularidad viene dada por:

$$Q = \sum_r \left(\frac{L_r}{L} - \left(\frac{k_r}{2L} \right)^2 \right) \tag{2}$$

Donde L es el número de enlaces, $\frac{L_r}{L}$ es la fracción de enlaces entre nodos del *cluster* r y $\left(\frac{k_r}{2L}\right)^2$ es la fracción de enlaces adyacentes a nodos de *cluster* r.

La modularidad es estrictamente menor a uno y toma valores positivos si hay más enlaces entre nodos del mismo cluster de los que habría de esperar por azar, y valores negativos si hay menos.

Valores altos de modularidad implican mejor valor de asortatividad entre pertenencia a grupos y cableado de la red. Una partición de un único *cluster* tendrá Q=0 (los dos términos son idénticos). Si, en cambio, cada nodo pertenece a una comunidad distinta $L_r = 0$ y entonces $Q < 0$.

Por otro lado, tenemos las medidas con información externa para comparar particiones. En este trabajo se utilizaran las medidas externas de precisión e información mutua.

En la medida de precisión se comparan dos particiones 1 y 2. Para compararlas se considera todos los posibles pares de nodos y evaluamos si éstos residen o no en la misma comunidad en ambas particiones. Para eso, se calcula el número total de pares de nodos que están en el mismo cluster tanto en la partición A como en la B. A ese número lo llamo a. Y también se cuenta el número total de pares de nodos que no están en el mismo cluster en ambas particiones (d). Teniendo esto en cuenta se define la precisión entre dos particiones 1 y 2 como:

$$P = \frac{a + d}{n(n-1)/2}$$

Donde n es el número de nodos de la red, por lo que el denominador es igual al número total de

pares de nodos. Esto último hace que los valores de precisión vayan entre cero y uno.

Por otro lado, otra medida externa para comparar el acuerdo entre particiones es la información mutua. Esta cuantifica el hecho de que si dos particiones son similares, obtengo mucha información de una de ellas si conozco la otra. Dadas dos particiones, se define la información mutua entre ellas (sin normalizar) como:

$$I\left(C_1, C_2\right) = \sum_{C_1} \sum_{C_2} \left(p\left(C_1, C_2\right) log \frac{p\left(C_1, C_2\right)}{p\left(C_1\right) p\left(C_2\right)} \right)$$

Donde $p\left(C_1, C_2\right)$ es la probabilidad conjunta de que un nodo elegido al azar pertenezca a la comunidad C_1 de la primera partición y a la C_2 de la segunda partición; $p\left(C_1\right)$ es la probabilidad de que un nodo elegido al azar pertenezca a la comunidad C_1 de una dada partición, ídem para $p\left(C_2\right)$. Notar que $p\left(C_1\right) p\left(C_2\right)$ es la probabilidad conjunta si asumimos independencia entre particiones. Por último, la versión normalizada de $I\left(C_1, C_2\right)$ es:

$$I_N = \frac{I\left(C_1, C_2\right)}{\left(H\left(C_1\right) + H\left(C_2\right)\right)/2}$$

Donde $H\left(C_1\right)$ es la entropia de Shannon de una de las particiones y es igual a calcular la información mutua (no normalizada) para una partición consigo misma, es decir, $H\left(C_1\right) = I\left(C_1, C_1\right)$. Por último, los valores de I_N van entre cero y uno.

Clusters a la Newman Girvan: *Edge Betweenness*

Un método para detectar comunidades en una red es el de *Edge Betweenness*. La idea de este algoritmo es partir de un *cluster* gigante e ir dividiéndolo a partir de remover enlaces que conecten nodos de baja similaridad, por lo que este algoritmo es divisivo.

Para ello lo que se debe hacer primero es definir una centralidad de enlaces, que será la intermediatez de los mismos (*link betweenness*). Esta medida de centralidad es proporcional al número de caminos más cortos entre todos los pares de nodos que atraviesen un dado enlace. En segundo lugar se realiza un agrupamiento jerárquico divisivo: se computa la centralidad de enlaces, luego se remueve el enlace de mayor centralidad. Se recalcula la centralidad de los enlaces y se repite hasta descartar el último enlace.

La pregunta es entonces cómo cuantificar el acuerdo entre cableado y partición en grupos. Si corto el método de forma tal de obtener pocos *clusters* se tendrán pocos enlaces internos en comparación con los externos. Si corto en muchos grupos voy a tener muchos enlaces externos. Para encontrar un balance se utiliza la medida de modularidad Q: se elige la partición que tenga mayor modularidad.

A continuación se muestra la red de interacción de delfines, con las comunidades arrojadas por el método *Edge-Betweenness* marcadas en distintos colores.

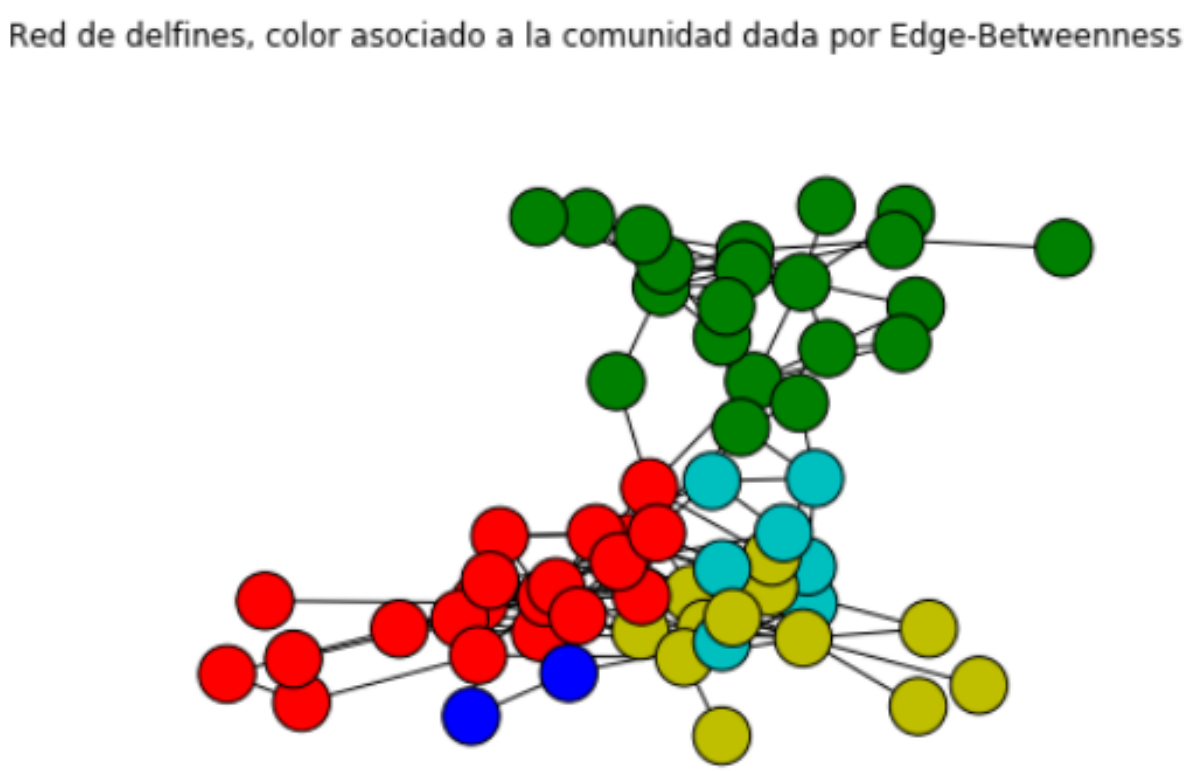

Figura 1: *Red de interacción de delfines, en distintos colores las comunidades detectadas por el método de Edge-Betweenness.*

El método detecta cinco comunidades de distintos tamaños, de 21,20,12,7 y 2 nodos respectivamente.

Fast greedy **aglomerativo**

El algoritmo *fast greedy* o codicioso es otro de los métodos para detectar estructuras de comunidades en redes. La ventaja de este método es que se trata de un algoritmo rápido y no computacional-dependiente. Este método, al igual que el anterior, se basa en maximizar la modularidad.

El método comienza con la red dividida en comunidades de un solo nodo, luego se inspecciona cada par de comunidades-o nodos- conectadas mediante al menos un enlace y se estima la variación de modularidad (ΔQ) producida si se combinasen, para luego unir el par de comunidades de mayor ΔQ. Se repite el procedimiento, calculando en cada paso la modularidad de la red, hasta finalmente llegar a una única comunidad que incluya a todos los nodos. Luego se elegirá la partición que tenga máximo valor de modularidad.

En la Figura 2 se muestra la red de delfines, coloreada por comunidades de acuerdo al método *Fast Greedy*. El algoritmo da una lista de todas las particiones generadas como se explicó anteriormente. Se eligió la partición con mayor modularidad, la cual presenta cuatro comunidades con 24, 21, 15 y 2 nodos respectivamente.

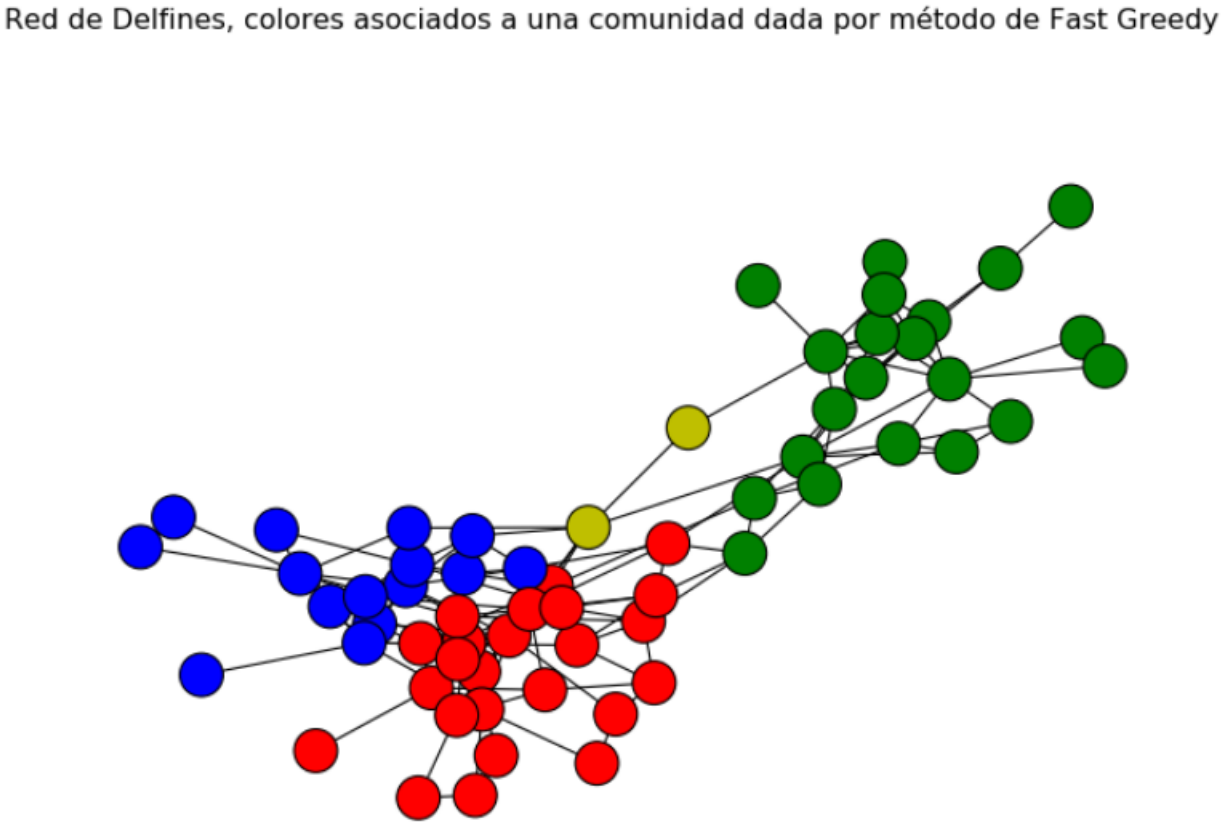

Figura 2: *Red de interacción de delfines, en distintos colores las comunidades detectadas por el método Fast Greedy.*

Louvain

El algoritmo Louvain es una heurística de reconocimiento de comunidades de complejidad computacional del orden del número de enlaces, por lo que es especialmente útil para redes pesadas. El método se basa en optimizar la modularidad en pasadas de dos pasos. En el primer paso se optimiza la modularidad con cambios locales, tratando de unir un nodo con sus vecinos. Se elige el cambio de mayor ΔM (si el cambio es positivo). Se repite esto para cada nodo de la red. En el segundo paso, se arma una nueva red donde cada nodo es una comunidad encontrada en el Paso 1. Se generan autoenlaces que corresponden a lazos intracomunidad. Estos dos pasos se repiten en varias pasadas hasta no poder obtener un incremento de modularidad.

En la Figura 3 se muestra la red de delfines, coloreada por comunidades de acuerdo al método de Louvain. El método detecta cinco comunidades, de 20,18,7,12 y 5 nodos.

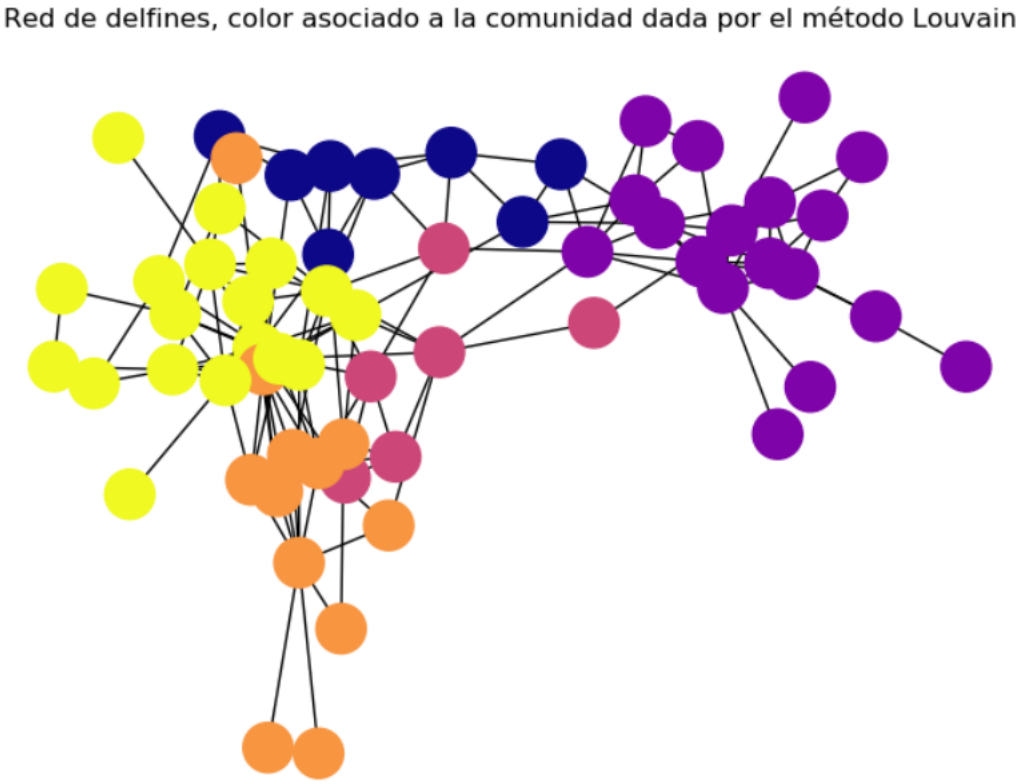

Figura 3: *Red de interacción de delfines, en distintos colores las comunidades detectadas por el método Louvain.*

Infomap

El método de detección de comunidades llamado Infomap se basa en el hecho de que existe una dualidad entre el problema de comprensión de datos y detectar una estructura y patrones en los mismos. Infomap usa caminatas aleatorias como indicador de estructuras (o comunidades) en la red. Los principales ingredientes de Infomap son: 1) La *map equation*: dada una partición de la red en comunidades, la *map equation* es la manera de cuantificar que tan eficiente es la descripción de una caminata aleatoria sobre la red. 2) Minimización de la *map equation* frente al conjunto de todas las posibles particiones de la red en comunidades. La idea es que una caminata aleatoria sobre la red te da información sobre las comunidades de ésta. Como se define una comunidad como un conjunto de nodos densamente conectados, el caminante que se encuentra en un nodo del cluster i tiene más probabilidad de pasar a un nodo del mismo cluster en el siguiente paso que moverse a uno de otro cluster, debido a la alta conectividad entre nodos de un mismo cluster. Teniendo esto en cuenta la idea en la que se basa infomap es en describir la caminata aleatoria sobre la red de la manera más eficiente posible (la menor cantidad de etiquetas posibles). Infomap propone un etiquetado de dos niveles: *index code-book* (códigos asignado a entrada y salida de una comunidad) y *module code-book* (código interno de cada nodo dentro de su comunidad). Con esta estrategia se pueden reutilizar nombres (en particular nombres cortos). En este esquema de comprensión **encontrar una descripción óptima (el *index code-book* por ejemplo)**

es encontrar buenos clusters. En la *map equation* están los bits necesarios en promedio para describir movimientos entre comunidades y los bits necesarios en promedio para describir movimientos dentro de comunidades. Por lo tanto, la descripción óptima en términos de longitud de descripción óptima (es decir buenos clusters) se logra minimizando dicha *map equation* frente al conjunto de todas las posibles particiones de la red en comunidades.

En el caso de nuestra red, el método de Infomap detecta seis comunidades en la red de delfines. Cada *cluster* cuenta con 20,16,12,7, 5 y 2 nodos respectivamente. En la Figura 4 se muestra la red coloreada según las comunidades halladas.

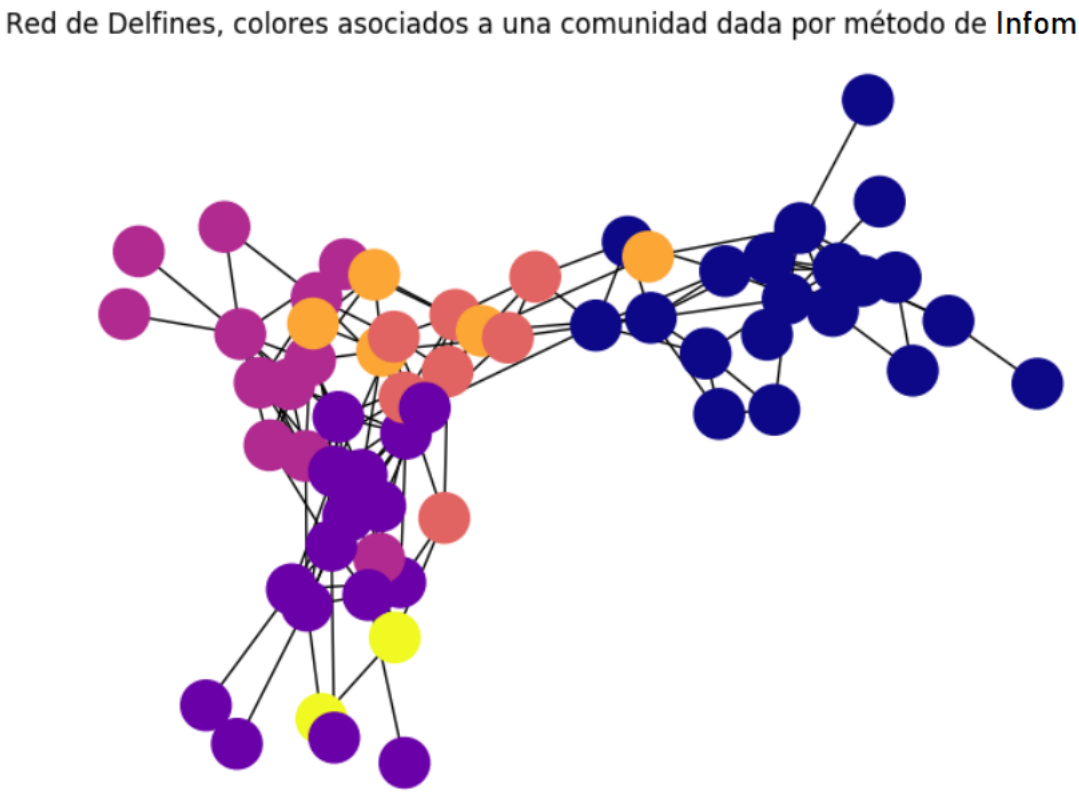

Figura 4: *Red de interacción de delfines, en distintos colores las comunidades detectadas por el método Infomap.*

Análisis de los resultados y discusión

Se puede notar que cada método arroja distintas comunidades, por lo que nos es necesario evaluar los resultados obtenidos según cada método. Para ello, se utilizarán tanto medidas internas como externas para evaluar las particiones obtenidas.

En primer lugar, se calculó la modularidad de las cuatro particiones. En la Figura 5 se muestran los valores obtenidos.

Modularidad			
Edge Betweenness	Fast Greedy	Louvain	Infomap
0.5193	0.492	0.5277	0.5277

Figura 5: Valores de modularidad para los cuatro métodos de detección de comunidades.

El método que arroja una partición más modular es Louvain, si bien los valores son similares. Para analizar si las particiones son efectivamente modulares hay que compararlas con algún modelo nulo. Para cada método de detección, se recableo la red y se calculó la modularidad en la red recableada manteniendo los clusters de la red real. Este procedimiento se itero un determinada cantidad de veces. Se generaron histogramas con las modularidades de las redes recableadas y se compara el valor real con la distribución creada.

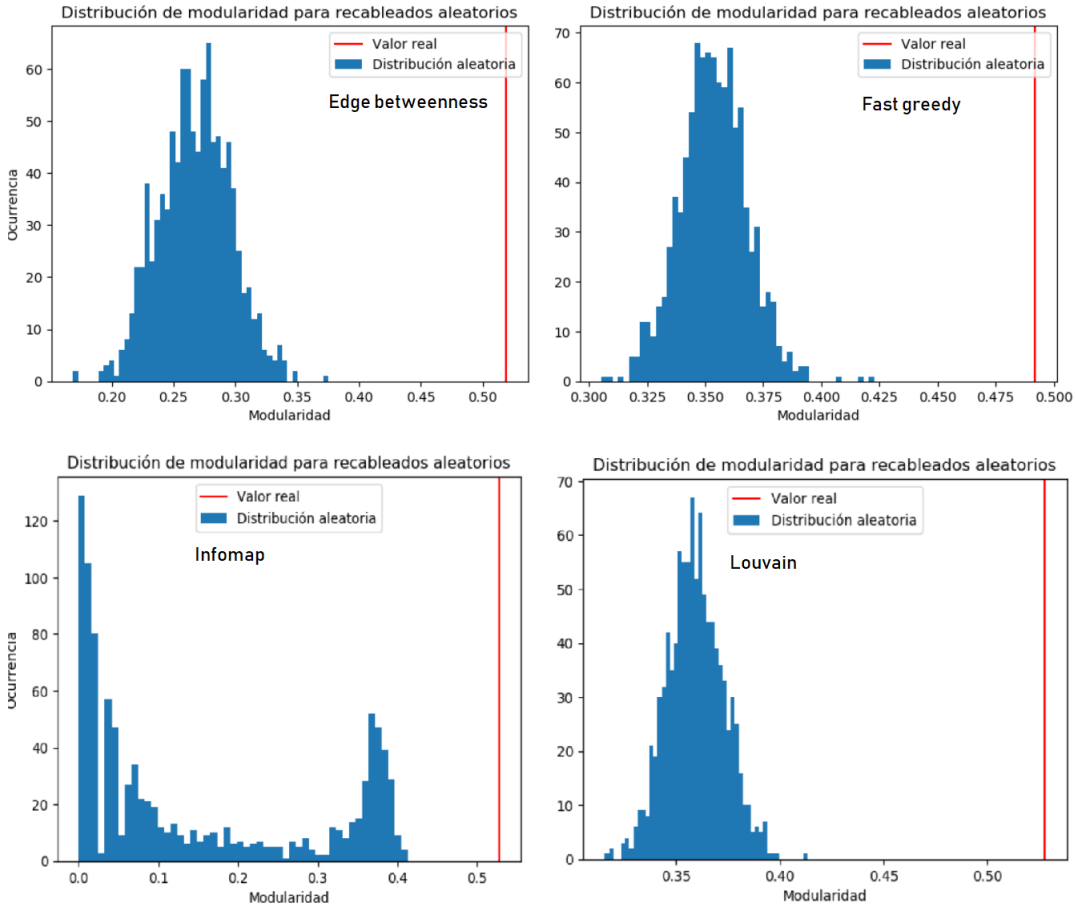

Figura 6: Distribuciones de modularidad para particiones según cada uno de los métodos de detección de comunidades para las redes recableadas. En rojo el valor de modularidad de la red real.

Pareciera que todas las particiones son modulares, por lo que la red debe serlo. Para cuantificar esta afirmación se puede dar el *p-value* en cada distribución. Habiendo realizado mil iteraciones, para los cuatro métodos no hay valores de modularidad más extremos que el valor de la red real. Por lo que podemos decir que para los cuatro métodos el *p-value* es menor a $\frac{1}{1000}$. Con esto se podría afirmar que la red de delfines es modular.

Otra medida interna para evaluar particiones es *Silhouette*. Esta propiedad está definida para cada nodo de la partición, lo que permite realizar distintas evaluaciones de la partición. En primer lugar, se

puede dar el valor promedio sobre todos los nodos de la red para cada una de las cuatro particiones. Los valores obtenidos se muestran en la Figura 7.

Silhouette promedio			
Edge-Betweenness	Fast Greedy	Louvain	Infomap
0.2876	0.1382	0.2664	0.2664

Figura 7: Valores de *Silhouette* promedio sobre la red para cada método de detección.

En la Figura 7 vemos que el mejor método de detección de comunidades según la medida de silhouette promedio es Edge-Betweenness (es la que tiene valor más alto de silhouette promedio). Por otro lado, la peor partición según silhouette promedio es Fast greedy.

Por otro lado, como el valor de *Silhouette* está definido para cada nodo, se puede realizar gráficos de los valores para todos los nodos de la red. Se identifica con el color a qué cluster corresponde cada nodo, en negro el valor de *Silhouette* promediado sobre la red.

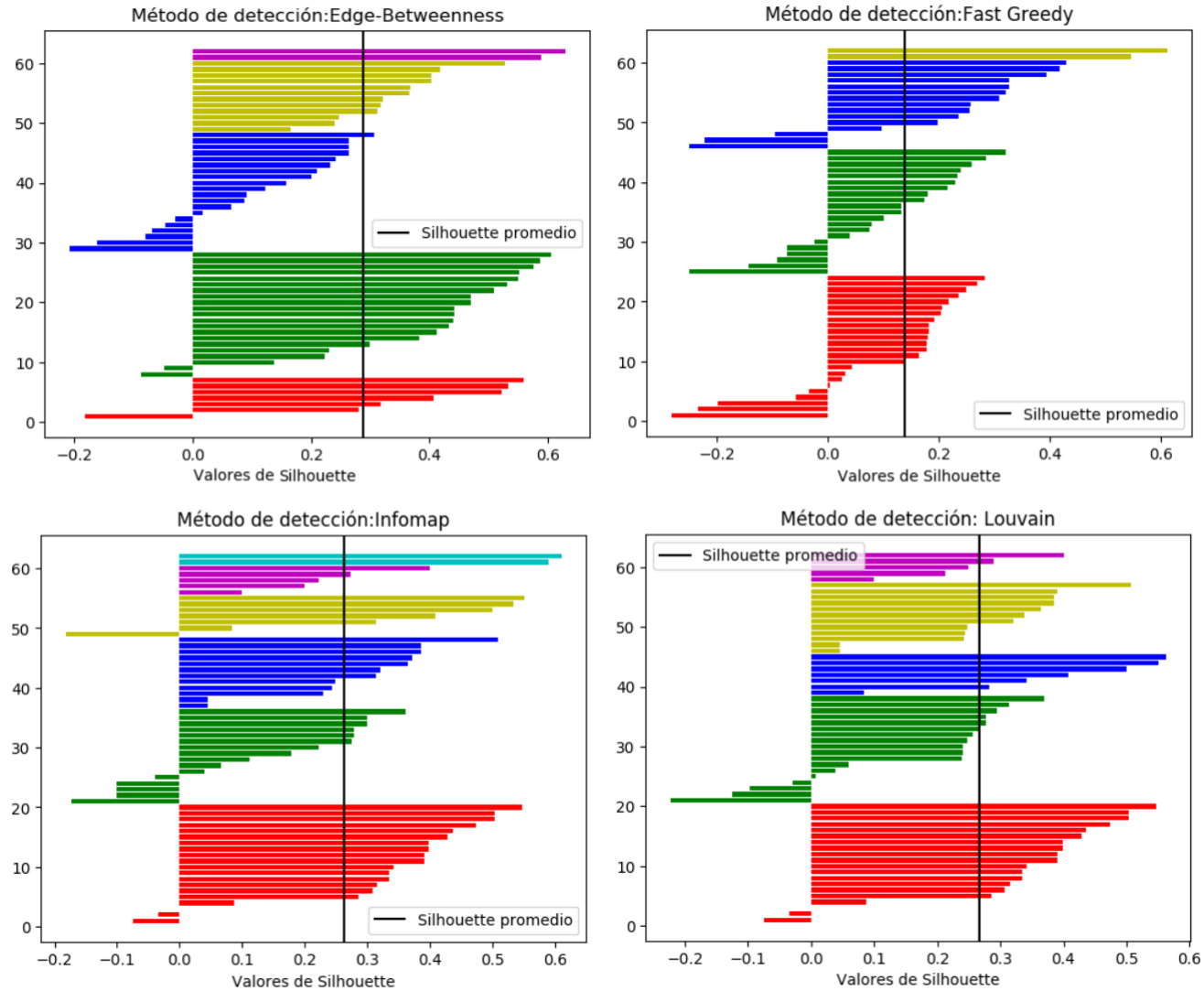

Figura 8: **Valores de *Silhouette* para los nodos de la red.** El color está asociado a un *cluster* dado el método de detección considerado.

Si la partición es buena espararía que no haya nodos que tengan asociados un valor de *Silhouette* negativos, que implica que están mal ubicados pues la distancia a los nodos de otros clusters es menor a la de los nodos del mismo cluster. Se puede ver que todas las particiones arrojan nodos con *Silhouette* negativo, siendo el método de Louvain el que presenta menos valores indeseados. En el otro extremo, el método *Fast Greedy* es el que más valores de *Silhouette* negativo tiene.

Lo que también se espera de una buena partición es que los clusters que la componen tengan un valor de *Silhouette* similar entre sí, y por ende cercano al promedio total. En la Figura 9 se muestra para cada partición el valor de Silhouette promedio para cada cluster. Se identifica a cada cluster por su tamaño.

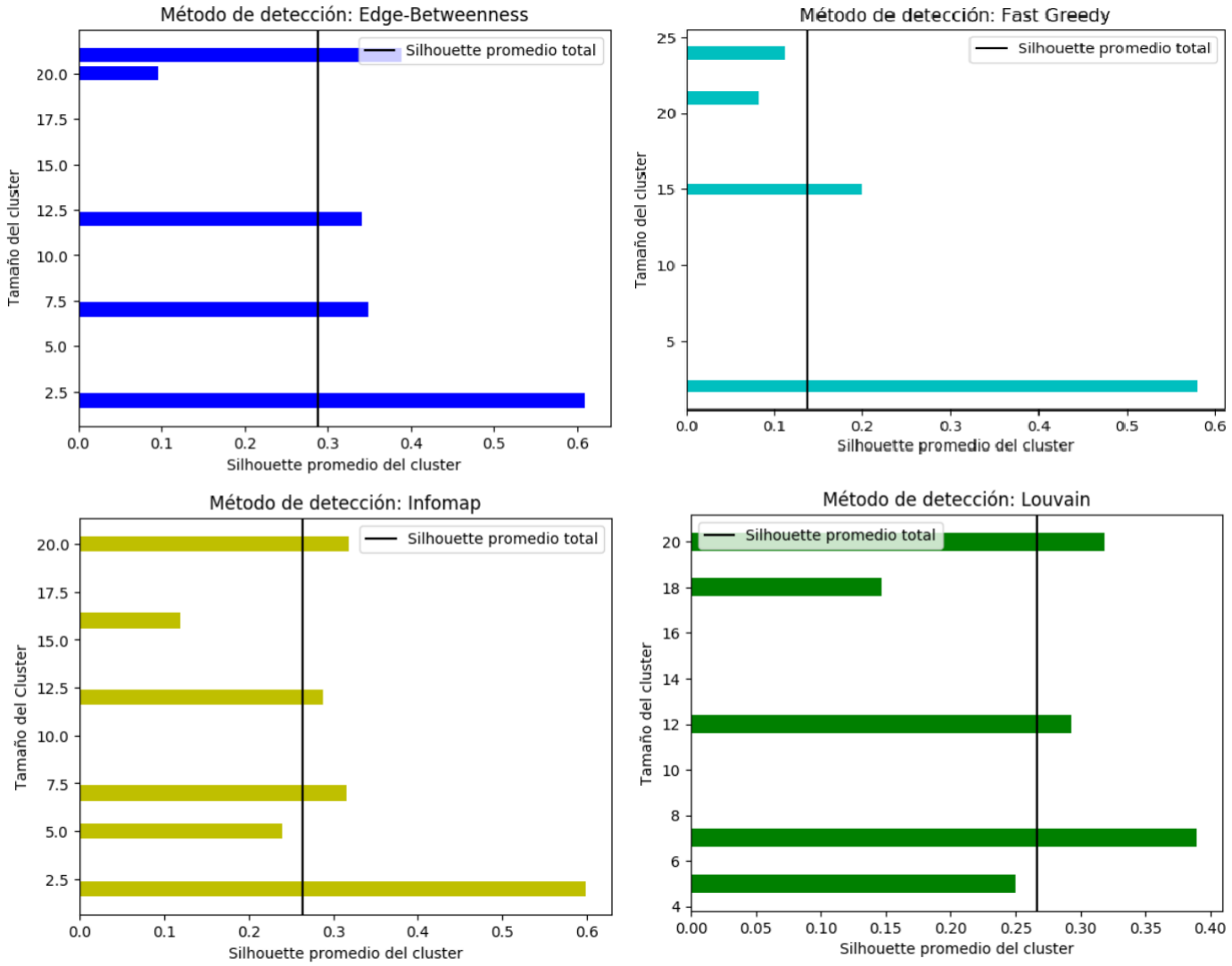

Figura 9: **Valores de *Silhouette* promedio para los *cluster* de la red, según el método calculado.** La línea negra marca el valor de *Silhouette* promedio sobre todos los nodos de la red.

Se puede ver que en los gráficos para *Edge-Betweenness*, *Fast Greedy* e Infomap hay un cluster con dos nodos de *silhouette* grande comparado con la media. El caso extremo es el del método *Fast Greedy*, donde el resto de los clusters tiene *silhouette* muy por debajo de la media, por lo que pareciera no ser una buena partición.

Para dar una medida de cuán buena es una partición en términos del criterio anterior se calculó la

desviación de los valores de *Silhouette* promedio para cada cluster al valor promediado sobre todos los nodos de la red, pesada por este último. De alguna manera, se puede ver cuánto se alejan el valor para cada cluster en términos del valor medio.

Edge-Betweenness	Fast Greedy	Infomap	Louvain
0.2122	0.1881	0.1954	0.1203
0.3511	0.4028	0.5548	0.2923
0.6654	0.4441	0.0818	0.3450
0.1847	3.2078	0.1841	0.2104
1,117		0.1028	0.3711
		1.2488	

Figura 10: Desviación de los valores de *Silhouette* para cada cluster al valor promedio según la partición.

Se puede notar que en las primeras tres particiones hay un cluster cuyo *silhouette* medio supera al valor promedio sobre la red en su valor o más. Esto se condice con lo señalado anteriormente, correspondiéndose con los clusters de dos nodos. En el caso de Fast Greedy ese cluster es más de cuatro veces el valor global. Esto pareciera indicar que la partición no resulta muy representativa.

A partir de la Tabla 10 se puede definir un criterio para definir si el método de detección de comunidades da una partición buena o mala, en términos de la medida de *Silhouette*. Un criterio para decir que las particiones son buenas es que más de la mitad de los clusters tengan desviación menor a 0.5. En este caso, todas las particiones son suficientemente buenas. Si se elige en cambio un criterio más restrictivo, pidiendo que más de la mitad tengan una desviación menor a 0.25, se obtiene que sólo la partición arrojada por Infomap cumple esta condición.

Por otro lado, se puede caracterizar de forma cuantitativa el acuerdo entre las particiones utilizando medidas de evaluación con información externa. Para eso se calculó la precisión y la información mutua entre particiones. En la Figura 11 se muestran los resultados obtenidos.

Edge betweenness	0.905341	0.843469	0.936542
0.783403	Louvain	0.879958	0.9688
0.662148	0.741122	Fastgreedy	0.903755
0.873041	0.911403	0.796621	Infomap

Precisión Info mutua

Figura 11: **Tabla con los valores de medidas externas para cada par de particiones.** En la triangular superior se exhiben los valores de precisión, mientras que en la inferior se muestran los valores de información mutua.

De acuerdo con la Tabla 11, si bien todas las particiones se asemejan entre sí en términos de la precisión (el valor menor es 0.84 entre *Fast greedy* y *Edge Betweenness*), las particiones más afines son

Louvain e *Infomap*. Se obtiene el mismo resultado analizando la información mutua.

Por último, se analizó cuantitativamente la relación entre el género de los delfines y la estructura de comunidades del grupo. Para una dada partición, se calculó el número de delfines masculinos y el número de femeninos dentro de un dado cluster. Luego, se asignaron los géneros de los nodos al azar y se recalculo el número de delfines masculinos y femeninos en cada cluster. La reasignación aleatoria de géneros se itero cien mil veces. Con esos datos, se armaron histogramas del número de delfines masculinos/femeninos en un dado cluster y se lo comparo con el valor real de nodos masculinos/femeninos en ese mismo cluster. En las Figuras 12, 13, 14 y 15 se muestra lo obtenido para cada partición.

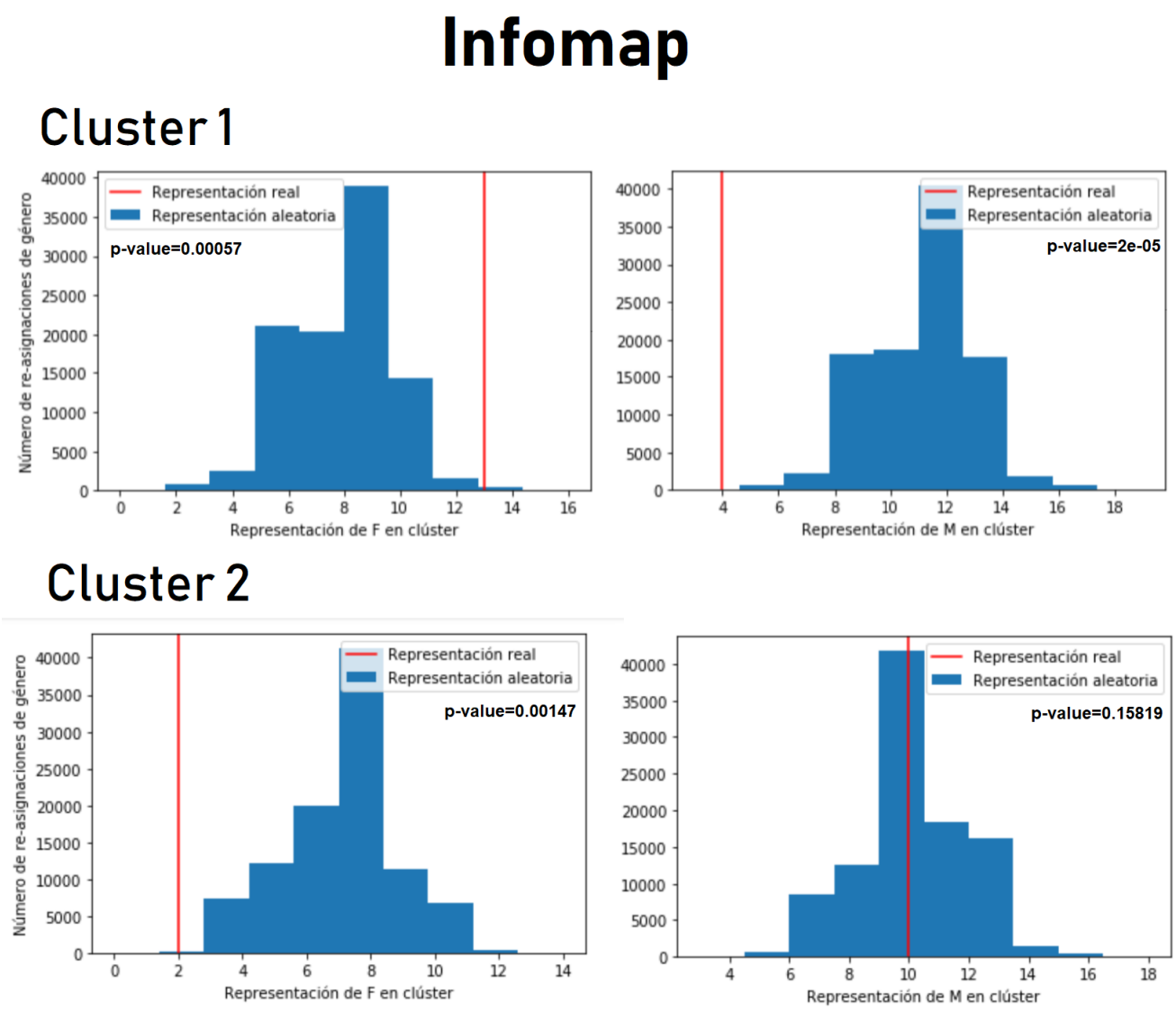

Figura 12: Histograma del número de delfines masculinos y femeninos en dos clusters de la red con los géneros asignados al azar. En rojo el valor real de delfines masculinos/femeninos en esos mismos clusters

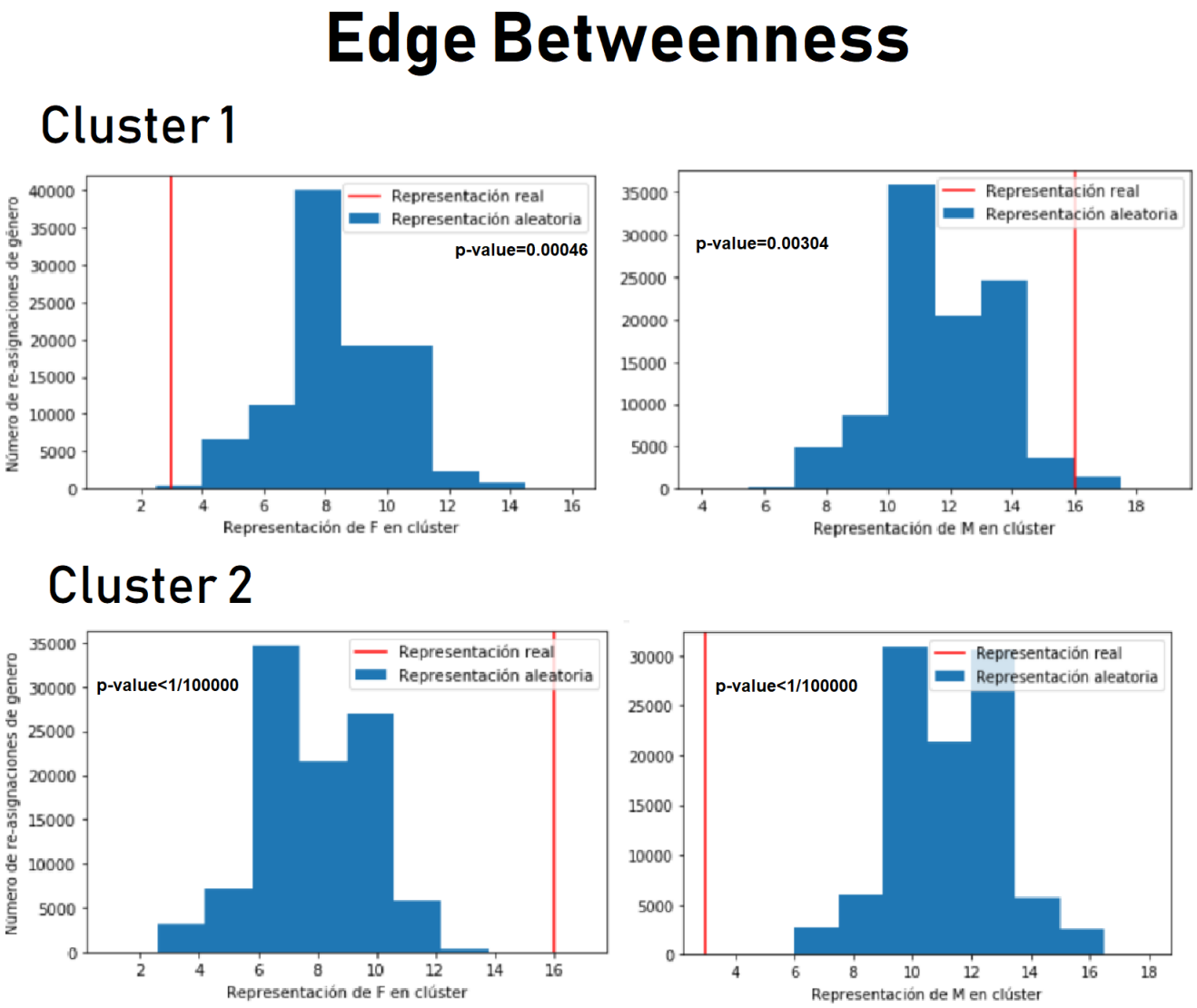

Figura 13: Histograma del número de delfines masculinos y femeninos en dos clusters de la red con los géneros asignados al azar. En rojo el valor real de delfines masculinos/femeninos en esos mismos clusters

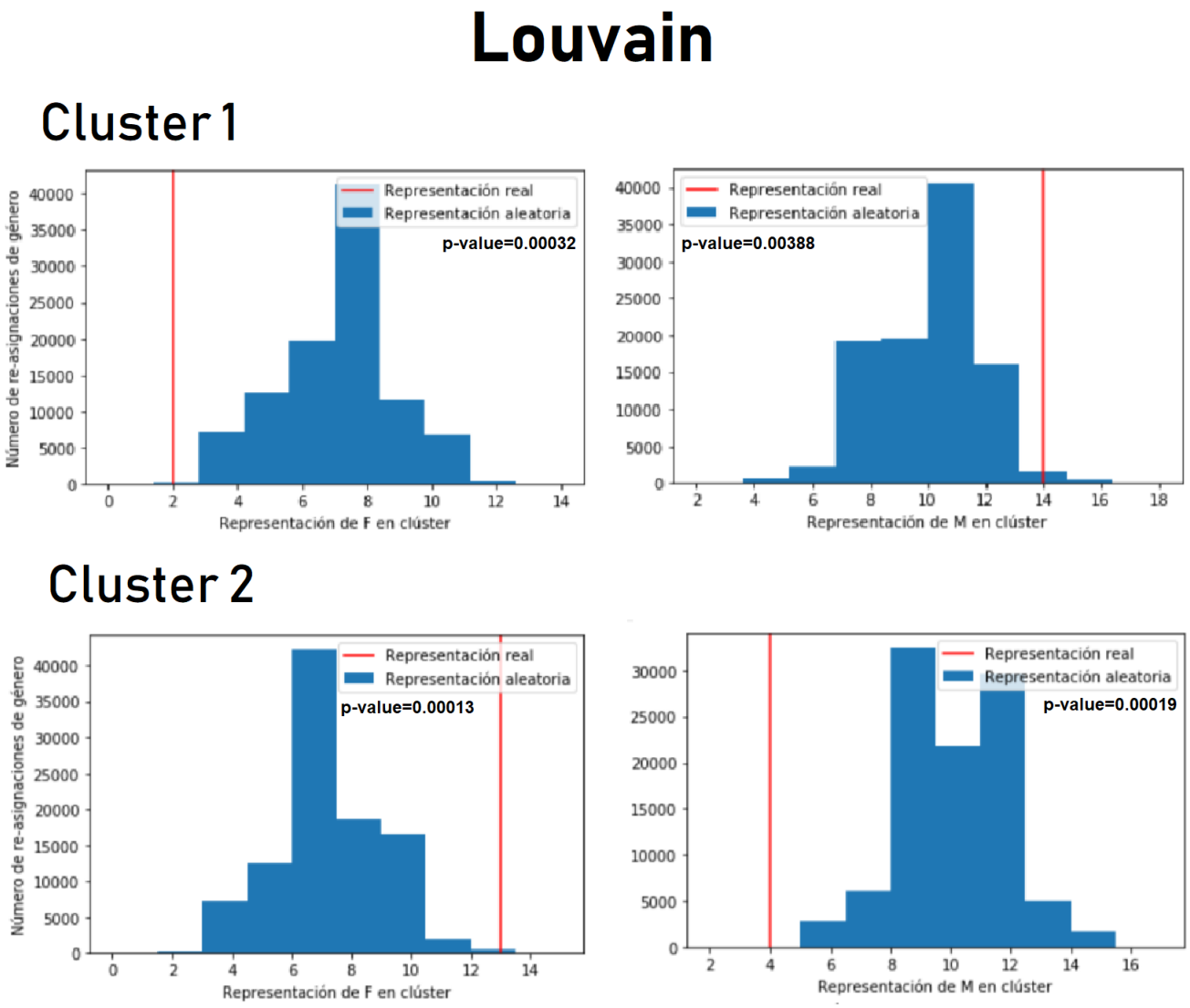

Figura 14: Histograma del número de delfines masculinos y femeninos en dos clusters de la red con los géneros asignados al azar. En rojo el valor real de delfines masculinos/femeninos en esos mismos clusters

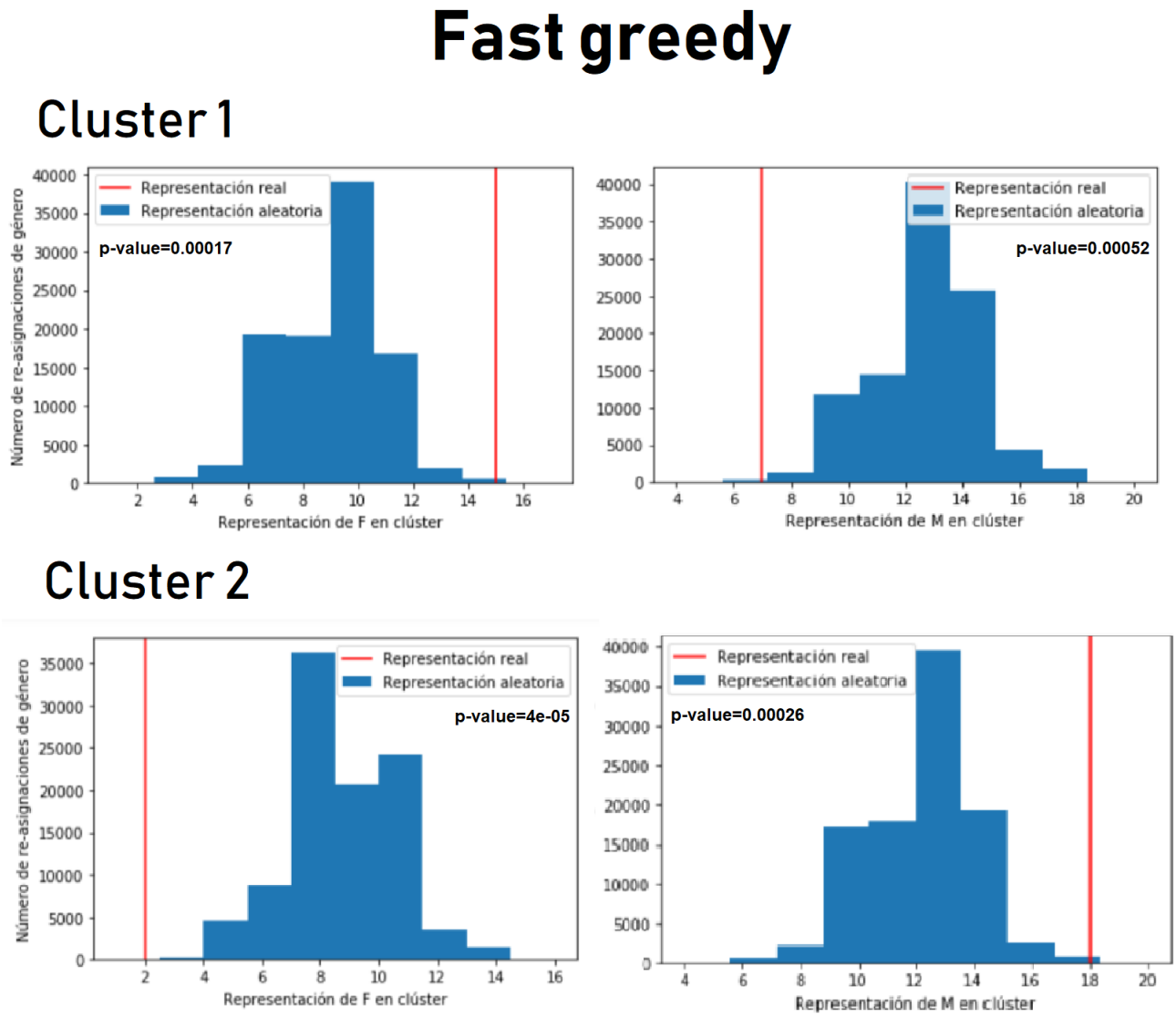

Figura 15: Histograma del número de delfines masculinos y femeninos en dos clusters de la red con los géneros asignados al azar. En rojo el valor real de delfines masculinos/femeninos en esos mismos clusters

En las Figuras 12, 13, 14 y 15 se puede ver que en general hay una correlación entre el género de los delfines y la estructura de comunidades de la red. Esto se ve en que el valor real suele caer en valores extremos de la distribución, como puede verse cuantitativamente en el *p-value*. En otras palabras, en general, delfines del mismo género tienden a estar en un mismo cluster, independientemente de cual sea el método de detección de comunidades usado.

Si bien los distintos algoritmos arrojan comunidades distintas, pareciera que resuenan con alguna estructura propia de la red, asociada al agrupamiento por géneros de los delfines.