Trabajo Computacional Nº 3

Fernando Cornes - Juan Herrera Mateos - Ignacio Sticco
Grupo 3

Redes Complejas con Aplicaciones a Sistemas Biologicos
7 de noviembre de 2018

1. Introducción

El término comunidad tiene su origen en el vocablo latino *communitas*, y se refiere a un conjunto, una asociación o un grupo de individuos (que pueden ser de seres humanos, animales ,etc), que comparten características, intereses u objetivos en común [1]. Por ejemplo una comunidad de personas puede delimitarse mediante el idioma, las costumbres, su visión del mundo, sus problemas y/o intereses. Actualmente encontramos comunidades en las llamadas redes sociales; el aumento exponencial de usuarios en las mismas ha incrementado el interés en el estudio de éstas.

En el contexto de las redes complejas, las comunidades, también conocidas como módulos o *clusters*, se definen de manera sencilla como agrupamientos de nodos similares. A menudo en este tipo de redes se encuentra una alta concentración de enlaces internos en cada uno de esos grupos, y una baja concentración de enlaces entre cada una de esas regiones. Los nodos en una dada comunidad tienen, por lo tanto, una mayor probabilidad de enlazarse entre sí que a nodos de otro grupo. Entonces, a partir del concepto de densidad de una red, las comunidades pueden definirse como grupos de nodos densamente conectados que presentan conexiones dispersas entre sí [2].

Puede ocurrir que un nodo de la red pertenezca a más de una comunidad, en cuyo caso se dice que la red está solapada. Estas situaciones se observan cotidianamente; por ejemplo, en una red social como Facebook, una persona cualquiera (representada por un nodo) puede pertenecer a varias comunidades o grupos si tomamos en cuenta sus intereses. En este caso no podríamos asignar cada vértice (persona) a una sola comunidad, y es aquí donde aparece el concepto de comunidad solapada, las cuales pueden compartir nodos entre sí.

1.1. Enfoques para la detección de comunidades

Según Newman y Girvan [12], existen dos formas diferentes para el descubrimiento de comunidades en redes complejas:

- ✓ **Particionamiento de grafos:** tiene su origen en la Informática, en el campo de la computación distribuida. Busca la mejor forma de asignar tareas a procesadores para minimizar las comunicaciones entre ellos. Consiste en dividir el grafo en g clústers de tamaño predefinido, de tal forma que los enlaces dentro de cada uno de ellos resultan ser más denso que los enlaces entre los clústers [6].

- ✓ **Modelado de bloques, también llamado clustering jerárquico o detección de la estructura de comunidades:** se origina en Sociología. Está motivado por el descubrimiento de grupos en una sociedad para facilitar el análisis de fenómenos sociales. Este enfoque se basa en la idea de que el grafo tiene una estructura jerárquica, es decir, pequeños grupos de nodos son parte de grupos medianos de nodos, que a su vez pertenecen a grupos más grandes y así sucesivamente.

En cualquier caso, el procedimiento implica dividir el grafo original en un conjunto de subgrafos disjuntos mediante la optimización de una función objetivo (por ej. función de la modularidad que será descripta posteriormente). El propósito de los dos enfoques es descubrir grupos de nodos relacionados en la red, y si es posible, la estructura jerárquica correspondiente, a partir de la información proporcionada por la topología de la red. Los descriptos devuelven particiones disjuntas del conjunto de nodos, es decir, no permiten el solapamiento de comunidades y cada nodo pertenece a una única comunidad.

1.2. Índices para evaluar y detectar comunidades

En la detección de comunidades en alguna red, todos los métodos propuestos hasta hoy en día tienen su base en alguna métrica para determinar la calidad de las comunidades halladas. La modularidad y el índice Silhouette son los más importantes en la comprensión de las propiedades estructurales de la red [10].

- ✓ **Modularidad:** Es la función de calidad más utilizada y fue propuesta por Newman y Girvan en 2004 [12]. Se basa en la idea de que una distribución en clústers no es lo que se espera por azar en una red, y por tanto, trata de cuantificar la intensidad de esta estructura de comunidades comparando la densidad de enlaces dentro y fuera de cada una de ellas con la densidad que se esperaría si los enlaces estuviesen distribuidos aleatoriamente en la red. A este modelo estadístico se le debe dotar de un modelo nulo que especifique qué es lo que se espera por azar. En este caso, los autores parten de la base de que la distribución de grados de los nodos es una propiedad intrínseca de la red y proponen un modelo nulo siguiendo este principio. La siguiente fórmula es la utilizada para calcular la modularidad (Q) de una partición determinada (P):

$$Q(P) = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta\left(c_i, c_j\right), \tag{1}$$

  donde: A_{ij} es un elemento de la matriz de adyacencia, m el número de enlaces de la red, k_i el grado del nodo i-ésimo, $\frac{k_i k_j}{2m}$ según el modelo nulo es el número de enlaces entre los nodos i-ésimo y j-ésimo, y $\delta\left(c_i, c_j\right)$ corresponde a una función binaria que toma valor igual a uno si los nodos i y j están en la misma comunidad o partición y cero el en caso contrario.

  La idea es que la red muestra una estructura modular coherente si el número de enlaces entre comunidades es menor que el esperado en una red aleatoria. Como $Q\,\epsilon\,[-1, 1]$, cuanto mayor es su valor, mejor es la partición, es decir, las comunidades encontradas están densamente conectadas internamente (hay más enlaces de los que cabría esperar aleatoriamente) y dispersamente conectadas entre sí. En una red aleatoria $Q = 0$. La modularidad se usa tanto para comparar la calidad de distintas particiones como para diseñar métodos de descubrimiento de comunidades que traten de maximizar su valor.

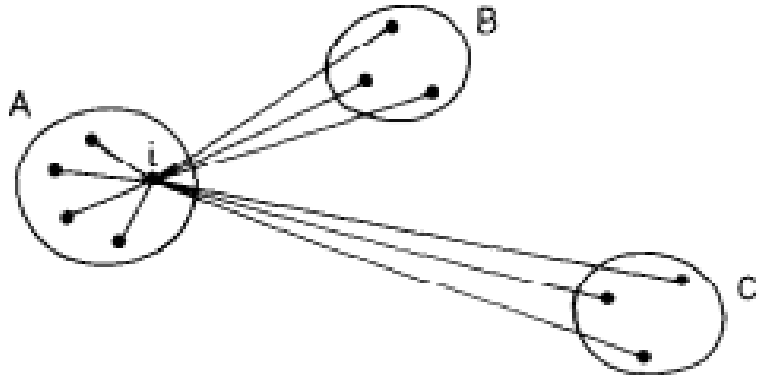
- ✓ **Coeficiente Silhouette:** Desarrollada por Rousseeuw en 1987 [14], esta métrica también mide la calidad de una configuración de un clúster. El índice determina qué tan bien está asignado un dado nodo a una comunidad, lo cual permite visualizar una distribución. Por ejemplo para un dado nodo i, lo que hace el algoritmo es calcular la distancia entre éste y sus vecinos del mismo clúster (A). Si B es un clúster vecino de A -ver Figura 1-, el más cercano a éste, el índice Sillouette del nodo i, $s(i)$, puede determinarse a partir de:

$$s(i) = \frac{b(i) - a(i)}{max\ \{a(i), b(i)\}}, \tag{2}$$

  donde: $a(i)$ es el promedio de distancias entre i y los demás nodos de A, y $b(i)$ el promedio de distancias entre i y los nodos del clúster vecino (clúster B).

El índice puede tomar un valor entre -1 y 1 , y cuanto mayor sea el valor de éste índice, mayor será la homogeneidad en el clúster; un valor de 1 representa un separación perfecta de otros grupos. Por ejemplo si $s(i) = 1$, el nodo i está bien ubicado en el clúster; si fuera $s(i) = -1$ se tiene el caso contrario. Los puntos en la frontera los cuales comunican un clúster con otro en general no están bien localizados.

Figura 1. Nodo i sobre el que se hace el cálculo del índice s.



Habiendo definido estos índices de calidad, es posible pasar a describir los algoritmos para la detección de comunidades que fueron utilizados a lo largo de este trabajo.

1.3. Algunos algoritmos para la detección de comunidades disjuntas

✓ Algoritmo de Blondel, también conocido como método de Louvain

La modularidad es una función que refleja que tan buena es una partición de un grafo: mientras mayor sea el valor de Q, mejor es la partición. Una buena estrategia, consiste en maximizar esa métrica. Dada la gran cantidad de particiones en un grafo, el problema de maximizar la modularidad es del tipo NP-Completo[4]. Es así como se han elaborado algoritmos voraces (*greedy*), que permiten lograr valores aproximados al óptimo de modularidad.

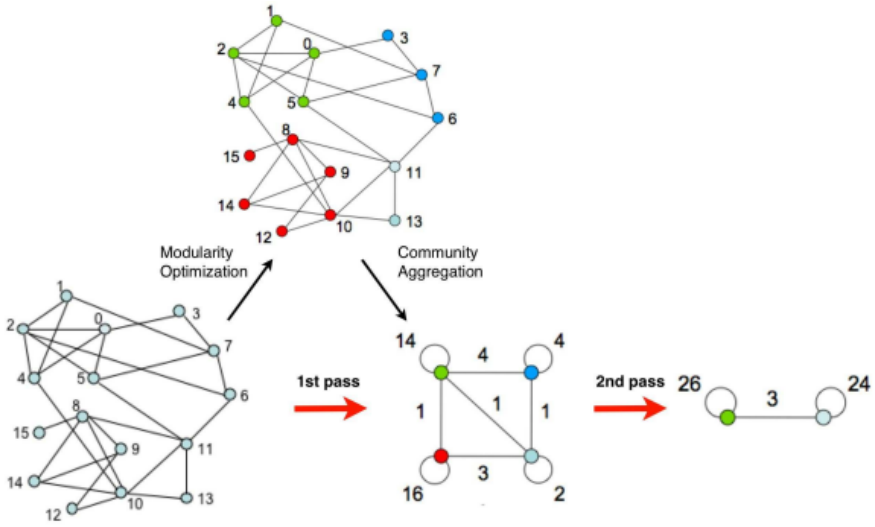
El método de Louvain se basa en la optimización de modularidad y es una de las técnicas más eficientes entre las mismas. Este algoritmo es capaz de procesar redes ponderadas y establecer una jerarquía entre las comunidades detectadas. La estructura general de Louvain se divide en dos fases o etapas principales, las cuales son aplicadas iterativamente: la primera es la encargada de formar las particiones y en la segunda se procesan las comunidades obtenidas y se identifican sus relaciones jerárquicas.

En la primera etapa se asigna a cada nodo una comunidad distinta. Luego se considera para cada nodo i su vecindario de nodos. Posteriormente se evalúa la ganancia de modularidad al remover el nodo i de su comunidad y colocarlo en la comunidad de j, un nodo vecino a i. El nodo i se une a la comunidad que entregue la máxima ganancia de modularidad con valor positivo, en otro caso continúa en la comunidad actual. Se observa que los nodos aislados nunca serán unidos a alguna comunidad, por lo que no afectan el resultado del proceso. Este proceso es aplicado iterativa y secuencialmente hasta que no se puedan hacer más mejoras locales en el grafo. Blondel señala que un mismo nodo puede ser analizado varias veces en este proceso, y además menciona que, aunque el proceso varía según el orden de los nodos seleccionados, las pruebas realizadas señalan que esta variación no es muy significativa en términos de modularidad, pero sí en los tiempos de ejecución del algoritmo [3].

La segunda etapa del algoritmo consiste en construir un nuevo grafo en el cual los nuevos nodos serán las comunidades encontradas en la etapa anterior. Luego se construyen los nuevos enlaces sumando los enlaces entre las comunidades fusionadas. Adicionalmente, se guardan los enlaces dentro de una misma comunidad como un enlace circular desde la comunidad i a la comunidad i. Una vez que esta etapa se completa se repite la etapa uno sobre este nuevo grafo.

El autor llama al proceso donde se utilizan estas dos etapas como una "pasada". Las pasadas son realizadas iterativamente hasta que no exista ninguna ganancia de modularidad en la etapa uno. En la Figura 2 se ilustra el proceso.

Figura 2. Visualización de los pasos del algoritmo. Cada pasada se compone de dos fases: una en la que la modularidad se optimiza permitiendo solo cambios locales de comunidades, y una donde las comunidades encontradas se agregan para construir una nueva red de comunidades. Los pasos se repiten de forma iterativa hasta que no se produce un aumento de modularidad posible.



✓ **Algoritmo Clauset, Newman et al. -*fast greedy*-**

Como mencionamos anteriormente, una de las técnicas que permiten obtener valores cercanos al óptimo de modularidad son los algoritmos voraces (greedy). Newman [11] con su método aglomerativo jerárquico fue uno de los pioneros en el área, sin embargo su método tenía problemas de eficiencia y complejidad. Clauset et al. [5] realizaron cambios en la manera en que era calculada la modularidad entre los grupos o clústers, eliminando todas las operaciones innecesarias y mejorando la elección de la métrica en cada iteración. Ellos propusieron un algoritmo jerárquico aglomerativo optimizado para la detección de comunidades en redes de dimensiones considerables. El algoritmo se basa en el cálculo de la modularidad de la Ec. 1, donde valores de Q superiores a 0.3 indican la presencia de una buena estructura de comunidad en la red o grafo.

El algoritmo determina a partir de un grafo constituido ya sea por tantos nodos como comunidades, los diversos incrementos de modularidad en cada posible unión de nodos de una determinada comunidad, seleccionado un ΔQ máximo (3) para fusionar esas comunidades.

Si inicialmente se establece (para grafos no pesados):

$$\Delta Q_{i,j} = \begin{cases} \frac{1}{2m} - \frac{k_i k_j}{(2m)^2}, & si\ i\ y\ j\ estan\ conectados \\ 0, & en\ otro\ caso, \end{cases} \tag{3}$$

$$a_i = \frac{k_i}{2m}. \tag{4}$$

Entonces el algoritmo se puede definir de la siguiente manera:

1. Calcular los valores iniciales (aumento de modularidad) de $\Delta Q_{i,j}$ y a_i de acuerdo a las ecuaciones anteriores en cada posible unión de nodos, y llenar una matriz, H, con el mayor valor de cada fila de la de la matriz ΔQ.

2.Seleccionar los valores $\Delta Q_{i,j}$ más altos de H y unir las comunidades correspondientes; actualizar la matriz ΔQ, la matriz H y a_i, e incrementar el valor de Q por $\Delta Q_{i,j}$.

3. Repetir el paso 2 hasta que quede sólo una comunidad.

✓ **Algoritmo Infomap**

La forma tradicional de identificar comunidades en grafos dirigidos y con pesos ha sido simplemente ignorar las direcciones y los pesos de los enlaces, con lo que se descarta información valiosa sobre la estructura del grafo. Esto es así porque mapeando el flujo de todo el grafo por las interacciones locales entre los nodos se logra conservar la información sobre la direccionalidad de las relaciones y los pesos de los enlaces.

Rossvall y Bergstrom [13] proponen el algoritmo Infomap, el cual se basa en la idea de buscar una descripción comprimida de un paseo aleatorio en un grafo dado. Su procedimiento es el siguiente:

1. A cada nodo se le da un nombre codificado mediante el código de compresión Huffman [9] que asigna etiquetas usando la probabilidad de visita de un caminante aleatorio a cada nodo-i .

2. Cada clúster recibe un nombre codificado [9].

3. Los nombres de los nodos pueden volver a ser utilizados (reciclados), siempre y cuando no se repitan en el mismo clúster.

4. El procedimiento de reciclado permite ahorrar el espacio requerido por la asignación de un nombre diferente a cada nodo.

5. Cuando un nodo pasa de un grupo a otro deberá indicar el nombre del nuevo clúster.

6. Si la red tiene una fuerte estructura de comunidad, entonces el reciclaje de los nombres de los nodos es conveniente.

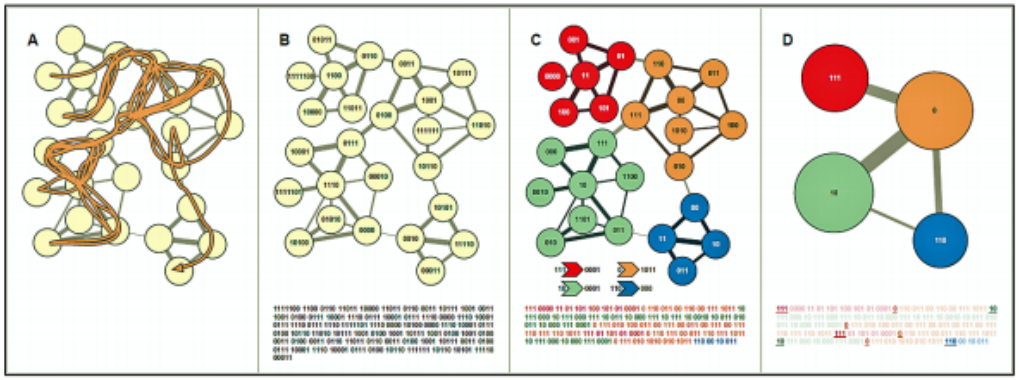
El método Infomap entiende la búsqueda de la estructura de comunidades como un problema de compresión de la información, detectando comunidades al comprimir la topología de la red. Se encuentra ilustrado en la Figura 3.

✓ **Edge Betweennes**

El método se basa en la eliminación de los enlaces que son menos centrales, los que unen o están entre comunidades. En lugar de construir comunidades añadiendo los enlaces más fuertes a un conjunto inicialmente vacío de nodos, éstas se construyen removiendo progresivamente enlaces del grafo original.

La intermediación o *betweenness*, una medida de centralidad e influencia de los nodos en la red, que utilizamos en el trabajo computacional pasado, fue propuesta por Freeman [7], y se define como el número de caminos más cortos entre pares de nodos que pueden pasar por un dado nodo de la red. Es una medida de la influencia de un nodo sobre el flujo de información entre otros nodos, especialmente en los casos que la información fluye en la red principalmente a través del camino más corto posible.

Figura 3. Detección de comunidades comprimiendo la descripción del flujo de información en grafos. En (A) se describe la trayectoria de un paseo aleatorio por el grafo -linea naranja-. En (B) mediante código de Huffman se le asigna un nombre único a cada nodo del grafo. Los 314 bits que se muestran bajo el grafo describen la trayectoria de mostrada en (A): se comienza con 1111100 para el primer nodo en el paseo de la esquina superior izquierda, 1100 para el segundo nodo, etc., terminando con 00011 para el último nodo del paseo aleatorio en la esquina inferior derecha. En (C) se muestra una descripción de 2 niveles del paseo aleatorio, en el que los principales clusters reciben nombres únicos, pues los nombres de los nodos dentro de los clusters se reutilizan. En (D) se muestran los nombres de los clusters y no las ubicaciones dentro de los mismos, proporcionando un óptimo granulado del grafo completo.



Este algoritmo para identificar comunidades consta de los siguientes pasos [8]:

1. Calcular el *betweenness* entre todos los enlaces de la red.

2. Remover los enlaces que tengan mayor valor (que en general son los que unen los clústers entre sí, pues por ellos pasan más caminos mínimos).

3. Recalcular el *betweenness* para todos los enlaces afectados por la eliminación.

4. Repetir desde el paso segundo hasta que no queden enlaces.

El resultado final del algoritmo de Girvan - Newman es un dendrograma. Cuando se ejecuta este algoritmo, el dendrograma se produce a partir de la parte superior hacia abajo (es decir, la red se divide en diferentes comunidades con la eliminación sucesiva de enlaces). Las hojas del dendrograma son nodos individuales y en éste se muestra la estructura completa de comunidades de la red

El objetivo de este trabajo es la detección de comunidades en una red generada a partir de la interacción de un conjunto de delfines. Para ello se utilizan los cuatro algoritmos descriptos previamente y se los examina por medio de los dos índices de calidad previamente definidos mencionados: la modularidad y la métrica silhouette.

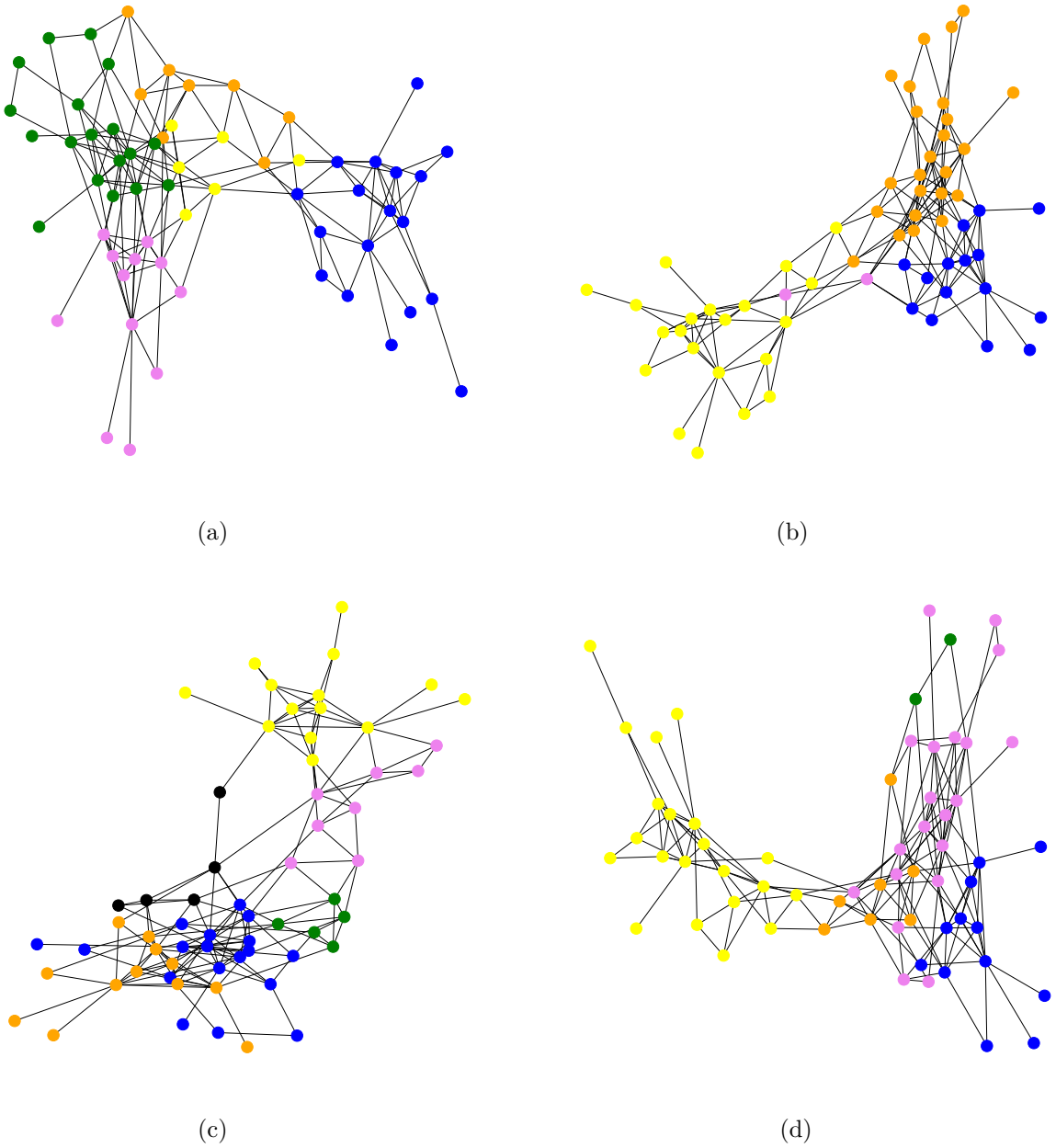
2. Resultados y discusión

2.1. Partición en clusters

En la Figura 4 mostramos a la red de 62 delfines de Nueva Zelanda. Cada gráfico corresponde a un algoritmo diferente de detección de comunas: Louvain, Fast greedy, Infomap y Edge betweenness. Los nodos fueron coloreados según la comuna de pertenencia. El número de comunas detectadas difiere según el algoritmo. Louvain detectó cinco, Fast greedy y Edge betweenness detectaron cuatro. Infomap devolvió una partición con

seis comunas, siendo este algoritmo el de mayor cantidad de comunas detectadas. En la Figura 8 se discutirá las causas que producen las diferencias en la cantidad de comunas según cada algoritmo.

Figura 4. Detección de comunas según los algoritmos utilizados: (a) Algoritmo de Louvain; (b) Algoritmo Fast Greedy; (c) Algoritmo Infomap; y, (d) Algoritmo Edge Betweennes.



(a) (b)

(c) (d)

2.2. Caracterización de las particiones

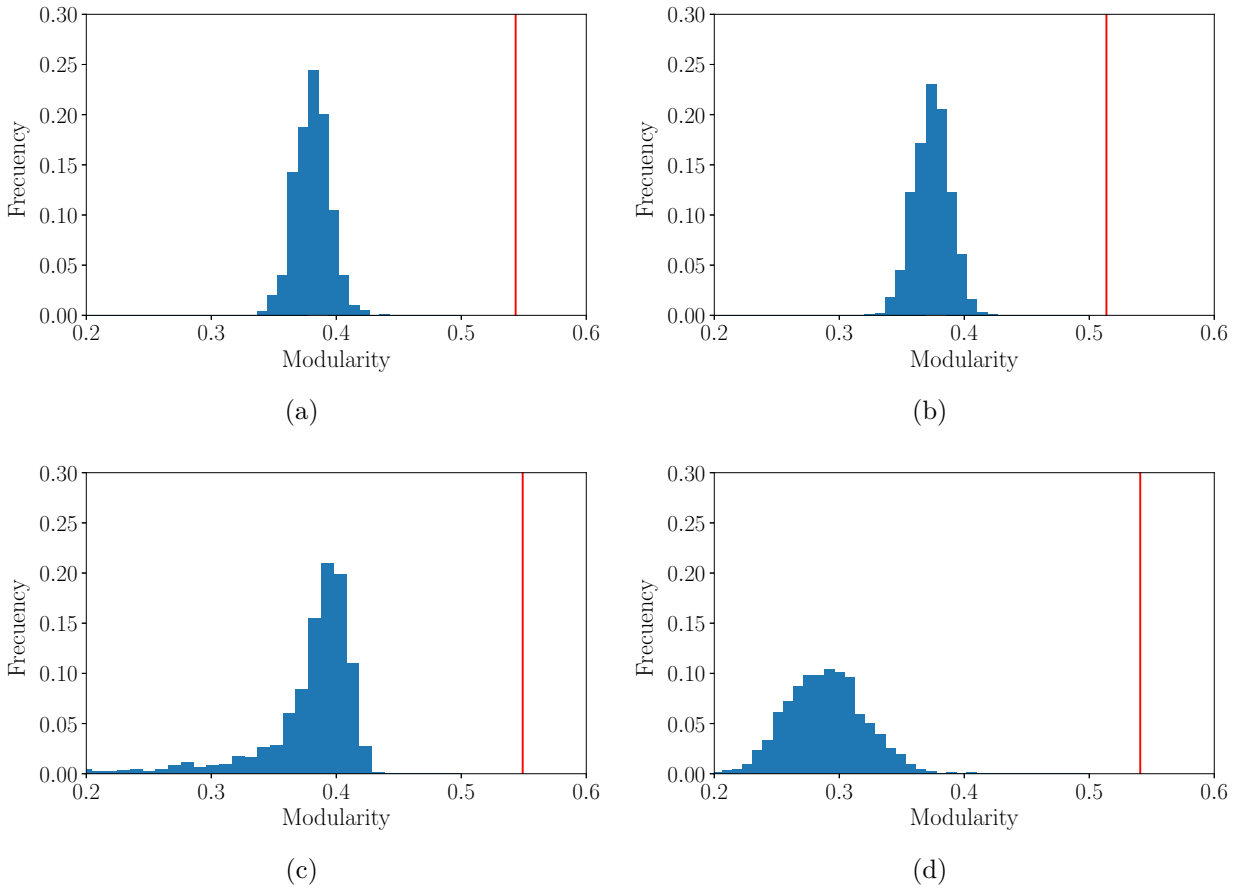
Modularidad

En la Figura 5 mostramos la modularidad para cada uno de los algoritmos de detección de comunas. En cada gráfico hay una recta roja que indica la modularidad de la red original según la partición en comunas dada por cada uno de los algoritmos empleados. Además, cada gráfico cuenta con un histograma que surge de

calcular modularidad sobre 1000 “recableos”de la red original. El proceso para obtener los histogramas tiene tres etapas: primero se realizó un recableo de la red de forma tal que cada nodo preserve la distribución de grado. Una vez realizado el recableo, se utilizaron los distintos algoritmos para calcular una partición de la red en comunas. En la última etapa se usaron la partición y la red recableada para calcular la modularidad.

Cabe destacar que en el caso de Infomap, solo fueron consideradas las particiones con más de siete comunas. En la Figura 6 se discute porque tuvimos esta consideración especial. Todas los algoritmos, excepto Infomap, presentan una distribución simétrica de modularidad para las redes recableadas. Las particiones dadas por todos los algoritmos dieron valores de modularidad mayor a los esperados por azar.

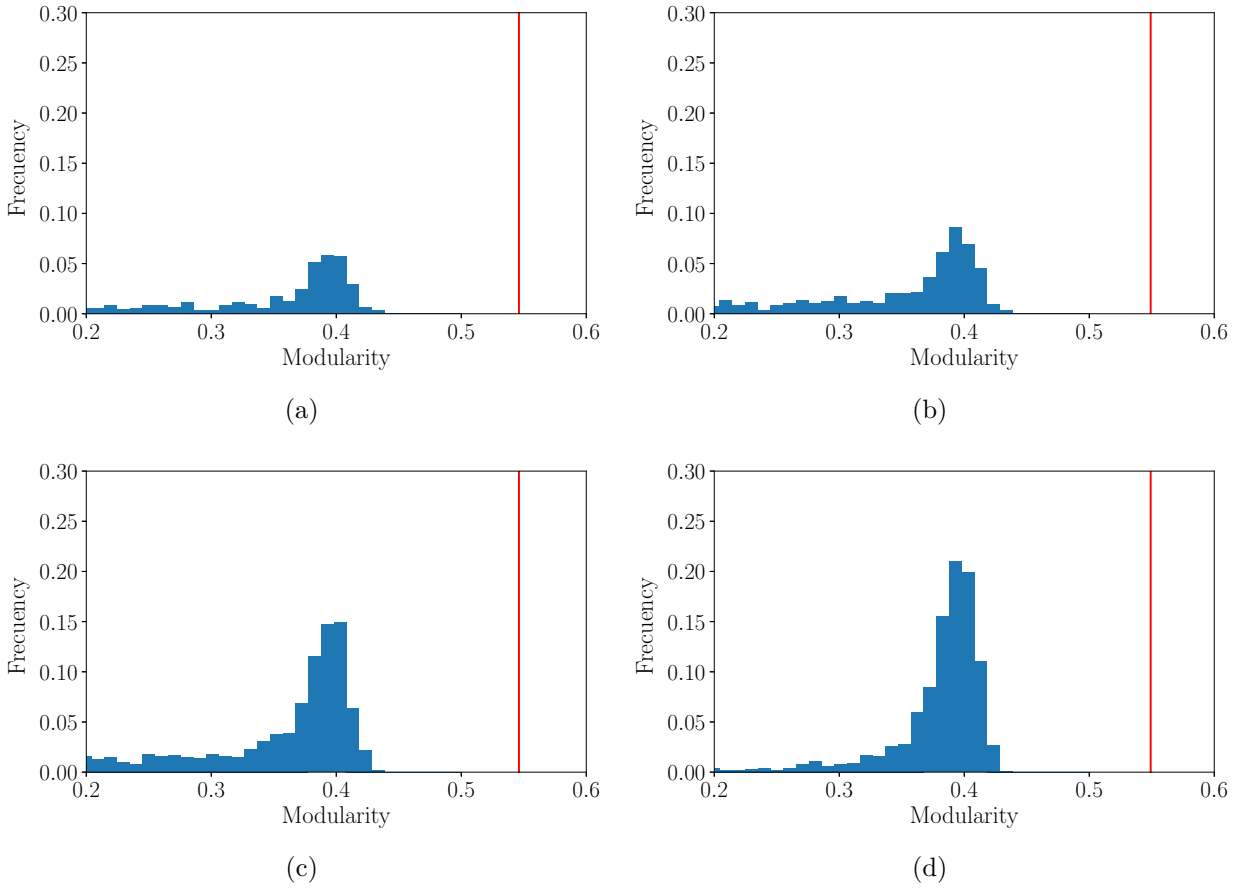
Figura 5. Histogramas de modularidad para distintos algoritmos de partición. Se realizaron 1000 iteraciones. En cada uno de los procesos se “recableó”la red un número de vece red original. La recta roja representa la modularidad de la red original: (a) Algoritmo de Louvain; (b) Algoritmo Fast Greedy; (c) Algoritmo Infomap (siete o más comunas); y, (d) Algoritmo Edge Betweennes.



En la Figura 6 mostramos cuatro histogramas de modularidad para redes recableadas usando el algoritmo Infomap de detección de comunas. Infomap devuelve particiones con un determinado número de comunas. Antes de calcular la modularidad, se excluyeron las particiones cuyo número de comunas era inferior a un determinado “umbral”. Cada uno de los gráficos de la Figura 6, muestra el histograma correspondiente para distintos umbrales considerados: 6(a) corresponde a particiones con cualquier número de comunidades, 6(b) corresponde a particiones con tres o más comunidades, 6(c) a particiones con 5 o más y 6(d) a particiones con siete o más comunidades.

Puede verse que a medida que se aumenta el umbral (se excluyen particiones con menos comunas), la distribución aumenta su valor más probable, es decir, la red se vuelve más modular. Esto ocurre porque cuando se hacen recableos de la red, ésta pierde la estructura (las comunas dejan de estar bien definidas). Como Infomap es un algoritmo que distingue comunas si hay poca información en común entre los nodos, tiende a agrupar muchos nodos dentro de una misma comuna en el caso de redes con estructura no definida (redes recableadas al azar). Es por eso que para muchos recableos al azar, Infomap detecta muy pocas comunas y esto hace que la modularidad tenga un sesgo hacia valores cercanos a cero (Fig. 6(a)).

Figura 6. Histogramas de modularidad para utilizando el algoritmo Infomap. Se realizaron 1000 iteraciones. En cada uno de los procesos se "recableó"(modificación de los enlaces de los nodos) la red un número de vece red original. La recta roja representa la modularidad de la red original: (a) Una o más comunidades; (b) Tres o mas comunidades; (c) Cinco o más comunidades; y, (d) Siete o más comunidades.



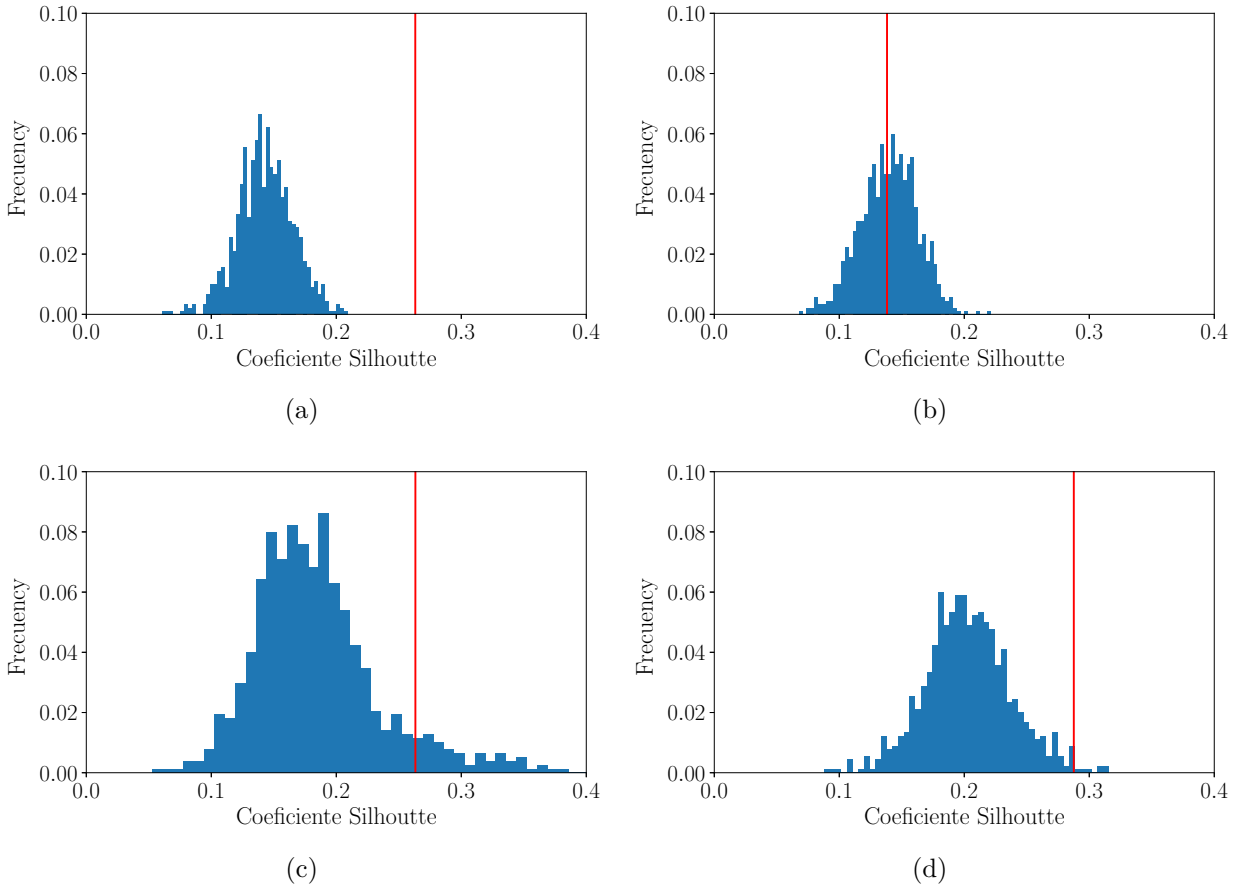
Silhoutte

Como se mencionó en la Introducción, otra manera de analizar las particiones de una red es mediante el índice Silhouette. De igual manera que en el caso de la modularidad (ver sección anterior) se estudiaron las particiones obtenidas usando cada uno de los algoritmos de clustering. En la Figura 7 se presentan los resultados obtenidos al estudiar Silhouette en la red original y en redes recableadas. Cabe destacar que se realizó el mismo procedimiento que el caso del análisis de la modularidad de las redes (1000 recableos, preservación de la distribución de grado, etc).

Como puede observarse en la Figura 7, en todos los algoritmos utilizados el histograma de Silhouette de las redes recableadas cumple una distribución simétrica, tal como ocurrió en el caso de la modularidad (ver Figura 5). Por otro lado, el valor de Silhouette de la red original (sin recablear) es en todos los casos, salvo en Fast Greedy, superior al valor medio esperado por azar.

Finalmente, al comparar los resultados obtenidos analizando la modularidad y el Silhoutte de la red, podemos concluir que la red original es modular debido a que los resultados obtenidos son significativamente mayores a los esperados por azar.

Figura 7. Histogramas de Silhoutte para distintos algoritmos de partición. Se realizaron 1000 iteraciones. En cada uno de las iteraciones se "recableó"(modificación de los enlaces de los nodos) la red un número de vece red original. La recta roja representa el valor de Silhoutte de la red original: (a) Algoritmo de Louvain; (b) Algoritmo Fast Greedy; (c) Algoritmo Infomap (una o más comunas); y, (d) Algoritmo Edge Betweennes.



Cantidad de comunidades

Con el objetivo de caracterizar las particiones, se analizó la cantidad de comunidades obtenidas mediante los diferentes algoritmos de clustering. Para ello, se estudió la cantidad de comunidades de la red original y se la comparó con el número de comunidades esperadas por azar. Al igual que en los casos anteriores, se realizaron 1000 procesos de recableo, en los cuales en cada uno de ellos el número de recableos fue igual a la

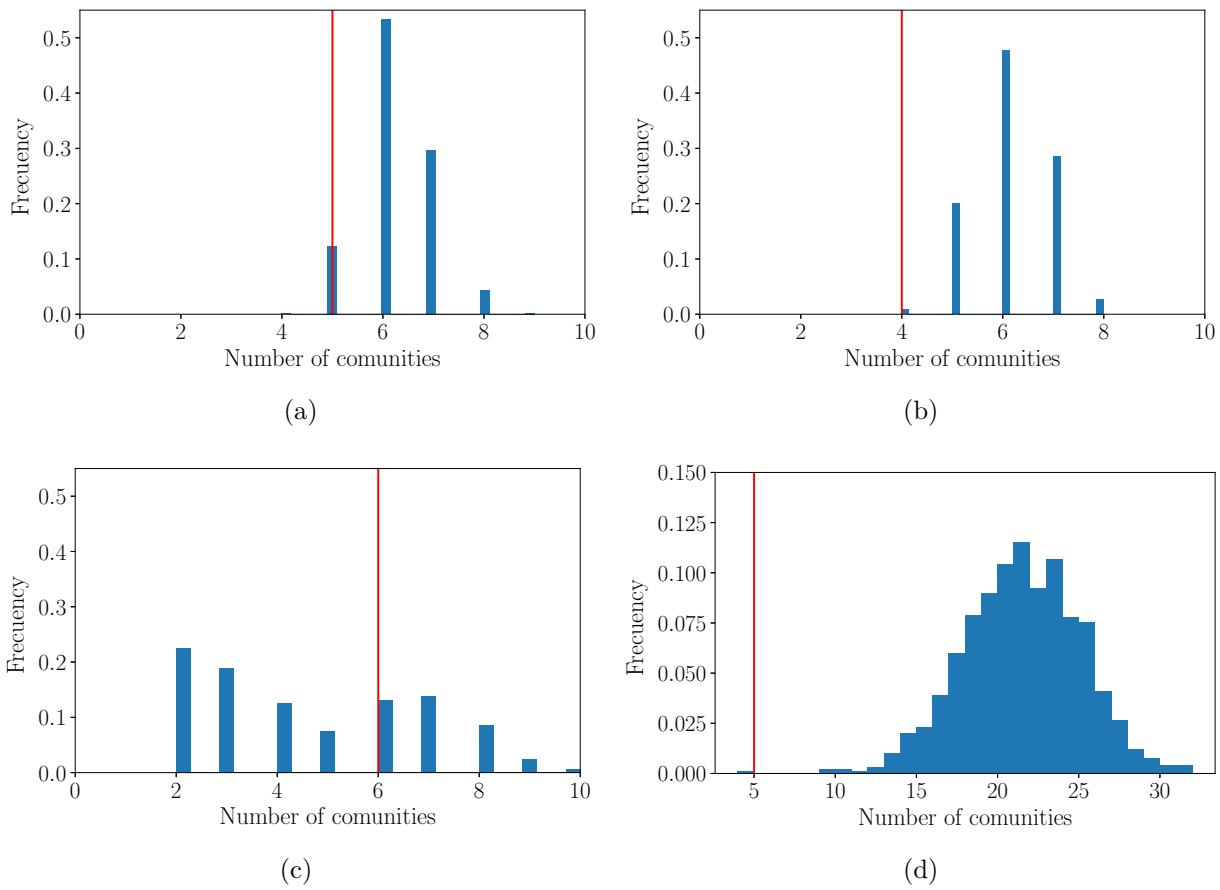
cantidad de enlaces de la red original.

En la Figura 8 se muestran los resultados obtenidos para cada uno de los algoritmos. En primer lugar, podemos observar que el número de comunidades de la red original (sin recablear) en cada uno de los algoritmos fue 4 (Fast Greedy) o 5 (Louvain, Infomap y Edge Betweennes). Notar que el número de comunidades dadas por Infomap no condice con el número de comunidades reportado en la Figura 4. Esto se debe al hecho que Infomap se basa en una caminata al azar y por lo tanto sus soluciones no son siempre las mismas.

En el caso de las redes recableadas, salvo en el caso de Infomap, se obtuvo una distribución simétrica. A su vez, cabe destacar que al utilizar el algoritmo de Edge Betweennes, la cantidad de comunidades obtenidas por azar fue cuatro veces mayor a la red original.

Por otro lado, en el caso de Infomap podemos observar que la mayoría de las redes recableadas son grandes. Es decir, que mediante este algoritmo se obtienen entre 2 y 3 comunidades. Recordemos que Infomap se basa en una caminata al azar dentro de la red. Por lo tanto, al no haber una estructura bien definida al recablear la red, las comunidades tienen gran tamaño. Por este motivo, el análisis de la modularidad en el caso de Infomap se realizó teniendo en cuenta solamente aquellos recableos en los cuales se obtuvieron siete o más comunidades (ver Figura 5).

Figura 8. Histogramas del número de particiones de la red para distintos algoritmos de partición. Se realizaron 1000 iteraciones. En cada una de los iteraciones se "recableó"(modificación de los enlaces de los nodos) la red un número de veces igual a la cantidad de enlaces de la red original. La recta roja representa la cantidad de comunas de la red original: (a) Algoritmo de Louvain; (b) Algoritmo Fast Greedy; (c) Algoritmo Infomap; y, (d) Algoritmo Edge Betweennes.

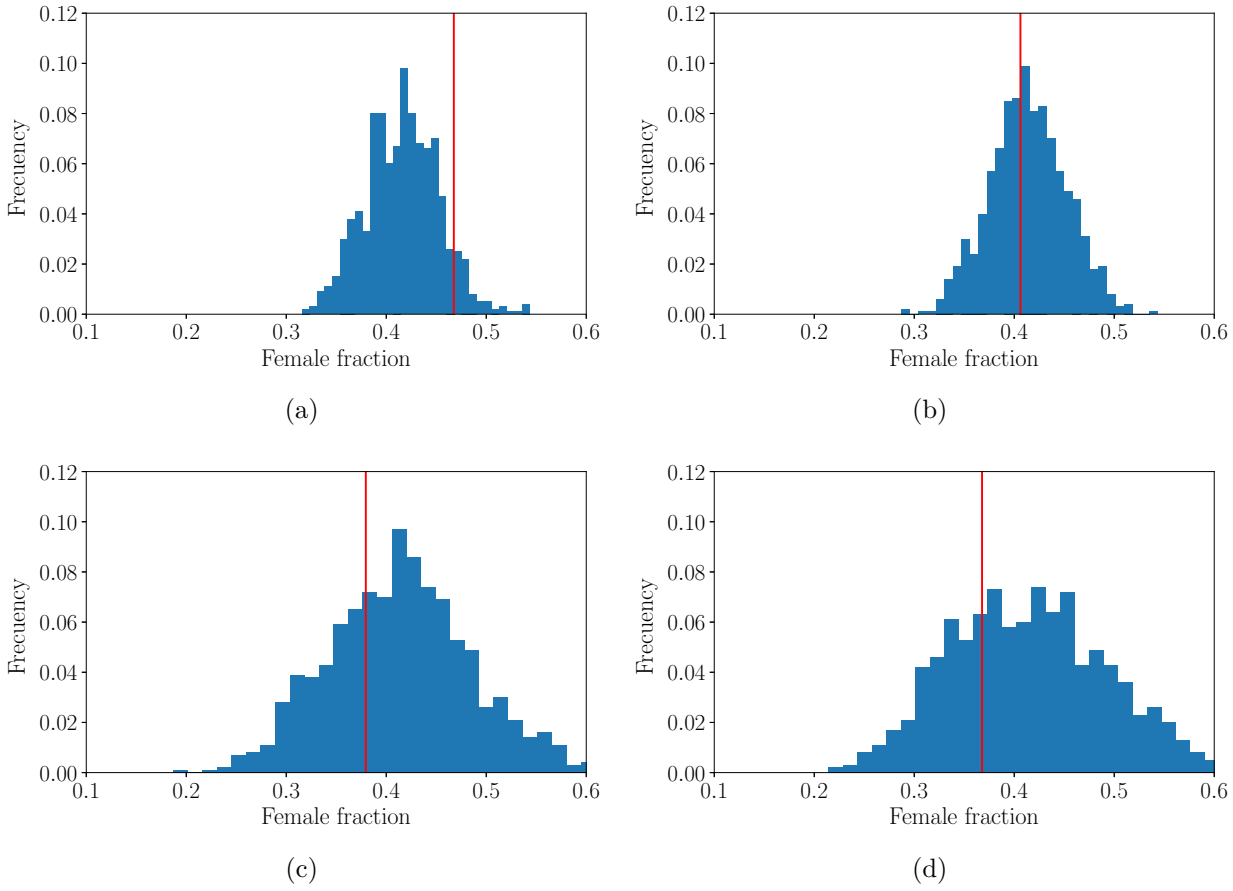


Comunidades y género de los delfines

Con el fin de saber si existe una relación entre el género de los delfines y la estructura de comunidades del grupo. Para ello, se calculó la fracción de delfines hembras en cada una de las comunidades. Este estudio se realizó sobre la red original y sobre 1000 redes sobre las cuales se realizó una proceso de permutación de los géneros de los delfines. Es decir, se mantuvo la estructura de la red (no se realizó ningún tipo de recableo) y se modificó de forma aleatoria el género de los delfines (manteniendo constante el número de hembras y machos). Cabe destacar que no se tuvieron en cuenta aquellos delfines sobre los cuales no fue posible identificar su género.

En la Figura 9 se presentan los resultados obtenidos para cada uno de los algoritmos de clustering. Se puede observar que en todos los casos, salvo en Louvain, no existe una correspondencia entre el género de los delfines y la estructura de comunidades. Es decir, que no es posible asociar una comunidad con delfines del mismo género. Sin embargo, en el primer trabajo realizado en el curso, se encontró que la red de delfines era homofílica. Es decir, que la mayoría de los enlaces entre delfines se da entre individuos del mismo género.

Figura 9. Histogramas de la fracción de hembras en las particiones para distintos algoritmos de partición. Se realizaron 1000 iteraciones sobre las cuales se realizaron "shuffles"(cambio aleatorio) del género de los delfines. La recta roja representa el valor de la fracción de hembras en las particiones de la red original: (a) Algoritmo de Louvain; (b) Algoritmo Fast Greedy; (c) Algoritmo Infomap (cinco o más particiones); y, (d) Algoritmo Edge Betweennes.



(a) (b)

(c) (d)

2.3. Acuerdo entre particiones

En esta sección mostramos el acuerdo entre particiones según la precisión (Tabla 1) y según Información mutua (Tabla 2). Notar que ambas "matrices"son simétricas pues, el acuerdo entre particiones debe ser recíproco. Edge betweenness e Infomap son los algoritmos cuyas particiones mostraron un mayor grado de acuerdo según el criterio de Información mutua. Es interesante destacar que este alto grado de acuerdo se verifica también para precisión. De hecho, los pares de algoritmos con alta precisión tienen también alto grado de información mutua mientras que los pares de algoritmos con baja precisión presentan baja información mutua. Al realizar el calculo de la Precisión e Información mutua con Infomap, dado su carácter aleatorio, fue necesario hacer el cálculo sobre varias realizaciones. Es por eso que los acuerdos entre particiones que involucran Infomap poseen incerteza.

Tabla 1. Acuerdo entre particiones por precisión.

Edge betweenness	0.980 ± 0.033	0.847	0.905
0.980 ± 0.033	**Infomap**	0.708 ± 0.001	0.999 ± 0.006
0.847	0.708 ± 0.001	**Fast greedy**	0.879
0.905	0.999 ± 0.006	0.879	**Louvain**

Tabla 2. Acuerdo entre particiones por Información mutua.

Edge betweenness	0.862 ± 0.054	0.680	0.784
0.862 ± 0.054	**Infomap**	0.794 ± 0.020	0.807 ± 0.042
0.680	0.794 ± 0.020	**Fast greedy**	0.746
0.784	0.807 ± 0.042	0.746	**Louvain**

Referencias

[1] Varios Autores. *Diccionario de la Real Academia Española*. Accedido el 27-10-2018. 2017. URL: http://dle.rae.es/srv/fetch?id=A5NKSVv.

[2] Albert-László Barabási. *Network Science*. Cambridge University Press, 2016. ISBN: 1107076269.

[3] Vincent D Blondel y col. "Fast unfolding of communities in large networks". En: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (oct. de 2008), P10008. DOI: 10.1088/1742-5468/2008/10/p10008. URL: https://doi.org/10.1088/1742-5468/2008/10/p10008.

[4] U. Brandes y col. "On Modularity Clustering". En: *IEEE Transactions on Knowledge and Data Engineering* 20.2 (feb. de 2008), págs. 172-188. DOI: 10.1109/tkde.2007.190689. URL: https://doi.org/10.1109/tkde.2007.190689.

[5] Aaron Clauset, M. E. J. Newman y Cristopher Moore. "Finding community structure in very large networks". En: *Physical Review E* 70.6 (dic. de 2004). DOI: 10.1103/physreve.70.066111. URL: https://doi.org/10.1103/physreve.70.066111.

[6] Santo Fortunato. "Community detection in graphs". En: *Physics Reports* 486.3-5 (feb. de 2010), págs. 75-174. DOI: 10.1016/j.physrep.2009.11.002. URL: https://doi.org/10.1016/j.physrep.2009.11.002.

[7] Linton C. Freeman. "A Set of Measures of Centrality Based on Betweenness". En: *Sociometry* 40.1 (mar. de 1977), pág. 35. DOI: 10.2307/3033543. URL: https://doi.org/10.2307/3033543.

[8] M. Girvan y M. E. J. Newman. "Community structure in social and biological networks". En: *Proceedings of the National Academy of Sciences* 99.12 (jun. de 2002), págs. 7821-7826. DOI: 10.1073/pnas.122653799. URL: https://doi.org/10.1073/pnas.122653799.

[9] David Huffman. "A Method for the Construction of Minimum-Redundancy Codes". En: *Proceedings of the IRE* 40.9 (sep. de 1952), págs. 1098-1101. DOI: 10.1109/jrproc.1952.273898. URL: https://doi.org/10.1109/jrproc.1952.273898.

[10] Bisma S. Khan y Muaz A. Niazi. *Network Community Detection: A Review and Visual Survey*. 2017. eprint: arXiv:1708.00977.

[11] M. E. J. Newman. "Fast algorithm for detecting community structure in networks". En: *Physical Review E* 69.6 (jun. de 2004). DOI: 10.1103/physreve.69.066133. URL: https://doi.org/10.1103/physreve.69.066133.

[12] M. E. J. Newman y M. Girvan. "Finding and evaluating community structure in networks". En: *Physical Review E* 69.2 (feb. de 2004). DOI: 10.1103/physreve.69.026113. URL: https://doi.org/10.1103/physreve.69.026113.

[13] M. Rosvall y C. T. Bergstrom. "Maps of random walks on complex networks reveal community structure". En: *Proceedings of the National Academy of Sciences* 105.4 (ene. de 2008), págs. 1118-1123. DOI: 10.1073/pnas.0706851105. URL: https://doi.org/10.1073/pnas.0706851105.

[14] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". En: *Journal of Computational and Applied Mathematics* 20 (nov. de 1987), págs. 53-65. DOI: 10.1016/0377-0427(87)90125-7. URL: https://doi.org/10.1016/0377-0427(87)90125-7.