Trabajo computacional 3: Detección de comunidades

Ferreira Chase, T., Leizerovich, M., Goren, G.

7 de noviembre de 2018

Resumen

En este trabajo se estudió la estructura de comunidades de una red de interacciones sociales entre delfines "nariz de botella" (género *Tursiops* del fiordo de Doubtful Sound) en Nueva Zelanda, presentada por primera vez en [1]. Se comparan las comunidades obtenidas mediante siete métodos de detección de comunidades diferentes, y se evaluaron en términos de su modularidad y sus *silhouettes*, comparando lo obtenido con un *ensemble* de recableos aleatorios de la red original. El acuerdo entre métodos fue analizado mediante la información mutua entre métodos y mediante un análisis basado en matrices de confusión. Finalmente, se estudió la correlación entre estas comunidades y la división por sexo de los delfines, observándose la presencia de una comunidad de machos bien definida y una comunidad de hembras menos clara, más vinculada con delfines macho.

1. Introducción

Es sabido que los delfines poseen actividad social. En particular, en el año 2005 Lusseau publicó un estudio de tres años de duración sobre la vida social de una población de 62 delfines del género *Tursiops* (conocidos como "delfines nariz de botella") que habitaba las aguas de Doubtful Sound, Nueva Zelanda [1]. En él, presentó un modelo de la misma bajo el formato de red, definiendo que dos delfines están conectados por un enlace si fueron vistos juntos más seguido que lo esperado por azar. En el presente trabajo, se estudió la estructura de comunidades de dicha red, relacionándolo con la información disponible sobre el sexo de los individuos.

Una representación gráfica de la red puede verse en la Figura 1, en la cual se señala a los delfines hembra (24 individuos) con color celeste y a los delfines macho (34 individuos) con color rojo. Los delfines para los cuales no se dispone información sobre su sexo (4 individuos) se marcan con verde.

Para el análisis subsiguiente, se llamará *partición* a una colección de conjuntos de nodos, los cuales llamaremos comunidades o *clusters*, tal que todos los conjuntos sean disjuntos de a pares y su unión sea el conjunto de nodos total de la red.

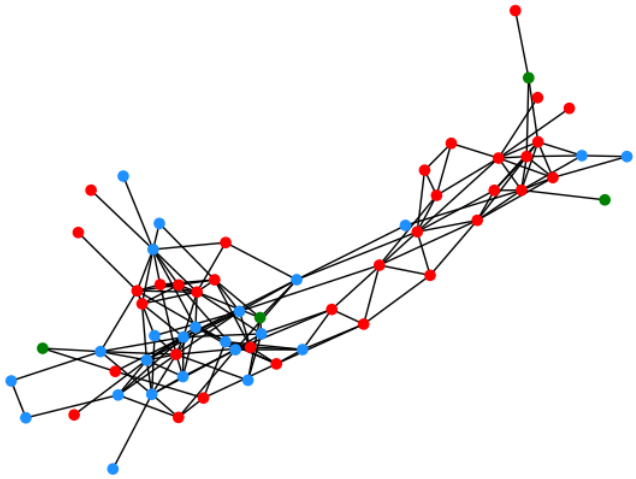


Figura 1: Red de delfines coloreada según el sexo de los mismos. Las hembras y los machos son representados por nodos celestes y rojos respectivamente, mientras que los indefinidos son representados por nodos verdes. La disposición gráfica de los nodos se determinó mediante un posicionamiento por fuerzas elásticas (algoritmo de Fruchterman–Reingold).

2. Métodos de detección de comunidades y particiones obtenidas

Se particionó la red utilizando 7 métodos distintos, los cuales denominaremos según *Infomap*, *Label propagation*, *Fastgreedy*, *Eigenvector*, *Louvain*, *Edge betweenness* y *Walktrap*. En la Figura 2 se presentan los coloreados de la red obtenidos mediante cada uno de ellos. En las secciones subsiguientes haremos un análisis detallado de los métodos de detección de comunidades utilizados y la consecuencia de su aplicación a la red de delfines. Las implementaciones de los métodos empleadas fueron las del paquete `igraph` para Python.

Junto con la visualización de cada partición mediante un coloreado del grafo, se presentan las *silhouettes* de cada cluster. El método de *silhouettes*, introducido en 1986 por Rousseeuw [2], consiste en asignarle a cada nodo i un coeficiente de *silhouette* $s(i)$, y graficar los valores de estos coeficientes para todos los nodos de un mismo cluster, ordenados de menor a mayor. La "silueta" obtenida es un indicador de qué tan bien responde el cluster a los criterios de *compacidad* (interconexionado denso dentro del cluster) y *separación* (los nodos del cluster están más cerca entre sí que de nodos de otros clusters).

El coeficiente de *silhouette* para el nodo i se define como $s(i) = (b(i) - a(i))/\operatorname{máx}(a(i), b(i))$ donde $a(i)$ es la distancia promedio del nodo a todos los demás nodos de su cluster y $b(i)$ es el mínimo, tomado sobre todos los clusters a los que i *no* pertence, de las distancias promedio del nodo i a los nodos de cada uno de dichos clusters. En notación matemática, si simbolizamos por $\{C_k\}_k$ a la colección de comunidades y por $|A|$ al número de elementos del conjunto A,

$$a(i) = \frac{1}{|C_i|} \sum_{j \in C_i, j \neq i} d(i,j)$$

$$b(i) = \min_{k: i \notin C_k} \left\{ \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j) \right\}$$

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

y convencionalmente $s(i) = 0$ si el cluster a que pertenece el nodo i es $\{i\}$, i.e. se trata de un cluster con un único elemento. Como puede verse, $-1 \leq s(i) \leq 1$ siempre, y un valor negativo de $s(i)$ indica que el nodo se encuentra más cerca de otros clusters que del cual le ha sido asignado. Si bien la curva de *silhouette* completa para cada cluster da información detallada sobre la adecuación de *cada* nodo al criterio de compacidad/separación, para comparar particiones es usual tomar el valor promedio sobre toda la red $\bar{s} = N^{-1} \sum_i s(i)$ donde N es el número de nodos total.

Es importante señalar que el coeficiente de *silhouette* está definido únicamente para grafos conexos. Si bien nuestro grafo de partida lo es, esto será relevante a la hora de realizar recableos aleatorios del mismo.

Por otro lado, otra magnitud importante a la hora de evaluar las particiones (y también de producirlas, en el caso de ciertos algoritmos) es la *modularidad* Q de una red, que se define como

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

donde A es la matriz de adyacencia de la red, k_i es el grado del i-ésimo nodo y c_i es la comunidad a la que pertenece (la notación $\delta(c_i, c_j)$ debe interpretarse como igual a 1 si los nodos i y j pertenecen a la misma comunidad, y 0 en caso contrario). Esta magnitud, introducida por Newman et al. en 2004 [5] mide cuántos enlaces intra-comunidades hay en la red y compara esta magnitud (mediante una resta) con la que cabría esperar por azar. Está normalizada de forma tal que $-1 \leq Q \leq 1$ siempre, es positiva si hay más enlaces intra-comunidades que lo que cabría esperar por azar y negativa en caso contrario. Cabe mencionar, como caso degenerado, que $Q = 0$ cuando la partición consiste en una única comunidad.

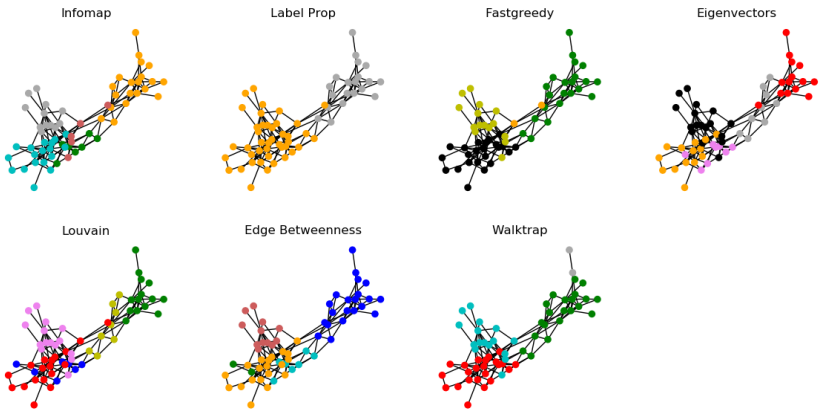


Figura 2: Red de delfines coloreada según los distintos métodos de detección de comunidades empleados.

2.1. Infomap

El método *Infomap*, así llamado por sus autores, fue presentado en 2009 por Rosvall y Bergstrom[3]. La idea general consiste en generar caminatas al azar a lo largo del grafo, las cuales son consideradas como secuencias finitas de nodos, y buscar la partición que minimice la longitud de descripción de las mismas, usando una codificación Huffman de dos niveles (un código indica las transiciones entre comunidades a lo largo de la caminata, mientras que un segundo código indexa a cada nodo dentro de una misma comunidad).

Como se observa en la Figura 3, este método arrojó una partición compuesta por 5 clusters diferentes. Uno de ellos (el amarillo) parece comprender a la mayoría delfines macho, y es el que posee coeficientes de *silhouette* más grandes. El cluster celeste presenta varios nodos con un valor de s negativo, indicando que no satisface el criterio de compacidad/separación sino que se encuentra amalgamado con los otros tres clusters cercanos, los cuales son marcadamente más compactos.

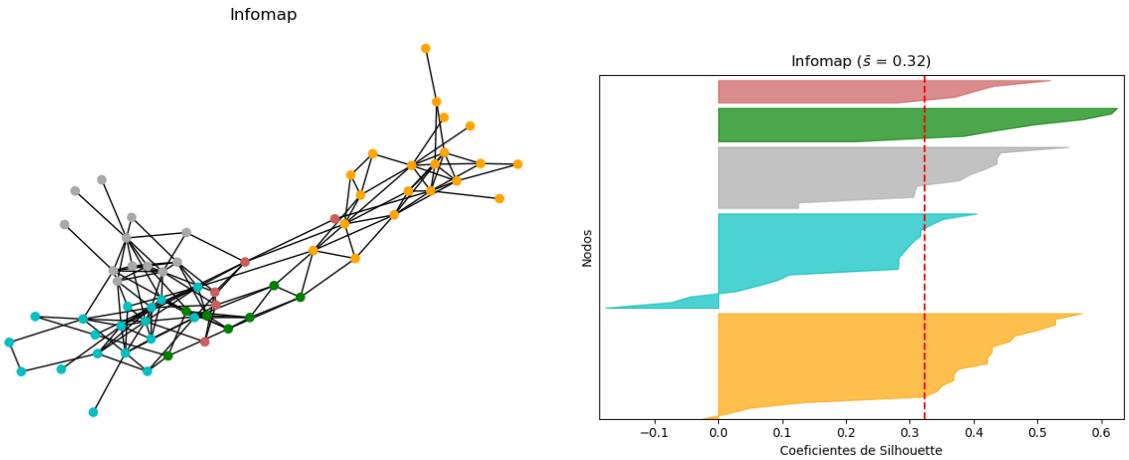


Figura 3: Red de delfines coloreada según método *Infomap*, y *silhouettes* de los clusters obtenidos. En línea punteada se marca el valor de *silhouette* promedio $\bar{s} = 0{,}32$.

2.2. Propagación de etiquetas (*label prop*)

El método de propagación de etiquetas empleado (*label prop*) fue presentado por Raghavan et al. en el año 2007 y se basa en alcanzar el régimen estacionario de un proceso difusivo definido sobre el grafo [4]. Comienza asignándole una etiqueta única a cada nodo, y en cada paso se reasigna la etiqueta de cada nodo como la etiqueta más frecuente entre sus vecinos. Este proceso es convergente, y se toma como partición a los clusters dados por cada una de las etiquetas.

Una particularidad del método es que, parafraseando a los autores, 'En el caso de redes homogéneas tales como grafos aleatorios de Erdós-Rényi que no tienen estructura de comunidades, el algoritmo de propagación de etiquetas identifica la componente gigante de estos grafos como una comunidad única' [4].

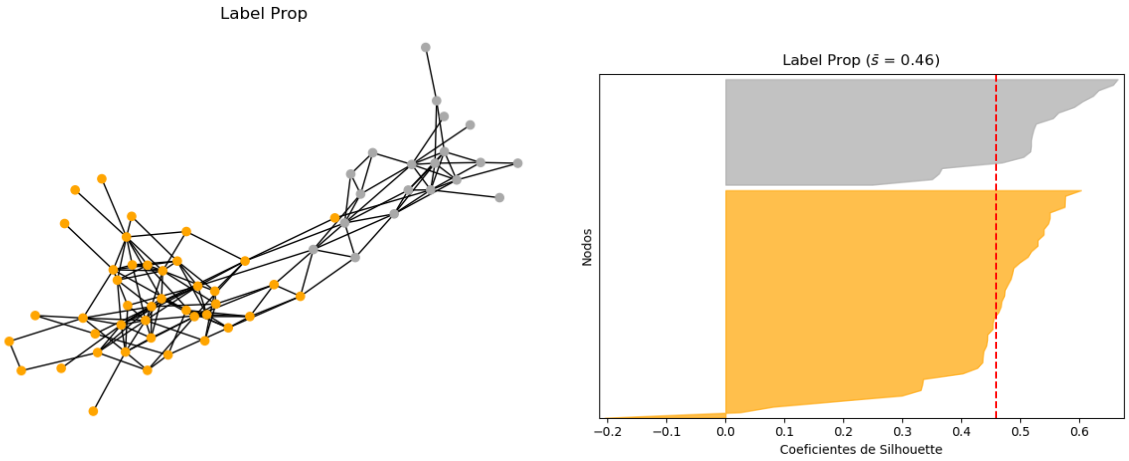


Figura 4: Red de delfines coloreada según método *label prop*, y *silhouettes* de los clusters obtenidos. En línea punteada se marca el valor de *silhouette* promedio $\bar{s} = 0{,}46$.

En la figura 4, a diferencia del resto de los métodos, se observan solo dos comunidades relativamente grandes, las cuales corresponden a los cúmulos de nodos tal como los dibuja el algoritmo de posicionamiento por fuerzas elásticas. Podemos inferir de allí que, en general, los valores de *silhouette* serán elevados salvo por un pequeño grupo en la frontera entre ambos. Además, las *silhouettes* resaltan el hecho de que el algoritmo "prefirió" agregar varios nodos puente, que conectan ambos clusters y por lo tanto necesariamente tendrán un valor bajo de *silhouette*, al cluster amarillo.

2.3. *Fast Greedy*

Por *Fast Greedy* nos referimos al algoritmo propuesto por Newman et al. en el mismo trabajo que en el cual introdujeron el concepto de modularidad, basado en una optimización rápida y *codiciosa* (en el sentido que se le da en el área de algoritmos) de esta magnitud [5]. Primeramente, se asigna cada nodo a una comunidad diferente. Luego, para cada par de comunidades conectadas mediante al menos un enlace se estima la variación en la modularidad ΔQ que se produciría si ambas fueran combinadas. Si $\Delta Q > 0$ entonces se fusionan las comunidades. Esta nueva partición se guarda en memoria y se calcula la modularidad de la nueva red. Este proceso se repite hasta que quede una única comunidad. Finalmente, para obtener la partición de la red definitiva, se examinan todas las particiones almacenadas y se selecciona la que maximiza Q (a partir de cierto momento, todas las variaciones ΔQ comienzan a ser negativas, por lo cual hay un único pico de Q a lo largo del procedimiento).

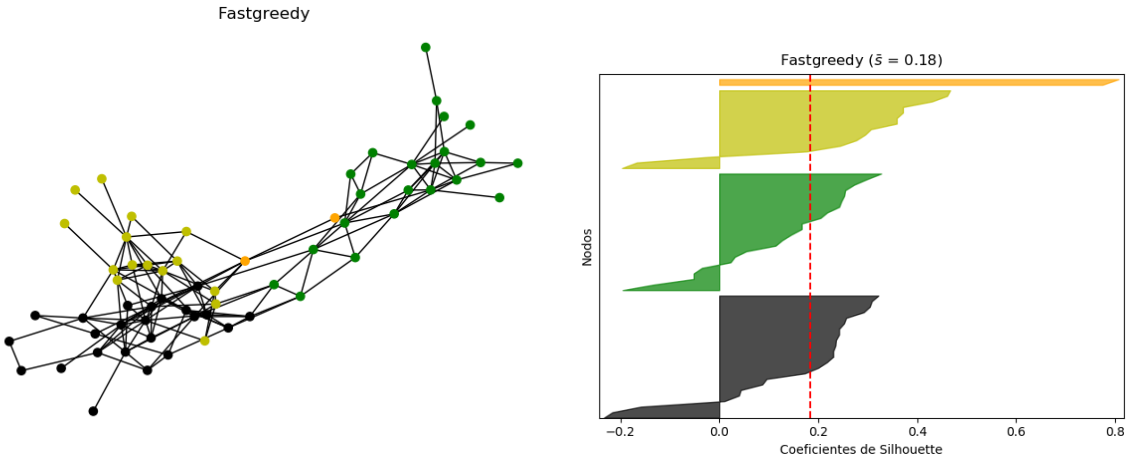


Figura 5: Red de delfines coloreada según método *Fast Greedy*, y *silhouettes* de los clusters obtenidos. En línea punteada se marca el valor de *silhouette* promedio $\bar{s} = 0{,}18$.

Como se puede ver en la Figura 5, en este caso se detectaron 4 comunidades, de las cuales una tiene solamente 2 nodos. Es interesante observar que, en general, para comunidades extremadamente chicas es posible obtener valores muy altos de *silhouette* sin que eso represente una característica especial de las mismas. El caso extremo y trivial es el de una comunidad de 2 nodos, puesto que efectivamente se trata de una *clique* o subgrafo completo (lo cual no es de ninguna manera interesante). El resto de las comunidades muestran un subconjunto de nodos con $s < 0$. Además de la comunidad de arriba a la derecha, la cual es detectada en todos los casos, se detecta también una división en dos comunidades del otro sector de la red.

2.4. *Eigenvectors*

El método del autovector principal, denominado aquí como *Eigenvectors* e introducido por Newman en el 2006 [6], consiste en encontrar la partición que maximice la modularidad de la red a partir del análisis espectral de la matriz de modularidades, la cual se define en relación a la Ecuación 1 como $Q_ij = A_{ij} - k_i \frac{k_j}{2m}$ donde A es la matriz de adyacencia y k_i es el grado del i-ésimo nodo.

El método se basa en encontrar divisiones binarias de la red, y luego aplicar múltiples veces el procedimiento. Se puede demostrar que la modularidad de la red está dada por $Q = \frac{1}{4m} \sum_i a_i^2 \beta_i$, donde m es el número de enlaces, $a_i = \boldsymbol{u}^i \cdot \boldsymbol{s}$ es el producto interno de $\boldsymbol{u}^i$, el i-ésimo autovector de la matriz de modularidad, con el *vector indicial* s, el cual define a cuál de las dos comunidades resultantes corresponderá cada nodo, y es tal que $s_j = \pm 1 \forall j$ (el signo indica la comunidad); y β_i es el autovalor correspondiente al autovector $\boldsymbol{u}^i$. Para intentar maximizar la modularidad de la red, se elige el vector s que privilegie en la suma al autovalor más grande, lo cual se obtiene eligiendo $s_i = 1$ cuando $\boldsymbol{u}_i^{(1)} \geq 0$ y $s_i = -1$ cuando $\boldsymbol{u}_i^{(1)} < 0$, donde $\boldsymbol{u}_i^{(1)} > 0$.

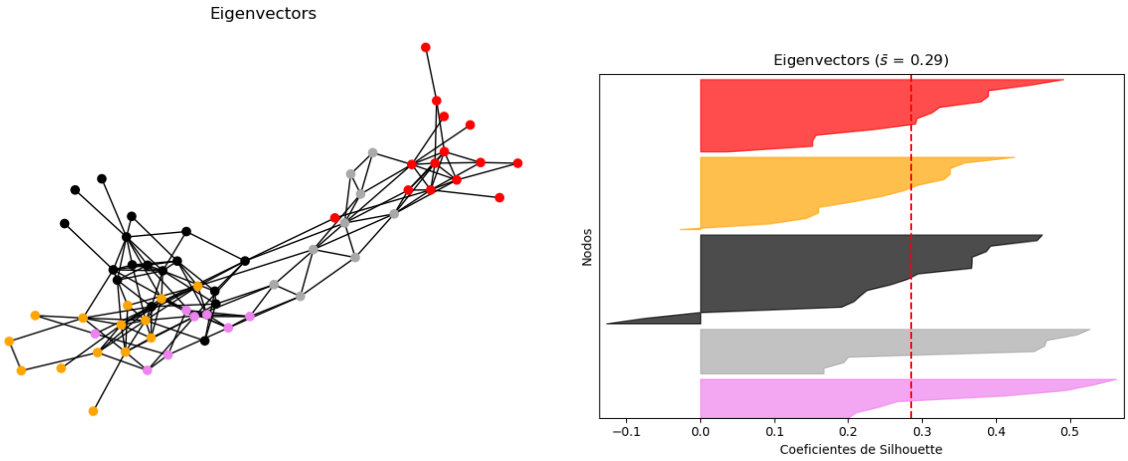


Figura 6: Red de delfines coloreada según método *Eigenvectors*, y *silhouettes* de los clusters obtenidos. En línea punteada se marca el valor de *silhouette* promedio $\bar{s} = 0{,}29$.

En este caso se obtienen 5 comunidades, ninguna de las cuales sigue una división tan obvia a partir de la visualización de la red como las de los métodos precedentes (ver Figura 6). Un único cluster presenta valores negativos de *silhouette* apreciables.

2.5. Louvain

El método que se popularizó bajo el nombre *Louvain* fue introducido por Blondel et al. investigadores de la universidad homónima, en una publicación del 2008 [7]. Al igual que los dos métodos anteriores, es un método de reconocimiento de comunidades a partir de la maximización de la modularidad, pero esta vez en dos pasos. Primero, realizando cambios locales en cada nodo de la red, se intenta maximizar la modularidad tratando de unir un nodo con sus vecinos. Luego se arma una nueva red (técnicamente, un multigrafo) donde cada nodo es una comunidad del paso anterior. Ambos pasos se repiten sistemáticamente hasta que la modularidad deja de aumentar.

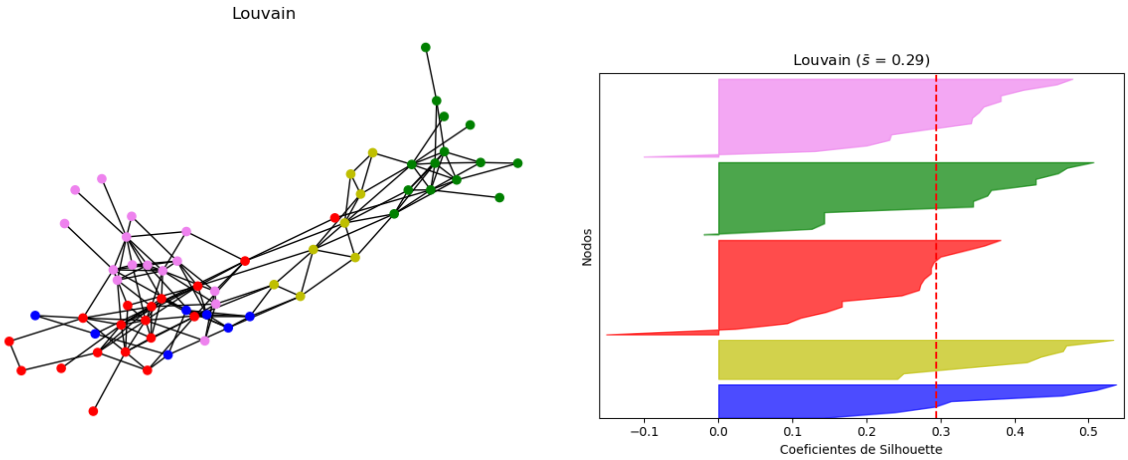


Figura 7: Red de delfines coloreada según método *Louvain*, y *silhouettes* de los clusters obtenidos. En línea punteada se marca el valor de *silhouette* promedio $\bar{s} = 0{,}29$.

También se detectaron 5 comunidades, las cuales guardan una gran similaridad con las detectadas mediante el método *Eigenvectors*. A través de la visualización del grafo, puede observarse que no son exactamente iguales.

Sin embargo, las *silhouettes* de las comunidades detectadas son notablemente parecidas, y sus valores medios de *silhouette* coinciden dentro de las primeras dos cifras decimales.

2.6. Edge Betweenness

La idea del algoritmo de Newmann-Girvan, llamado así por los autores del artículo de 2004 que lo introduce y aquí denominado *Edge Betweenness*, es partir de un cluster gigante de una única componente y eliminar paso a paso los enlaces que sean más importantes para el mantenimiento de la topología de la red, cualidad denominada comúnmente *centralidad*. En este caso se define la centralidad para un enlace como su 'intermediatez' (*betweenness* en inglés), la cual es mayor mientras más caminos cortos entre nodos atraviesen el enlace. Primero, se computa la centralidad de todos los enlaces de la red. Luego se remueve el enlace con mayor centralidad y se recalcula la centralidad de los enlaces que sobreviven. Este proceso se repite iterativamente hasta que no queden enlaces en la red. Este proceso da lugar a una estructura jerárquica, la cual se visualiza mediante un *dendrograma*, y que en cada nivel del mismo presenta una posible manera de particionar el grafo. El criterio para 'cortar' el dendrograma y definir la partición que devuelve el algoritmo es la maximización de la modularidad Q.

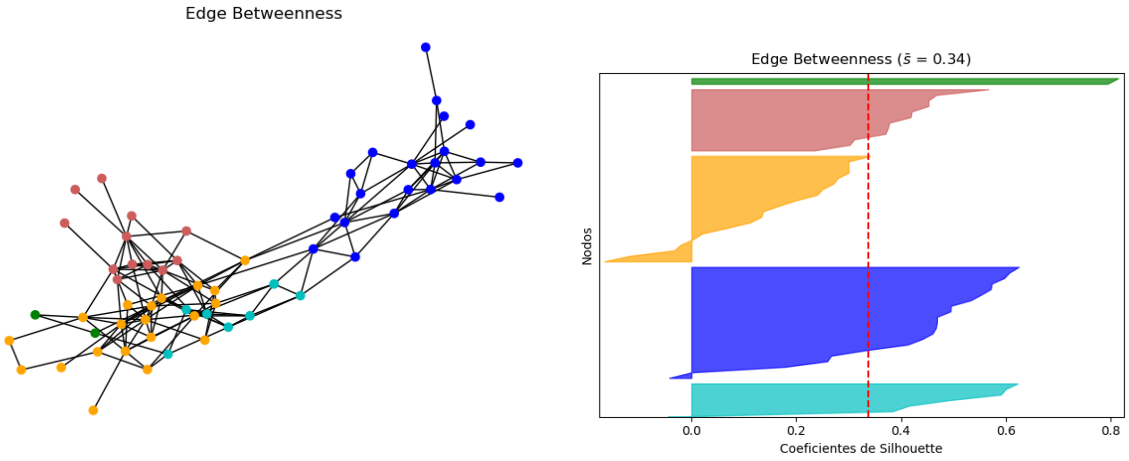


Figura 8: Red de delfines coloreada según método *Edge betweenness*, y *silhouettes* de los clusters obtenidos. En línea punteada se marca el valor de *silhouette* promedio $\bar{s} = 0{,}34$.

Este método detectó 5 comunidades, una de las cuales tiene solo 2 nodos. Nuevamente se observa una comunidad "puente", con valores negativos de *silhouette*, separando a otras dos comunidades dentro del cúmulo de la izquierda, y una única comunidad para el cúmulo de la derecha.

2.7. Walktrap

Por último, el método *Walktrap*, introducido por Pons y Latapy en 2005, se basa en la idea de que un caminante aleatorio que da una cantidad pequeña de pasos tiende a quedarse atrapado en las partes más densas de la red. Se generan caminatas aleatorias de pequeña cantidad de pasos, mediante las cuales se estiman las probabilidades P_{ij}^t de llegar de i a j en una cantidad t de pasos, y se emplean estas probabilidades para definir una métrica sobre los nodos de la red. A su vez, esta métrica se emplea para definir una distancia entre clusters. Una vez definida esta última, se procede a realizar un proceso de agrupamiento jerárquico. Se parte de asignar a cada nodo su propia comunidad, y luego en cada paso se combinan dos comunidades elegidas mediante un criterio que se en la noción de distancia definida previamente. El proceso, opuesto en cierto sentido al del

método precedente, da lugar a un dendrograma y el corte se elige maximizando la modularidad.

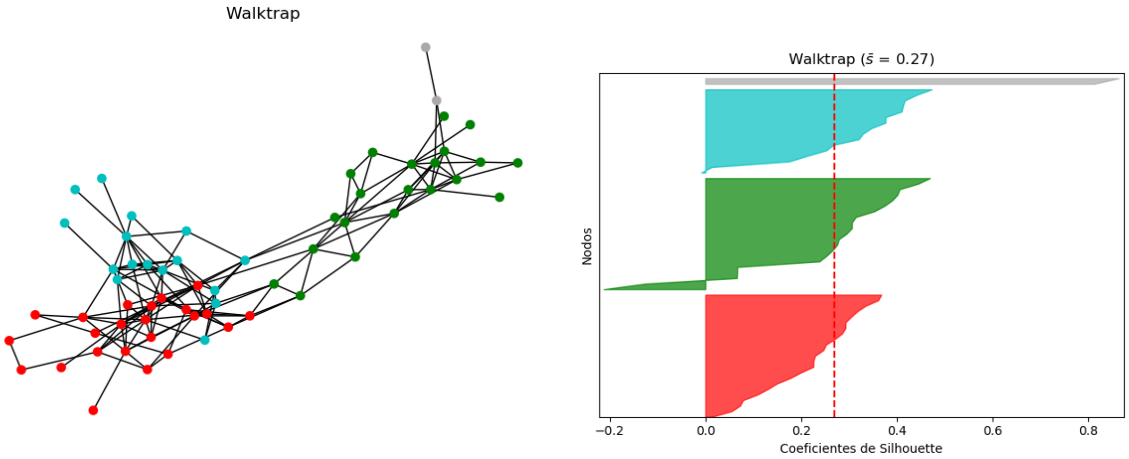


Figura 9: Red de delfines coloreada según método *Walktrap*, y *silhouettes* de los clusters obtenidos. En línea punteada se marca el valor de *silhouette* promedio $\bar{s} = 0{,}27$.

En nuestros datos, este método detectó 4 comunidades (ver Figura 9), siendo una de ellas un par de nodos del cúmulo de la derecha relativamente lejanos al resto de los nodos cercanos. Las otras 3 comunidades reproducen el patrón encontrado mayoritariamente, que consiste en una comunidad en el cúmulo de la derecha y dos comunidades en el de la izquierda.

3. Análisis por recableo

Para cada método de partición utilizado, además de $\bar{s}$ se calculó la modularidad de la red. Para poder estimar si los valores obtenidos son atribuibles a una estructura de modularidad en los datos, se procedió a comparar la red con un modelo nulo. Con este objetivo, se realizaron 10000 recableos diferentes para la red original. Los recableos se realizaron mediante *enroques*, que consisten en seleccionar al azar dos enlaces que no compartan nodos, y permutar los enlaces dentro del grupo de 4 nodos resultante (quitar los enlaces existentes y agregar enlaces donde no los había). Debido a que el análisis de silhouettes no tiene sentido para redes disconexas, y a que los métodos de detección de comunidades suelen aplicarse solo sobre componentes conexas, para obtener cada recableado se realizaron por lo menos 200 *enroques* y se continuó hasta que el grafo resultante fuese conexo.

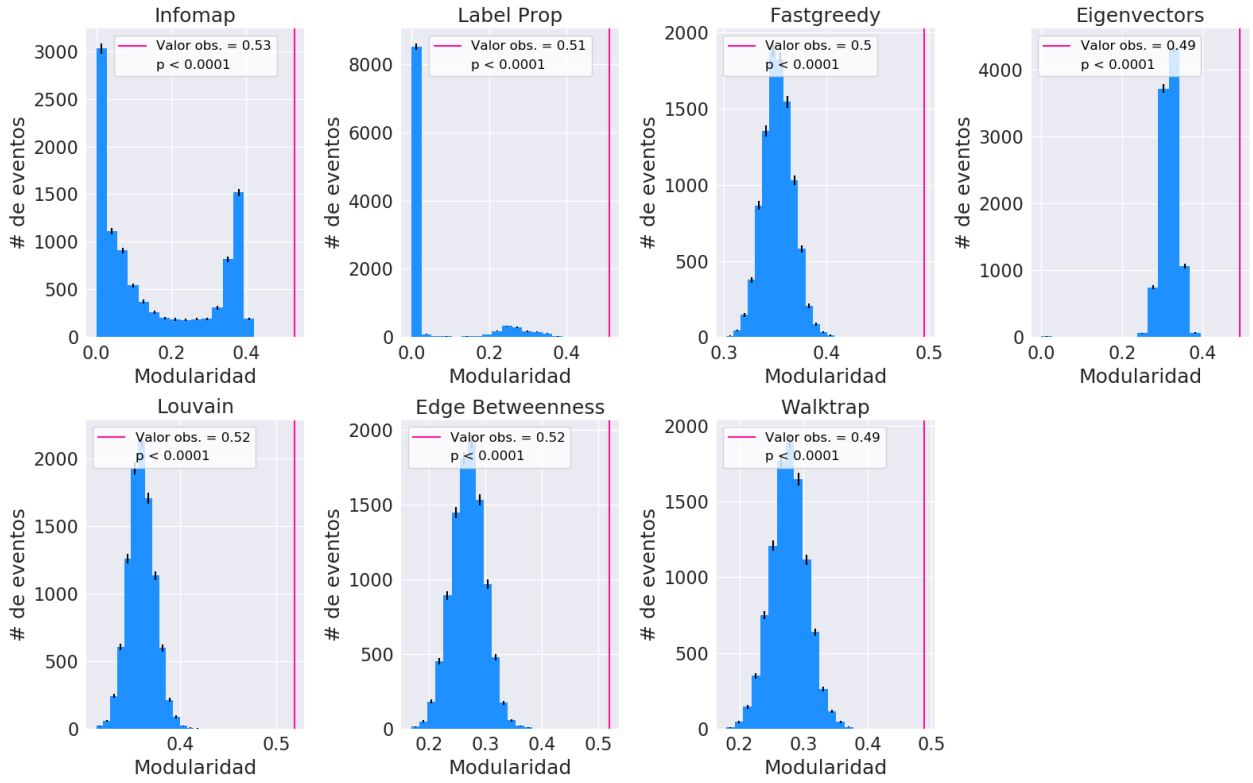


Figura 10: Serie de histogramas de modularidad correspondientes a redes recableadas a partir de la red de delfines original. Se compara el histograma con el valor de modularidad de la red original y se especifica el p_{valor}. Se muestra este análisis para cada método de partición estudiado.

En la Figura 10, se muestran los histogramas obtenidos mediante recableo para la hipótesis nula, y las líneas verticales señalan la posición de los valores de modularidades observados para cada método. para todos los métodos los p-valores de los histogramas son menores a 10^{-4}, por lo que podemos concluir que estamos en presencia de una red modular. Es interesante notar que tanto para *Infomap* como para *Label propagation* existe un pico para modularidad igual a cero. Esto se explica, tal como se mencionó antes, por el hecho de que ambos tienden a detectar particiones de un solo cluster cuando la estructura de la red no es realmente modular.

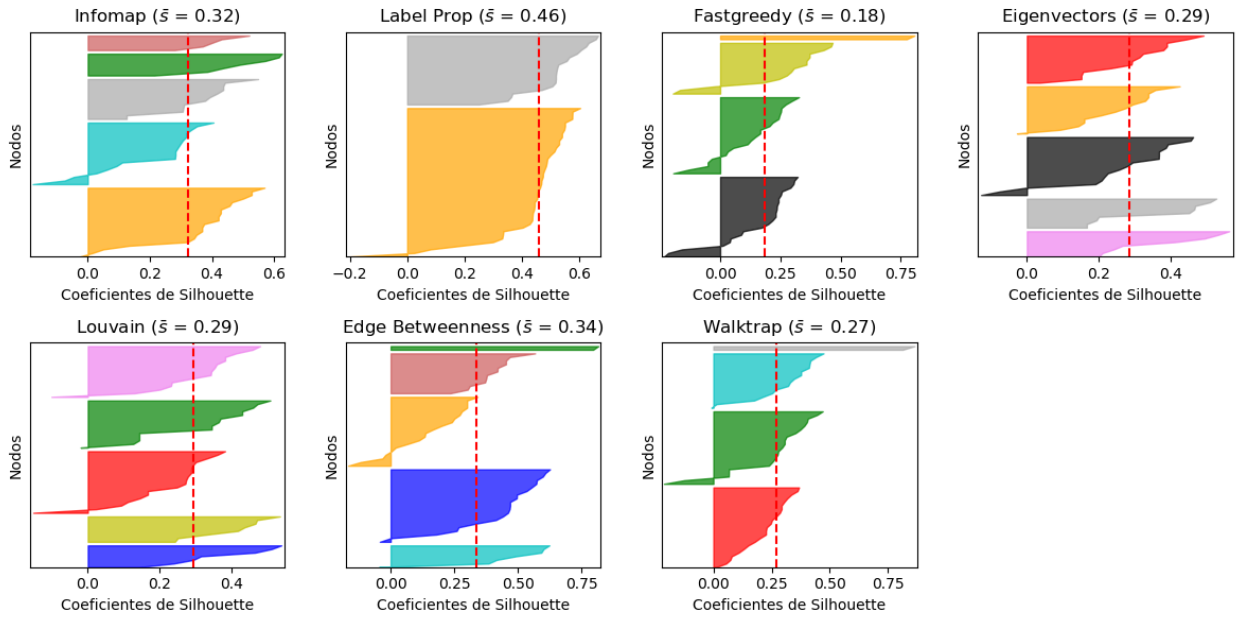


Figura 11: Gráficos de *silhuette* para todos los métodos de partición.

Respecto del análisis de las curvas de *silhouette*, en la Figura 11 se muestran todas las curvas obtenidas para los distintos métodos. *Label propagation* es el método que maximiza los valores de silhouettes más grandes y minimiza los valores de silhouettes negativos, obteniéndose el valor medio $\bar{s}$ más grande, por lo que resultaría la mejor partición según este criterio. Sin embargo es importante recordar que el análisis de *silhouettes* no cuantifica necesariamente la bondad de las comunidades sino únicamente su acuerdo con el criterio de compacidad/separación. Es natural entonces que, al comparar entre métodos distintos, tenga mejores valores de *silhouette* aquel que tienda a encontrar una menor cantidad de clusters.

En la Figura 12 se grafican los histogramas de $\bar{s}$ obtenidos mediante recableo, En este caso, dado que el valor $s(i)$ no está definido unívocamente para particiones de un único cluster, se debió eliminar del análisis dichas particiones. Esto implicó descartar el 14 % de los recableos para *Infomap* y el 78 % de los recableos para *Label prop*, razón por la cual se omitió a este método del análisis.

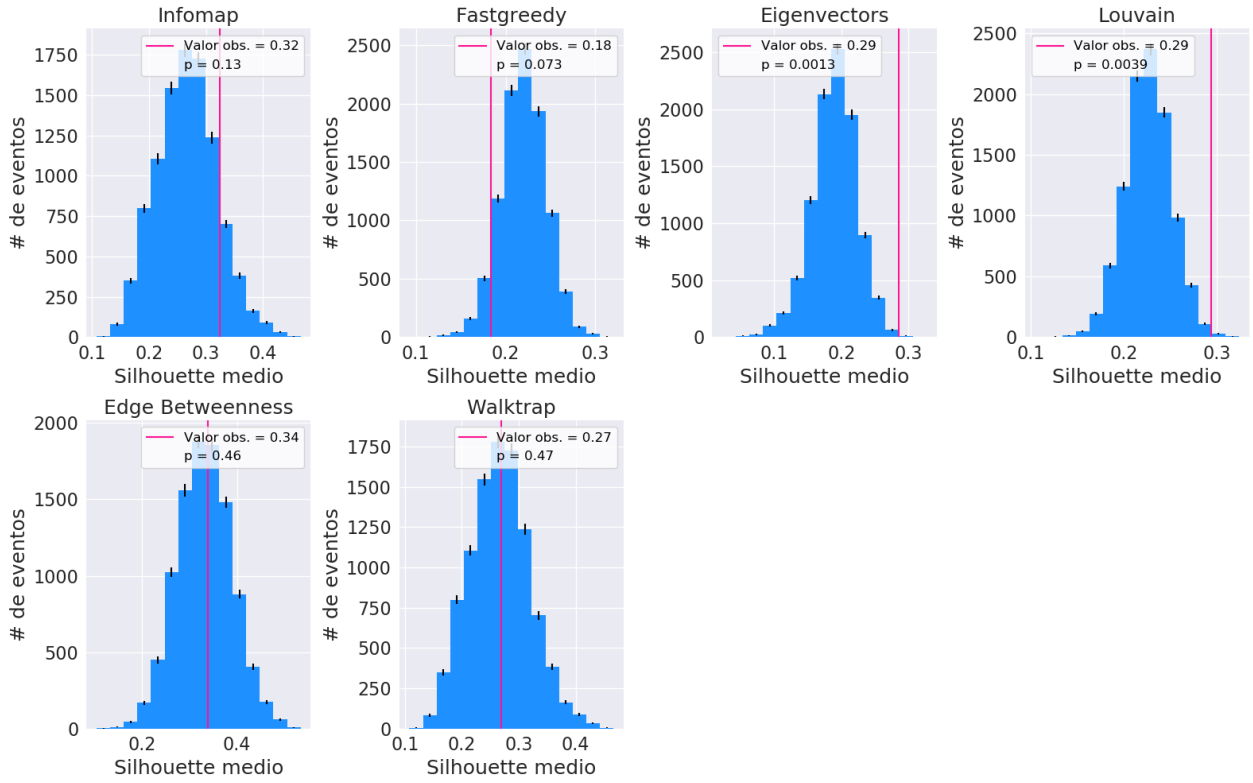


Figura 12: Serie de histogramas de silhouette promedio correspondientes a redes recableadas a partir de la red de delfines original. Se compara el histograma con el valor de modularidad de la red original y se especifica el p_{valor}. Se muestra este análisis para cada método de partición estudiado.

En este caso, se obtienen distribuciones cuyos valores medios no distan tanto de los valores observados. Si bien se observa que los métodos *Fastgreedy*, *Eigenvector* y *Louvain* poseen un $p_{valor} < 0{,}1$, la cola de la distribución en la que cae el valor observado no es consistente entre los distintos métodos, con lo cual tampoco se puede concluir algo significativo respecto de la red original en estos casos. Vemos que los valores de *silhouettes* son mucho más dependientes del método que los de modularidad y no hablan directamente de la estructura de la red; en todo caso, las distribuciones observadas nos hablan más bien de cómo responde cada algoritmo ante casos patológicos, en los cuales no hay ninguna comunidad que pueda ser encontrada.

4. Acuerdo entre métodos

En esta sección, analizamos qué tanto acuerdo hay entre las particiones encontradas mediante dos métodos, uno basado en teoría de la información y otro basado en un simple conteo de cuántas veces las particiones coinciden respecto a una decisión de clasificación específica.

4.1. Información mutua

La información mutua es un observable que puede utilizarse para cuantificar qué tan independientes son dos distribuciones de probabilidad a partir de datos experimentales. Está dada por

$$I_M^* = \sum_{C_1,C_2} p(C_1,C_2) \log\left(\frac{p(C_1,C_2)}{p(C_1).p(C_2)}\right) \tag{2}$$

donde $p(C_1)$ es la probabilidad de que un nodo elegido de manera uniformemente aleatoria pertenezca al cluster C_1 de una cierta partición, mientras que $p(C_2)$ es el equivalente correspondiente a una segunda partición, la cual es comparada con la primera; y la notación $p(C_1, C_2)$ corresponde a la distribución de probabilidad conjunta para ambas particiones. Dado que nos interesa comparar el acuerdo entre distintos pares de particiones, elegimos trabajar con la información mutua normalizada

$$I_M = \frac{I_M^*}{\frac{1}{2}(H(p(C_1)) + H(p(C_2))} \tag{3}$$

donde $H(q)$ es la entropía de la distribución de probabilidad q, dada por $H(q) = \sum_x q(x) \log(q(x))$. Mediante el uso de esta cantidad, vamos a estudiar si las particiones generadas por un método son independientes de otro método, o si se generan particiones similares.

Las probabilidades conjuntas involucradas se obtienen experimentalmente como las fracciones de nodos que pertenecen simultáneamente a cada par de clusters. La distribución experimental resultante corresponde a un histograma 2D donde se suma una unidad al casillero ij cuando un nodo pertenece a C_i mediante la primera partición, y a C_j mediante la segunda. Finalmente, este histograma se normaliza por el numero total de nodos.

De la Ecuación 3 se puede deducir que si las particiones son idénticas entonces $I_M = 1$, mientras que si las particiones son completamente independientes $I_M = 0$.

Se compararon las informaciones mutuas entre los 7 métodos y se dispusieron los resultados en un mapa de colores (ver figura 13). Dado que algunos métodos generan los clusters a partir de un con cierto grado de aleatoriedad (como caminatas aleatorias) la matriz de la figura 13 no resultó simétrica al calcularse una vez cada uno de sus elementos, por lo que se tomó el promedio del elemento ij con el elemento ji para que resulte simétrica.

Figura 13: *Mapa de colores representando la información mutua entre todos los pares de métodos.*

En general, todos los métodos presentan una información mutua relativamente alta. Se podría hipotetizar que el par *Eigenvectors – Louvain* tendría un valor especialmente alto, debido a que sus *silhouettes* particiones son extremadamente similares, pero este no es el caso, lo cual se explica dado que las particiones en sí, al mirar el grafo, no comparten tantos nodos realmente. Se puede observar que los pares *Walktrap – Fastgreedy* y *Edge Betweenness – Infomap* fueron los métodos con mayor información mutua, es decir que estos pares de métodos los que arrojaron las particiones más correlacionadas entre sí y al mirar los grafos correspondientes puede observarse que comparten una enorme cantidad de nodos (y difieren fundamentalmente en las comunidades más pequeñas, tales como las comunidades de dos nodos). Por otro lado, se puede observar que el par *Label prop – Louvain* fue el par con menor información mutua, debido a que cada método detectó una cantidad notablemente distinta de comunidades.

4.2. Análisis por matriz de confusión

Otro observable que podemos calcular para estimar que tanto se parece una partición a otra es el que denominaremos **precisión**, y se basa en la construcción de una matriz de confusión. Las matrices de confusión son matrices que sirven para comparar el desempeño de dos esquemas de clasificación sobre un mismo conjunto de datos. En estas matrices, la entrada ij corresponde al número de casos en los que el primer esquema clasifica un elemento como perteneciente a la clase i y, simultáneamente, el segundo esquema lo clasifica como perteneciente a la clase j.

Si bien cada partición constituye un método de clasificación de los nodos, estos no son directamente comparables dado que no hay un orden privilegiado en el cual disponer los clusters de una dada partición, i.e. los clusters, en cuanto clases dentro de las cuales "colocar" nodos, son entidades indistinguibles (comparar esto con el caso en que se quiere saber si un algoritmo es capaz de distinguir entre perros y gatos de la misma manera que un humano). Por lo tanto, las clases que se compararán no serán clases de nodos sino clases de *pares* de nodos. Dado un par no ordenado de nodos $\{i, j\}$, La primera clase será "i y j pertenecen al mismo cluster", y la segunda será "i y j pertenecen a clusters diferentes". Cada partición asignará los pares a estas clases de una manera diferente, y por lo tanto podemos utilizarlas para cuantificar el acuerdo entre ellas. En la Figura 14 se muestra un esquema de la matriz de confusión resultante.

Definimos que existe un error en la partición evaluada si dos nodos que pertenecen a la misma comunidad de referencia son asignados a diferentes comunidades en la partición en cuestión, o bien dos nodos de diferente comunidades en la partición de referencia son asignados a la misma comunidad en la partición en cuestión. Estos corresponden a los elementos de fuera de la diagonal de la matriz. Finalmente, definimos la precisión como la proporción de casos exitosos (no errores), según

$$p = \frac{a + d}{a + b + c + d} \tag{4}$$

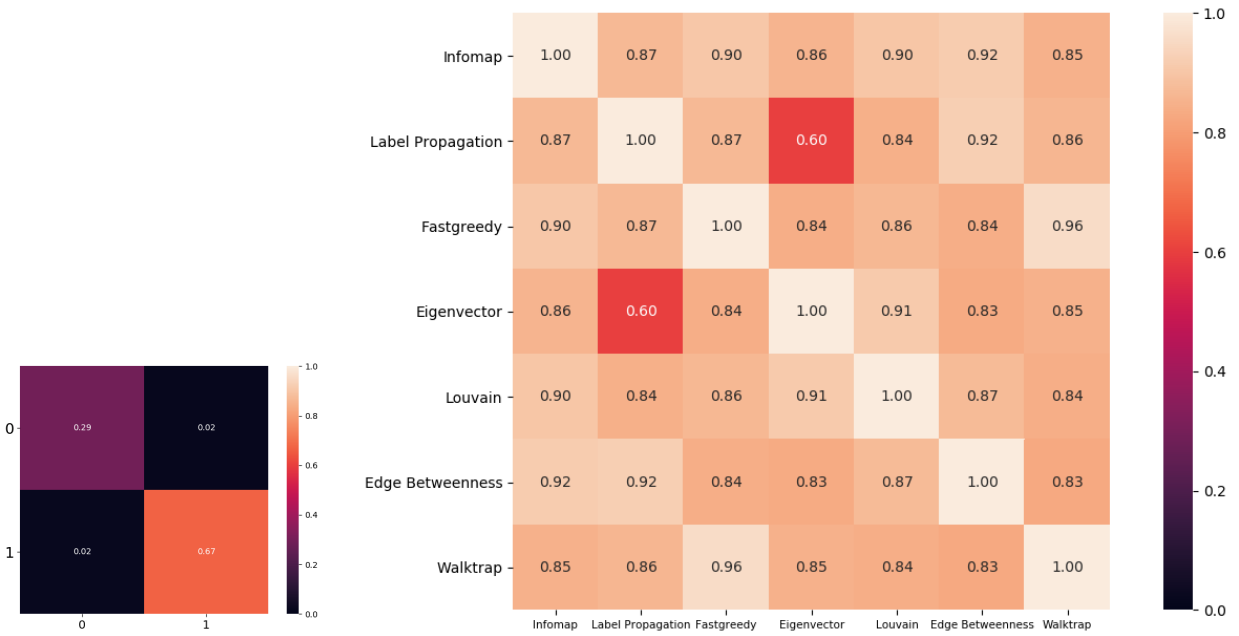
donde se empleó la notación de la Figura 14.

Es importante notar que, en nuestro caso, al comparar entre distintos métodos de detección de comunidades, no hay una partición de referencia privilegiada.

	$C_{ref}(i) = C_{ref}(j)$	$C_{ref}(i) \neq C_{ref}(j)$
$C(i) = C(j)$	a	b
$C(i) \neq C(j)$	c	d

Figura 14: Esquema del formato de la matriz confusión empleada.

En la Figura 15 (a) se muestra un ejemplo de matriz de confusión para la comparación de dos métodos, y calculando las precisiones para cada par de métodos se obtiene la figura 15 (b).



(a) (b)

Figura 15: (a) Matriz de confusión para la comparación entre los métodos *fast greedy* y *walktrap*. (b) Comparación entre métodos por medio del observable precisión.

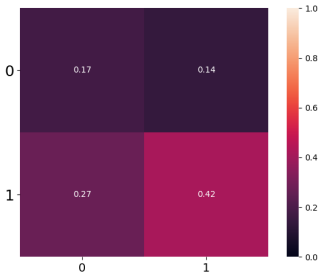
Se observa que los pares de métodos con mayor precisión son *Walktrap – Fast Greedy* (0.96), *Edge betweenness – Infomap* y *Edge betweenness – Label prop* (ambos con precision 0.92). A su vez, el par de métodos con menor precisión es *Eigenvctors – Label prop* (0.60). Se observa una compatibilidad entre estos resultados y los obtenidos mediante información mutua con respecto a los dos pares mejores rankeados, pero una discordancia con respecto a los pares peores rankeados, lo cual refuerza la idea de que ambos métodos de comparación de particiones resaltan propiedades diferentes de las mismas.

5. Relación entre sexos y estructura de comunidades

Dado que ninguno de los 7 métodos de detección de comunidades empleados tiene en cuenta el género de los delfines a la hora de armar clusters, es válido preguntarse si la distribución de machos y hembras está relacionada de alguna forma con los clusters obtenidos. En la figura 16 se observa el cálculo de información mutua entre la división por géneros y cada uno de los métodos de partición. Se observa allí que los métodos tienen una información mutua pequeña con la partición por géneros. Es decir, los clusters generados a partir de los métodos y los géneros están poco correlacionados. También se calculó la precisión, mediante el mismo método que el detallado arriba, para comparar los 7 métodos con la división por géneros, y se observa nuevamente que los valores son más bajos que los valores para la comparación entre métodos, si bien todos rondan el 60 % de aciertos. Se observa que las particiones que mas se asemejan a la partición por genero son las que corresponden a los métodos *Infomap* y *Edge betweenness*.

	Genero
Fastgreedy	0.160896
Eigenvector	0.164051
Edge betweenness	0.207911
Louvain	0.205337
Walktrap	0.134061
Infomap	0.246455
Label prop	0.187634

Figura 16: Información mutua entre los métodos de clustering y los géneros de los delfines.



(a)

	Genero
infomap	0.612374
label_prop	0.593337
fastgreedy	0.589106
eigenvector	0.565838
louvain	0.593337
edge_betweenness	0.610259
walktrap	0.579588

(b)

Figura 17: (a) Se muestra como ejemplo la matriz de confusión del método *Fast Greedy* (filas) versus la partición por géneros (coolumnas) tomando como referencia la partición por géneros. (b) Listado de las precisiones resultantes de comparar los métodos analizados con la partición por géneros.

Por otro lado, se puede cuantificar de otra forma la relación entre la estructura de la red y los métodos a partir del Test exacto de Fisher. Dadas cantidades fijas N_m y N_h de delfines machos y hembras en la red respectivamente ($N_m + N_h = N$), y tamaños fijos de los clusters, se plantea un modelo nulo en el cual no hay relación alguna entre los clusters y la división por géneros. Bajo este modelo nulo, la probabilidad de que un cluster C, de tamaño $|C|$, tenga una dada cantidad n de hembras (y por lo tanto una cantidad $|C| - n$ de machos) está dada por una distribución hipergeométrica, a saber

$$p(n; |C|, N_h, N_m) = \frac{\binom{N_h}{n}\binom{N_m}{|C|-n}}{\binom{N}{|C|}}$$

Para entender esto se puede considerar al problema como un experimento de éxitos (contar una hembra en el cluster) y fracasos (contar un macho en el cluster), donde la probabilidad de contar a una hembra depende de cuantos machos ya conté, ya que existe un número fijo de estas en la red total.

Para evaluar si existe sub- o sobrerrepresentación de algún sexo en una dada comunidad, basta entonces calcular cuál sería la probabilidad dado el modelo nulo de observar lo que se observó o algo más extremo. Esto es precisamente el p-valor del Test exacto de Fisher.

En la Figura 18 se muestran los p-valores obtenidos para cada cluster para cada partición de la red. Se encuentran resaltados los casos en los que se obtuvo $p < 0{,}05$ con el color correspondiente a la población sobrerrepresentada (rojo es macho, celeste es hembra). Es decir que, fijando arbitrariamente un umbral en 0,05, elegimos considerar que existe sobrerrepresentación cuando el p-valor obtenido es menor al mismo.

Para este análisis, se excluyó a los 4 delfines para los cuales no se tenía información de sexo (no fueron contados como parte de ninguna comunidad, ni como parte de la red total).

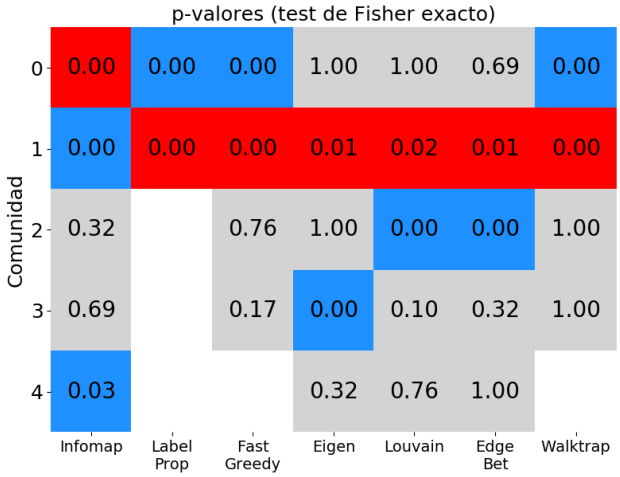


Figura 18: Se presentan los p-valores obtenidos para el test exacto de Fisher sobre la representación de sexos en cada cluster de cada uno de los métodos estudiados. Para los valores menores a 0.05, el color indica cuál es la población sobrerrepresentada (rojo es macho, celeste es hembra).

Tal como se observa en la figura, todos los métodos detectan un cluster en el cual se encuentra sin lugar a dudas una sobrerrepresentación de machos; se trata del cluster que aparece a la derecha en las visualizaciones del grafo. Por otro lado, todos los métodos detectan por lo menos una comunidad de hembras, y solo *Infomap* detecta dos con un p-valor menor al umbral elegido. Sin embargo, al observar este cluster en el grafo (la numeración de las comunidades corresponde al ordenamiento de las curvas de *silhouette* de abajo hacia arriba), vemos que el cluster al que corresponde el p-valor 0.03 tiene solo 4 nodos, y no es uno de los clusters principales detectados por el método. Lo mismo pasa con los p-valores 0.17 y 0.10 observados para *Fast Greedy* y *Louvain* respectivamente, los cuales son los únicos dos p-valores relativamente bajos que no caen debajo del umbral.

Por todo este análisis podemos concluir que existe definitivamente un (único) grupo de hembras con un conexionado interno denso, además del grupo de machos. Además, incorporando el análisis de *silhouettes*, podemos afirmar que este grupo de hembras está menos aislado del resto de la red que el grupo de machos.

Para enriquecer

6. Percolación de cliques

El algoritmo de percolación de cliques es un método para generar particiones en una red que forma comunidades a partir de cliques. Este método depende de la red y del grado k de los cliques elegido.

El método consiste en encontrar todos los k-cliques de una red, y utilizarlos como semilla. Es decir, una comunidad va a estar formada por un k-clique más todos aquellos nodos que compartan k-1 enlaces con el k-clique semilla. Notar que mediante este método, existen nodos que pueden resultar en más de una comunidad si comparten k-1 enlaces con 2 cliques distintos. Esto se denomina solapamiento entre comunidades.

Por ejemplo, en la figura 19 se puede observar una red que fue particionada con este método utilizando 4-cliques. Además, en rojo se pueden observar los nodos que resultaron en más de una comunidad.

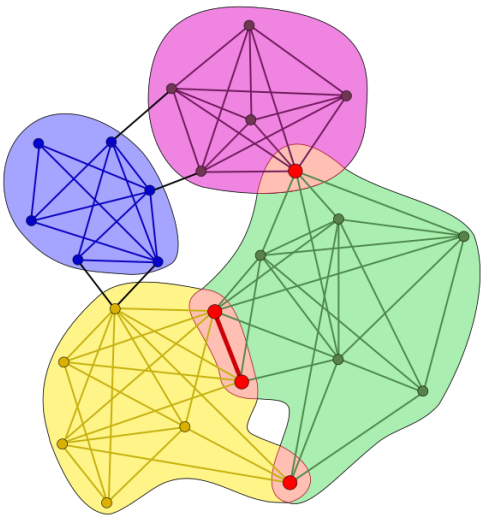


Figura 19: Ejemplo de una red particionada mediante percolación de 4-Cliques. En rojo, los nodos que quedaron en más de una comunidad a la vez.

Se aplicó el método de percolacion de cliques a la red de Delfines, para un rango de k entre 2 y 6. Se busco el delfin mas sociable como aquel que resulto en la mayor cantidad de comunidades para los distintos valores de k. En el cuadro 1 se puede observar el delfín que resultó ser el mas sociable en función el grado del método.

Grado del método:	k = 2	k = 3	k = 4	k = 5	k = 6
Delfín más sociable:	Grin	Grin	Grin	Jonah	Beak

Cuadro 1: *Delfin mas sociable en funciond el grado del método.*

Referencias

[1] Lusseau, D. (2007) Evidence for social role in a dolphin social network. Evolutionary Ecology 21(3): 357-366

[2] Rousseeuw, P.J. (1986) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Computational and Applied Mathematics. 20: 53-65. doi:10.1016/0377-0427(87)90125-7

[3] Rosvall, M. y Bergstrom, C.T. (2008) Maps of random walks on complex networks reveal community structure. PNAS 105(4): 1118-1123. doi:10.1073/pnas.0706851105

[4] Raghavan, U.N. et al. (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76, 036106.

[5] Clauset, A. et al. (2004) Finding community structure in very large networks. Phys. Rev. E 70, 066111. doi:10.1103/PhysRevE.70.066111

[6] Newman, M.E.J. (2006) Finding community structure using the eigenvectors of matrices, Physical Review E 74 036104. doi: 10.1103/PhysRevE.74.036104

[7] Blondel, V.D. et al. (2008) Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment (10), pp. 10008, 12 pp. doi:10.1088/1742-5468/2008/10/P10008

[8] Newman, M.E.J. y Girvan, M. (2004) Finding and evaluating community structure in networks. Physical Review E 69, 026113. doi:10.1103/PhysRevE.69.026113

[9] Pons, P. y Latapy, M. (2005) Computing communities in large networks using random walks. eprint arXiv:physics/0512106 [physics.soc-ph]