Trabajo Computacional 3: Análisis de Comunidades

Lucio García, Lucas Longo, María Luz Vercesi

Redes Complejas
7 de noviembre de 2018

1. Introducción

La división en comunidades se basa en la idea de que existen grupos en los que se divide naturalmente la red. Hay diferentes criterios para decidir qué significa *naturalmente*, y por lo tanto existen diferentes algoritmos para encontrar estos grupos. Usualmente la idea de una modularidad máxima se relaciona con una estructura de comunidad óptima [1]. Esta hipótesis es el punto de partida en el que se basan diversos algoritmos de detección de comunidades, entre los que se encuentran: fast-greedy y Louvain. Debido a que la hipótesis de máxima modularidad presenta sus limitaciones, también existen otros tipos de métodos que detectan comunidades mediante caminatas aleatorias, como infomap, o quitando enlaces según centralidad, como edge-betweeness. En este trabajo, se utilizaron y compararon los cuatro métodos nombrados, donde, exceptuando infomap, todos los algoritmos fueron desarrollados por nosotros.

El método de **Louvain** se basa en calcular la variación de modularidad resultante de unir un nodo con cada uno de sus vecinos, y seleccionar el aumento máximo. Este proceso se repite para cada nodo en la red. Una vez que se recorrieron todos, se arma una nueva red donde cada nodo representa una comunidad con una cantidad de auto enlaces que está dada por la cantidad de enlaces que había dentro de esa comunidad. Estos pasos se repiten hasta que no se puede lograr un aumento en la modularidad.

En el caso de **fast-greedy** lo que se busca también es lograr una maximización de la modularidad. El método comienza con cada nodo asignado a una comunidad "propia". A continuación se van agrupando nodos en una misma comunidad siempre y cuando el resultado de dicha unión sea un aumento de la modularidad. El algoritmo puede finalizar cuando se alcanza una meseta de particiones óptimas las cuales no causen aumento de modularidad, ya que si se siguen uniendo comunidades la modularidad comenzará a disminuir hasta que todos los nodos formen parte de una única comunidad. La partición que se selecciona es la que presenta modularidad máxima.

Para el método de **edge-betweenness**, se comienza con la comunidad entera (todos los nodos y enlaces). Se calcula la centralidad de edge-betweenness de cada enlace, y se elimina un único enlace con valor máximo. Se recalcula la centralidad para el nuevo grafo, y se repite el paso anterior. De esta manera, cada cierta cantidad de pasos, la comunidad se dividirá en un componente más que el paso anterior, hasta llegar a que cada nodo es su propia comunidad. Para decidir cuál es la mejor división, también se utiliza el criterio de máxima modularidad. A diferencia de los métodos anteriores, la modularidad se puede calcular en cada paso y por eso se puede evitar el problema de máximos locales.

Por último, **infomap** es un algoritmo que utiliza el código de compresión de Huffman para realizar un etiquetado en dos niveles: index-code book y module-code book. El primero son los códigos asignados a la entrada y salida de cada comundiad, mientras que el segundo se refiere al codigo interno de cada nodo dentro de su comunidad. Este método minimiza una funcion llamada *ecuación de mapa*, que representa la eficiencia de la descripción dada por una caminata aleatoria [2].

La red que se analizará es una red de delfines, que se encuentra representada en la Fig.1. Está compuesta por 62 nodos de los cuales 34 son delfines machos (azul en la figura), 24 son delfines hembras (rosa), y 4 son delfines sin un género identificado (verde). En un análisis previo de esta red se encontró que presenta homofilia de género. Para ello se habían utilizado dos estrategias. Por un lado se procedió a realizar una reasignación aleatoria de género en los nodos de la red, manteniendo la topología de la misma y contando la cantidad de enlaces f-m (hembra-macho) en cada reasignación. Se encontró que la cantidad de enlaces f-m de la red real era mucho menor ($p_{valor} = 0{,}0006$) que la esperada según la distribución de enlaces de la hipótesis nula. La segunda estrategia consistió en medir la asortatividad según género en la red encontrándose que la red presenta una modularidad positiva de $M = 0{,}311$.

Emplearemos estos dos métodos para analizar la existencia de homofilia dentro de cada comunidad para ver si estas comunidades son representativas de toda la red o no lo son en cuanto a la homofilia.

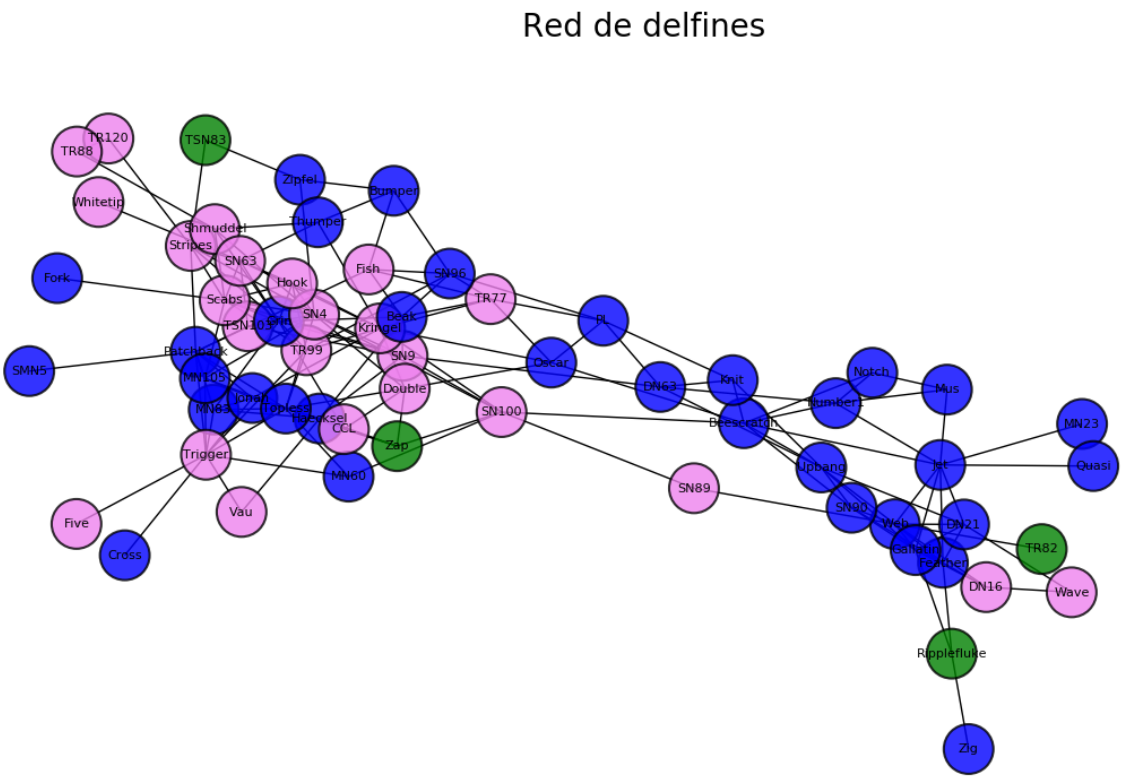


Figura 1: Grafo de la red real: en azul los delfines macho, en rosa las hembras y en verde los de género no identificado.

2. División en comunidades

En la figura 2 se pueden ver los resultados que se obtuvieron al aplicar cada método. Los nodos que pertenecen a una misma comunidad estan pintados del mismo color. El método fast-greedy resultó en 4 comunidades, Louvain y edge-betweenness ambos resultaron en 5, mientras que infomap resultó en la mayor cantidad de comunidades (6). Esto indica que podría haber cierta consistencia entre ellos, aunque cada uno identificó a distintos delfines dentro de cada comunidad. En varios casos, esto puede deberse a que el algoritmo no haya resuelto de forma eficiente el solapamiento entre comunidades.

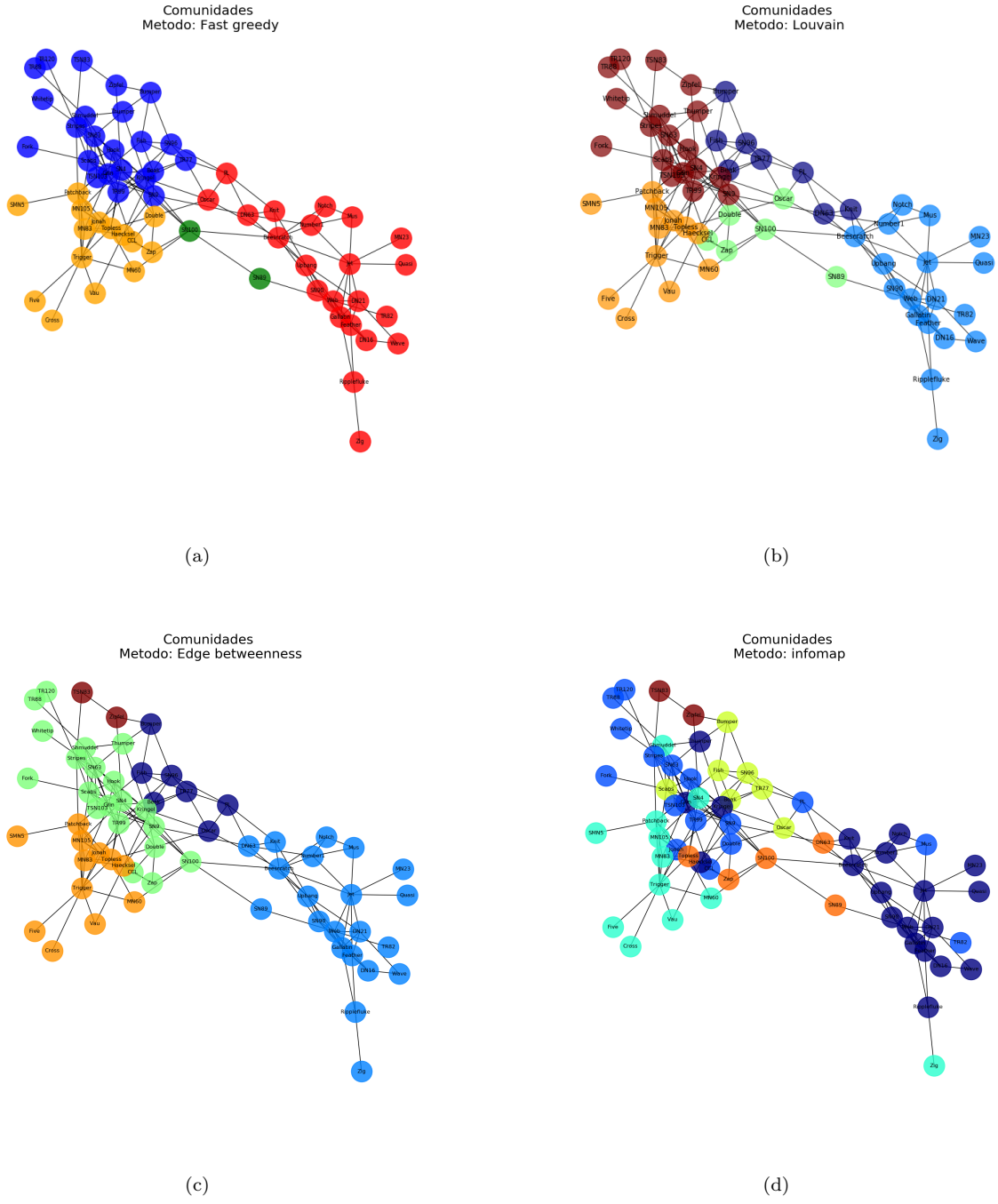
3. Modularidad y silhouette

La mayoría de los métodos utilizan como medida para encontrar particiones a la modularidad. Esta medida de calidad de la partición descansa sobre una hipótesis importante que es que las redes al azar carecen de una estructura de comunidad inherente. Esta hipótesis tiene como consecuencia que comparando la densidad de enlaces de la red real con la densidad de enlaces obtenida de una red recableada, permite decidir si la comunidad original corresponde a un 'verdadero' subgrafo denso o si en realidad su conectividad se debe a una emergencia por azar. Se medirá la diferencia entre la cantidad de enlaces de la comunidad c y la cantidad de enlaces esperados en la misma comunidad por una proceso de recableo que mantiene los grados de la siguiente manera:

$$M_c = \frac{L_c}{L} - \left(\frac{k_c}{L}\right)^2 \tag{1}$$

Aquí L_c es la cantidad de enlaces entre nodos dentro de la misma comunidad y k_c la suma de los grados de esos nodos. Si M_c es un valor positivo entonces esto dice que la comunidad c tiene mas enlaces que los esperados por azar y que entonces representa a una 'verdadera' comunidad. Puede obtenerse el valor de modularidad total M sumando los M_c de cada comunidad.

Otra medida complementaria de qué tan buena es una partición es el valor de silhouette que posee cada nodo. A diferencia de la modularidad la cual se calcula para una comunidad, silhouette pasa a ser una medida local brindando información sobre qué tan bien está asignado cierto nodo a su comunidad. Este valor puede calcularse como:



(a) (b)

(c) (d)

Figura 2: Grafo de las comunidades encontradas mediante el método de (a) fast-greedy (b) Louvain (c) edge-betweenness (d) infomap.

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

Aquí $a(i)$ es la distancia media del nodo i a los nodos de su propia comunidad, mientras que $b(i)$ es la mínima distancia promedio del nodo i con nodos de otra comunidad. Si el valor de silhouette de un nodo es cercano a 1 quiere decir que ese nodo en general estará bien asignado a su comunidad aunque pueden existir excepciones.

En el caso del método de fast-greedy se obtuvieron cuatro comunidades. En la figura (2b) puede verse que una de estas comunidades (verde) que el método encuentra está formada por dos delfines únicamente. Se calcularon las modularidades teórica de estas comunidades 'blue', 'red', 'orange' y 'green', y los resultados se resumen el el Cuadro 1 con lo cual se ve que la comunidad verde resultó ser la de más baja modularidad. La modularidad total de la partición encontrada por el método de fast-greedy fue de $M = 0{,}495$. Además para cada nodo se calculó su valor de silhouette, obteniéndose un valor promedio de $S = 0{,}184$ y la distribución de la Fig7a.

Una característica importante para mencionar es que con este método se observó que algunos delfines pueden cambiar de comunidad cuando el algoritmo se implementa repetidas veces. Por ejemplo, el delfin 'Oscar' cambia entre la comunidad azul y la comunidad roja. Podemos observar en la figura 7a del apéndice que este delfin posee un valor de silhouette negativo, es decir, que es un delfin que podría no estar bien asignado a su comunidad, aunque también hay delfines con silhouette negativo que quedan siempre asignados a una misma comunidad por el algoritmo.

Método fast-greedy				
Comunidades	**Enlaces (red real)**	**Enlaces (red rewiring)**	**Modularidad (rewiring)**	**Modularidad (teórica)**
Blue	33	25.2	0.0489	0.168
Red	39	15.7	0.1460	0.192
Orange	10	8.8	0.0070	0.128
Green	1	0.038	0.0060	0.0054

Cuadro 1: Modularidad en las comunidades de fast-greedy. El valor de enlaces por rewiring corresponde al valor medio de enlaces de la distribución obtenida por 20000 recableos

El recableado se realizó para los distintos métodos, aunque sólo se presentan los resultados de modularidad en el cuadro 1 para el método fast-greedy. En el apéndice, en las figuras 4, 5 y 6, se pueden ver los histogramas correspondientes a cada comunidad para los distintos métodos, pudiéndose observar que cada una presenta una mayor modularidad que la esperada al azar.

En cuanto al valor de silhouette, las figuras 7a, 7b y 8 del apéndice muestran lo encontrado en cada método. En particular, al obtener cada división en comunidaddes, el método de edge-betweenness permitió calcular modularidad y silhouette para cada división. Así se pudo confirmar que la separación en 5 comunidades, con una modularidad $M = 0{,}52$, también es la óptima en función del valor de silhouette, con un valor medio de $S = 0{,}34$. En la figura 8 se puede ver que la separación en dos grandes comunidades produce un resultado mayor en valor medio, lo cual tiene sentido porque es usual que pocas comunidades grandes tengan silhouette alto. Sin embargo, lo que se busca es un alto silhouette en conjunto con una alta modularidad.

4. Comparación de las particiones

Para cuantificar el acuerdo entre las particiones según los distintos métodos, se utilizó la matriz de confusión (que compara casos positivos y negativos entre clases) y el valor de precisión. Teniendo los datos de cada método, se consideró cada par de delfines y se compararon entre métodos su pertenencia a una misma o distintas comunidades. Así se obtuvieron valores de precisión que se pueden observar en el cuadro 2.

Se puede ver que aunque las comunidades no sean exactamente las mismas, hay una superposición importante entre las particiones entre los métodos de fast-greedy, edge-betweenness y Louvain. Sin embargo, con el método de infomap, con el cual se encontraron más comunidades que con los otros, la coincidencia es mucho menor. Por esta razón, los resultados de las siguientes secciones son válidos para los métodos que tuvieron mayor coincidencia entre sí.

	Edge-betweenness	**Louvain**	**Infomap**
Fast-greedy	0.88	0.91	0.68
Edge-betweenness		0.91	0.69
Louvain			0.71

Cuadro 2: Precisión obtenida al comparar las particiones según los distintos métodos utilizando la matriz de confusión.

5. Test de Fisher exacto

El test de Fisher exacto es un test de significación estadística que predice el p-valor exacto de un experimento. Usualmente el p-valor es comparado con un nivel de significancia (α) elegido de manera arbitraria que suele tomar por convención alguno de los siguientes valores: $0{,}01, 0{,}05, 0{,}005$ y $0{,}001$. Si el p-valor es significativamente menor que α, significa que la hipótesis nula es rechazada; sin embargo, si el p-valor es mayor no se puede afirmar que haya ninguna diferencia significativa entre las pruebas y la hipótesis nula. En este caso el test de Fisher puede servir para determinar si en las comunidades encontradas hay homofilia o no.

Para ello partimos conociendo la composición en género de nuestra red, formada por $M = 34$ machos y $F = 24$ hembras. A continuación podemos pensar que si nuestra comunidad que conocemos tiene n_c individuos

de los cuales m_c son machos y f_c son hembras esto es equivalente a sacar una muestra de n_c individuos de la red original y contar la cantidad de hembras y machos en esa muestra. En definitiva, queremos evaluar la probabilidad de que en nuestra comunidad de n_c individuos hayan f_c hembras y $m_c = n_c - f_c$ machos que serán los valores que nosotros hemos observado en las particiones realizadas. En este caso un valor bajo o alto de p-valor, nos dice que la probabilidad de haber obtenido el valor que se observó es muy baja y que por lo tanto esto será equivalente a aceptar la hipótesis de homofilia o de forma equivalente rechazar la hipótesis nula de que no hay homofilia en la comunidad analizada. Se puede demostrar que la probabilidad de observar f_c hembras en una muestra de n_c delfines a partir de una red de N delfines, con M machos y F hembras está dada por:

$$p(f = f_c) = \frac{\binom{F}{f_c}\cdot\binom{M}{m_c}}{\binom{N}{n_c}} = \frac{\binom{F}{f_c}\cdot\binom{N-F}{n_c-f_c}}{\binom{N}{n_c}} = \frac{\binom{24}{f_c}\cdot\binom{58-24}{n_c-f_c}}{\binom{58}{n_c}} \tag{3}$$

En el cuadro 3, se pueden observar los resultados obtenidos luego de hacer el test de Fisher exacto para el algoritmo fast-greedy. Si tomamos $\alpha = 0{,}05$ las dos comunidades que rechazan la hipótesis nula son *Blue* y *Red*. En el caso de *Blue* vemos que el número de hembras es mayor al de machos y esto significa que hay homofilia entre hembras. En el caso de la comunidad *Red* el p-valor es muy cercano a 1, lo cual se explica por el número bajo de hembras encontradas, es decir que también hay homofilia pero de machos. Por la forma en que se calculó el p-valor, tanto $p < 0{,}05$ como $p > 0{,}95$ indican homofilia, entre hembras y entre machos respectivamente.

En los cuadros 3, 4 y 5 se pueden observar los resultados obtenidos en cada método. En todos los casos se encontraron dos comunidades con homofilia, una entre hembras y otra entre machos.

Test de Fisher exacto (método fast-greedy)					
Comunidades	**N delfines**	**N hembras**	**N machos**	**pFischer**	**pValor**
Blue	22	15	7	0.0027	0.0032
Red	20	2	18	0.0007	0.9990
Orange	14	5	9	0.220	0.7880
Green	2	2	0	0.166	0.1660

Cuadro 3: Análisis de la variable género dentro de cada comunidad de fast-greedy por medio del Test de Fisher exacto.

Test de Fisher exacto (método Louvain)					
Comunidades	**N delfines**	**N hembras**	**N machos**	**pFischer**	**pValor**
0	8	3	5	0.2938	0.6706
1	17	4	13	0.0499	0.9670
2	5	3	2	0.2478	0.2888
3	12	4	8	0.2163	0.7722
4	16	10	6	0.0330	0.0252

Cuadro 4: Análisis de la variable género dentro de cada comunidad de Louvain por medio del Test de Fisher exacto.

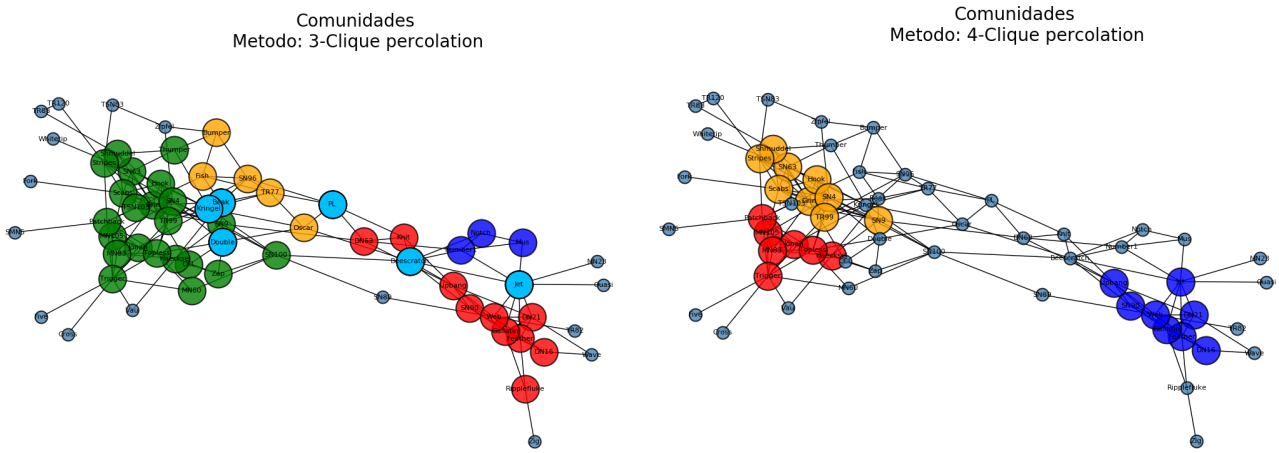
Test de Fisher exacto (método edge betweenness)					
Comunidades	**N delfines**	**N hembras**	**N machos**	**pFischer**	**pValor**
0	7	2	5	0.6878	0.8748
1	19	3	16	0.0096	0.9993
2	19	16	3	$7\text{x}10^{-6}$	$5\text{x}10^{-6}$
3	12	3	9	0.3243	0.9511
4	1	0	1	1	1

Cuadro 5: Análisis de la variable género dentro de cada comunidad de edge-betweenness por medio del Test de Fisher exacto.

6. Percolación de Cliques y Overlapping

Además de la maximización de la modularidad, hay otras medidas que pueden darnos una buena idea de comunidad: una de ellas es el concepto de cliques. Estos cliques suelen ser utilizados como semillas para encontrar comunidades más grandes. Para ello, lo que hicimos fue tomar un grado k y buscar todos los cliques del grafo de dicho grado. Luego se construye un grafo de cliques comenzando desde uno cualquiera y avanzando por el método de 'rolling cliques', es decir, se avanza hacia un clique adyacente, donde dos cliques se consideran conectados o adyacentes si comparten $k-1$ nodos.

En la Fig. 3 podemos ver los resultados del algoritmo para percolación de 3-cliques(a) y de 4-cliques(b). Una primera diferencia es la cantidad de comunidades que ambos métodos encontraron. En menor tamaño se muestran los delfines a los cuales la percolación no pudo alcanzar, que se encuentran en mayor cantidad en la percolación de 4-cliques que en la de 3-cliques. Muchos de estos delfines se encuentran en la periferia de la red y están conectados a una cierta comunidad por medio de un enlace, por lo que podrían ser asignados a alguna de estas fácilmente, aunque no fueron las que originalmente encontró el método de percolación. En celeste se muestran los nodos que pertenecen a más de una comunidad y que podríamos llamar los delfines más sociables: 'Jet', 'Beescratch', 'Kringel', 'Beak', 'Double' y 'PL'. Quisimos también evaluar el valor de grado k del silhouette de estos nodos para observar si hay alguna relación entre ellos y el hecho de pertenecer a más de una comunidad.



(a) (b)

Figura 3: Comunidades encontradas según el método de percolación de cliques: a) 3-cliques b) 4-cliques. En a) los nodos celestes corresponden a los delfines más sociables ya que pertenecen a más de una comunidad. Los nodos mas pequeños son aquellos nodos donde el método no llega a percolar pero algunos de ellos pueden ser fácilmente asignados a una comunidad ya que muchos están unidos por un solo enlace a alguna de estas.

Delfines Overlapping(k)	**Fast-greedy (sil.)**	**Louvain (sil.)**	**Edge betweenness (sil.)**
Jet(9)	0.21	0.46	0.57
Beescratch(8)	-0.21	-0.06	0.26
Kringel(9)	-0.20	-0.23	-0.02
Beak(6)	0.20	0.34	0.415
Double(5)	0.16	0.40	0.13
PL(5)	-0.026	0.53	0.59

Cuadro 6: Análisis comparativo de los delfines que se encuentran en comunidades solapadas por el método de 3-cliques. Entre paréntesis se muestran el grado de estos delfines y en cada columna el valor de silhouette de cada delfin en cada método.

Los valores de grado mínimo y máximo en la red de delfines son $k_{min} = 1$ y $k_{max} = 11$ con un $k_{medio} = 5$ con lo cual parece que no existe una evidencia de que el grado sea algo determinante en los delfines con overlap, pero sí es interesante observar que corresponden a grados que se encuentran por encima del valor medio de grado, todos los delfines con overlap poseen grado $k \geq 5$. Si comparamos el valor medio de los silhouettes encontrados en cada método con el valor medio de los delfines con overlap se puede observar que el valor de silhouette de estos últimos se encuentra por debajo de la media del valor de silhouettes de todos los nodos, pero no se puede afirmar que haya una correlación entre el rol de estos nodos en la red con su valor de sillhouette.

6.1. Algoritmo de percolación de cliques:

El algoritmo se basa en la estrategia de *rolling cliques*. La Fig. 10 muestra un esquema de cómo implementamos esto para el caso de percolación de 3-cliques pero el método mantiene la misma idea para cliques de mayor grado. Se comienza por hacer una lista llamada ***bolsa_cliques*** de todos los 3-cliques que tengamos en nuestra red. Al mismo tiempo se toma otra lista vacía llamada ***bolsa_comunidad***. Se saca un primer clique de la lista *bolsa_cliques* y lo guardamos en la lista *bolsa_comunidad*, eliminándolo de la primera. Luego se saca un nuevo clique de *bolsa_cliques* y se fija si es adyacente (matchea) con alguno de la *bolsa_comunidad* (al principio solo habrá uno pero luego *bolsa_cliques* se irá "llenando"). Si es adyacente, entonces ese clique pertenece a la comunidad que se está armando y se agrega a *bolsa_comunidad* y es eliminado de *bolsa_cliques*. En caso de no matchear con ninguno de los de *bolsa_comunidad* en ese momento se lo devuelve a *bolsa_cliques*. El proceso continúa probando cliques y agregándolos a *bolsa_comunidad* hasta que llegue un momento en que el tamaño de esta lista ya no varía más. Entonces la comunidad se da por finalizada, se vacía la lista *bolsa_comunidad* y se comienza a armar una comunidad nueva con los cliques que quedaron en *bolsa_cliques*.

NOTA : Un clique devuelto a la *bolsa_cliques* en algún momento del algoritmo puede aún así pertenecer a la comunidad que se está formando y que solo por una cuestión de tiempo y de estado de la *bolsa_comunidad* en ese momento, es que tuvo que ser devuelto. Sin embargo la iteración asegura que ese clique sea en algún momento posterior puesto correctamente en la *bolsa_comunidad* hasta que la misma ya no cambie su tamaño, entonces nos damos cuenta que la comunidad quedó terminada.

7. Conclusiones

Se pudieron utilizar distintos métodos para la detección de comunidades en la red de delfines. Se analizaron los valores de modularidad y silhouette de las particiones encontradas, para confirmar que se trataran de las óptimas. También pudimos implementar un algoritmo de percolación de 3-cliques y de 4-cliques, y en el primero encontramos comunidades donde existen delfines que pueden pertenecer a más de una comunidad al mismo tiempo, y que pueden ser considerados los más sociables. Vimos que el grado de estos delfines se encontraba por encima o era igual al valor de grado medio de los delfines de la red.

Utilizando la matriz de confusión y el valor de precisión, se pudo analizar el acuerdo entre las particiones encontradas mediante distintos métodos, encontrando una buena coincidencia entre los de fast-greedy, edge-betweenness y Louvain.

Utilizando el test de Fisher exacto, se pudo realizar un análisis de homofilia para las comunidades, encontrando en todos los casos que dos de las comunidades presentan homofilia (una de machos y otra de hembras).

Finalmente, se encontraron comunidades mediante el método de percolación de cliques, para 3-cliques y 4-cliques. Se encontró que los delfines que pertenecen a más de una comunidad presentan un grado mayor o igual que la media.

Referencias

[1] A.L. Barabasi. ***Network Science*** **Chapter 9. Communities**, 2016.

[2] http://www.mapequation.org/assets/publications/EurPhysJ2010Rosvall.pdf

Muy buen trabajo. Completo y con extras de yapa muy buenos. Bien explicado además y con bu

8. Apéndice de Figuras

8.1. Modularidades

8.1.1. Fast-Greedy

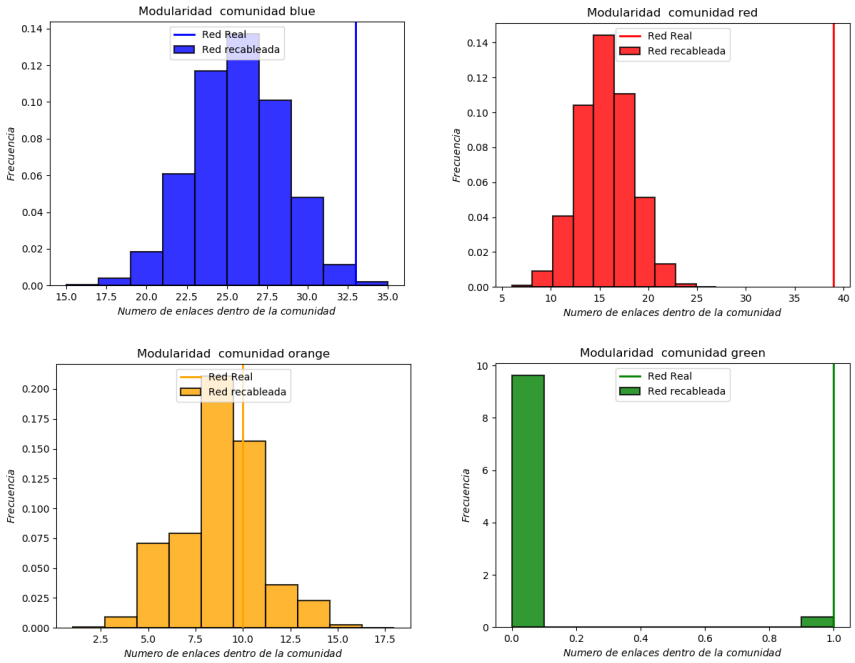


Figura 4: Histogramas para analizar las modularidades en las comunidades del método fast-greedy para 20000 recableados de las redes. Las líneas verticales indican el valor de enlaces dentro de la comunidad para la red real.

8.1.2. Louvain

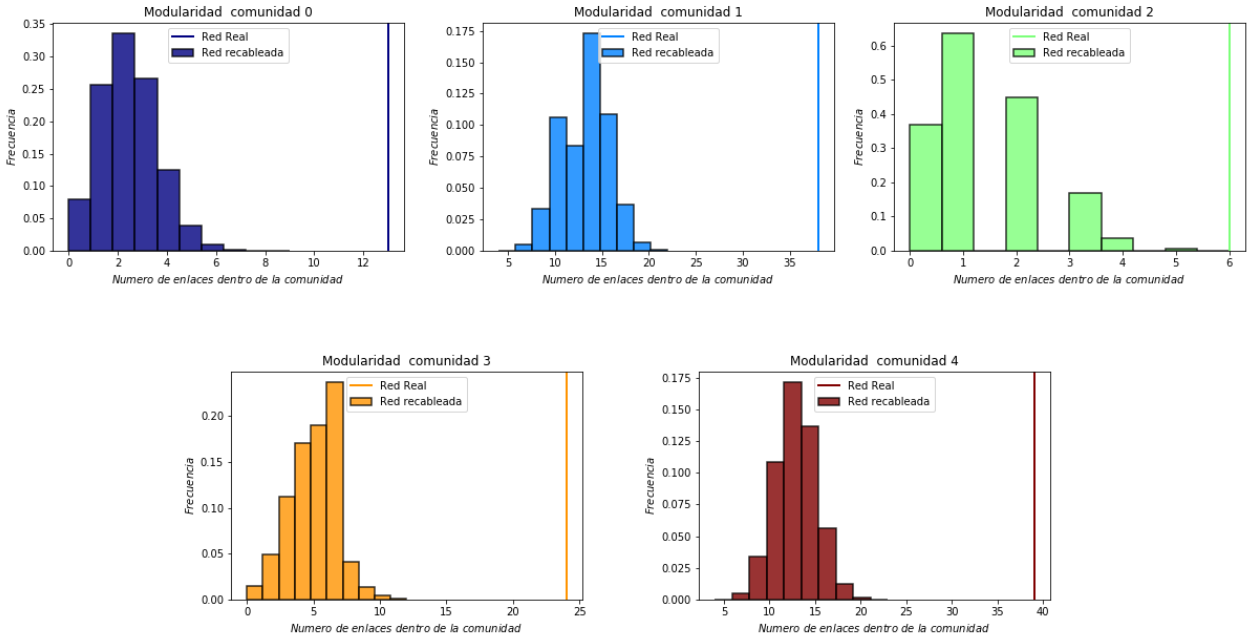


Figura 5: Histogramas para analizar las modularidades en las comunidades del método Louvain para 20000 recableados de las redes. Las líneas verticales indican el valor de enlaces dentro de la comunidad para la red real.

8.1.3. Edge-betweenness

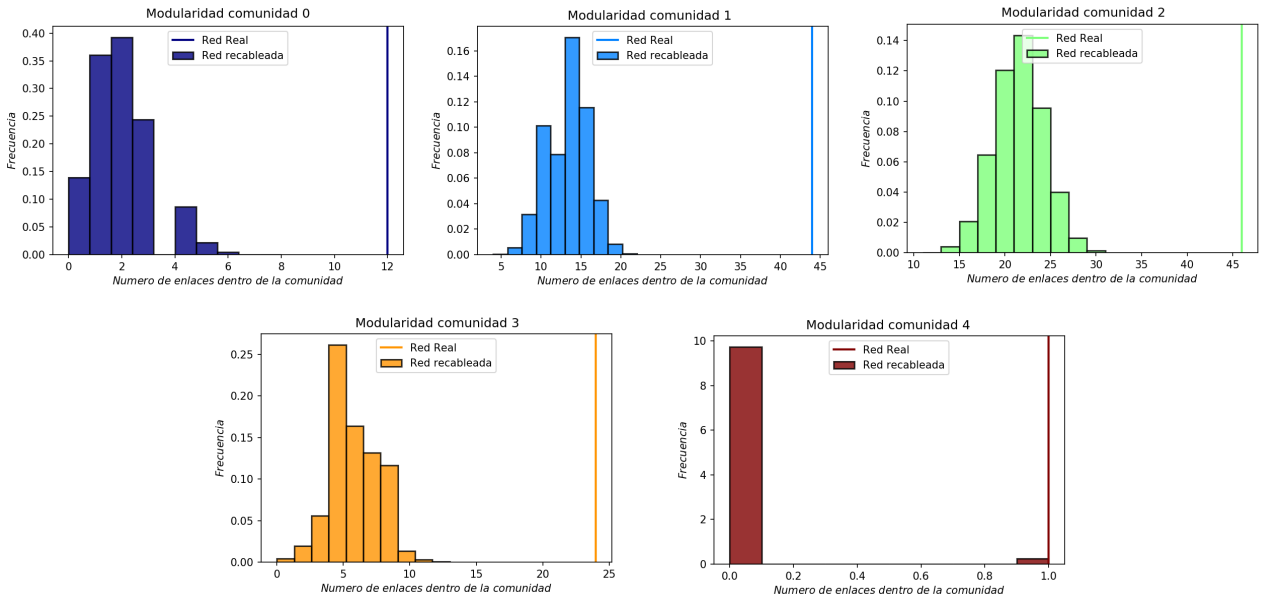
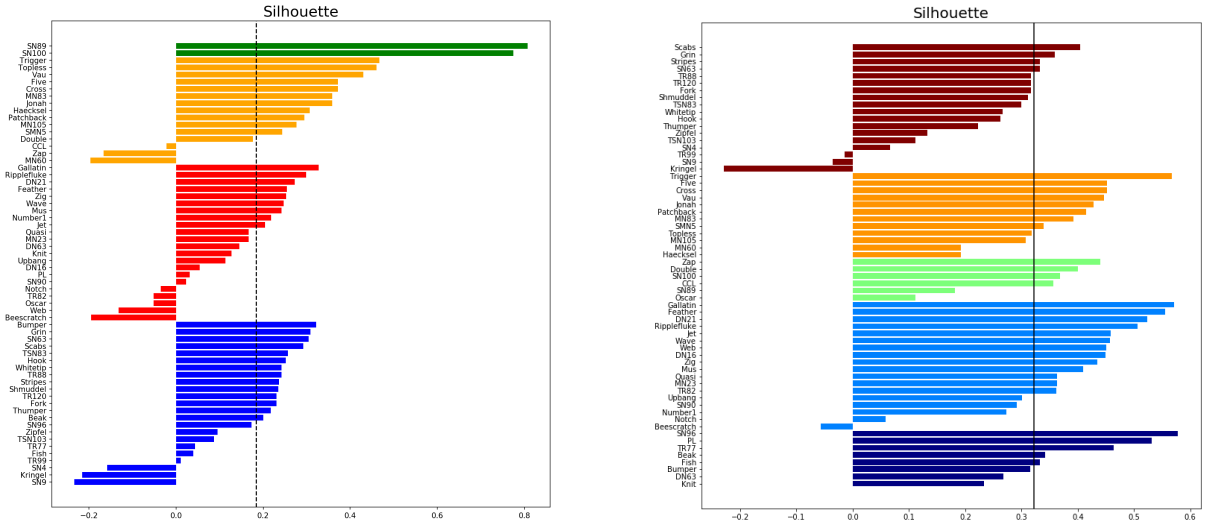


Figura 6: Histogramas para analizar las modularidades en las comunidades del método edge-betweenness para 20000 recableados de las redes. Las líneas verticales indican el valor de enlaces dentro de la comunidad para la red real.

8.2. Silhouettes

8.2.1. Fast-greedy y Louvain



(a) (b)

Figura 7: Gráfico de los valores de silhouette para cada delfin en las distintas particiones obtenidas al utilizar el método de (a) fast-greedy y (b) Louvain. La línea punteada marca el valor medio.

8.2.2. Edge-betweenness

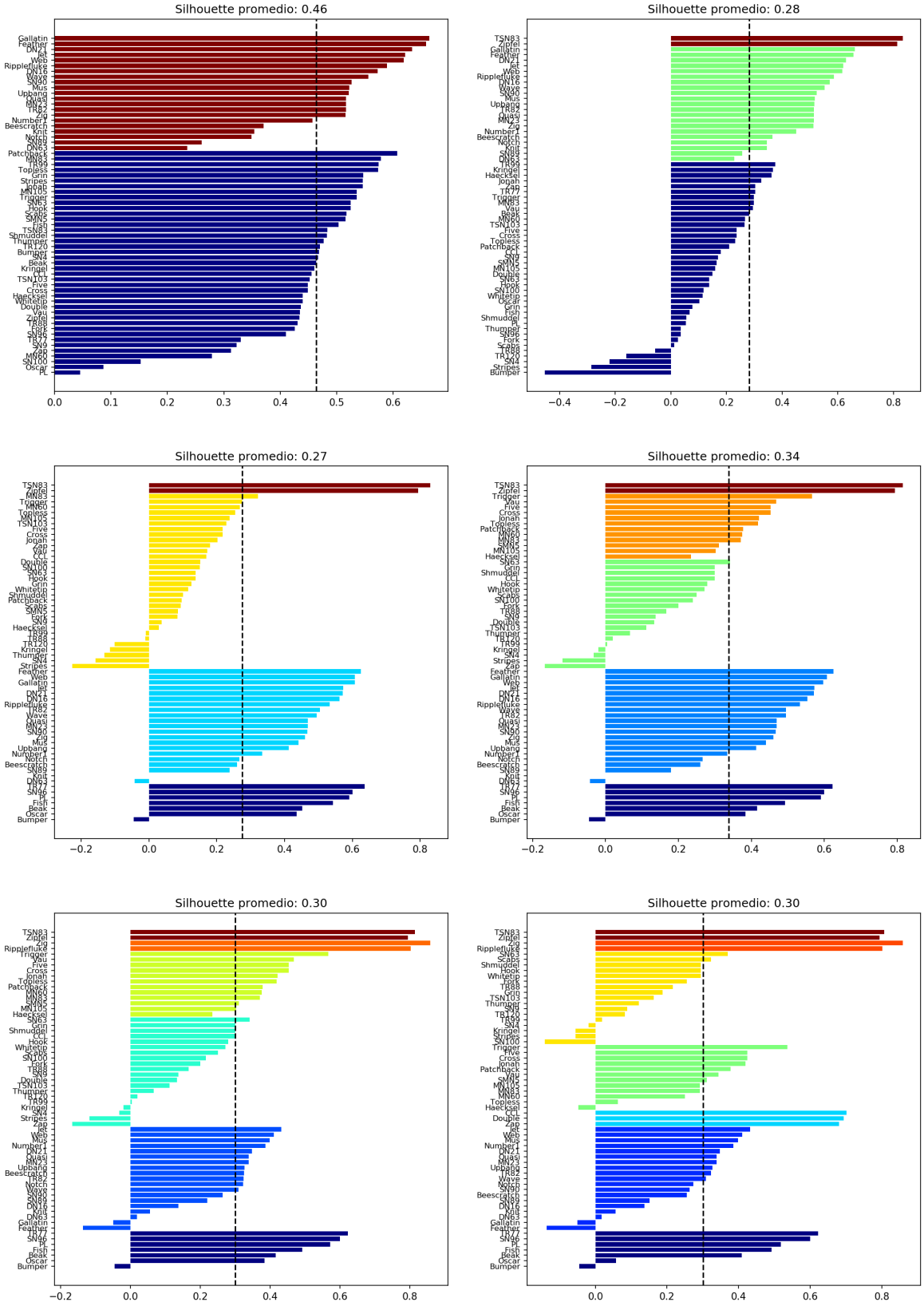


Figura 8: Gráficos de los valores de silhouette para cada delfin en las distintas particiones obtenidas al utilizar el método de edge-betweenness, para diferentes cantidades de comunidades. La línea punteada marca el valor medio.

8.3. Distribución de enlaces mixtos f-m

8.3.1. Fast-Greedy

Hay algo raro, los nodos verdes no están conectados entre sí y sin embargo están e

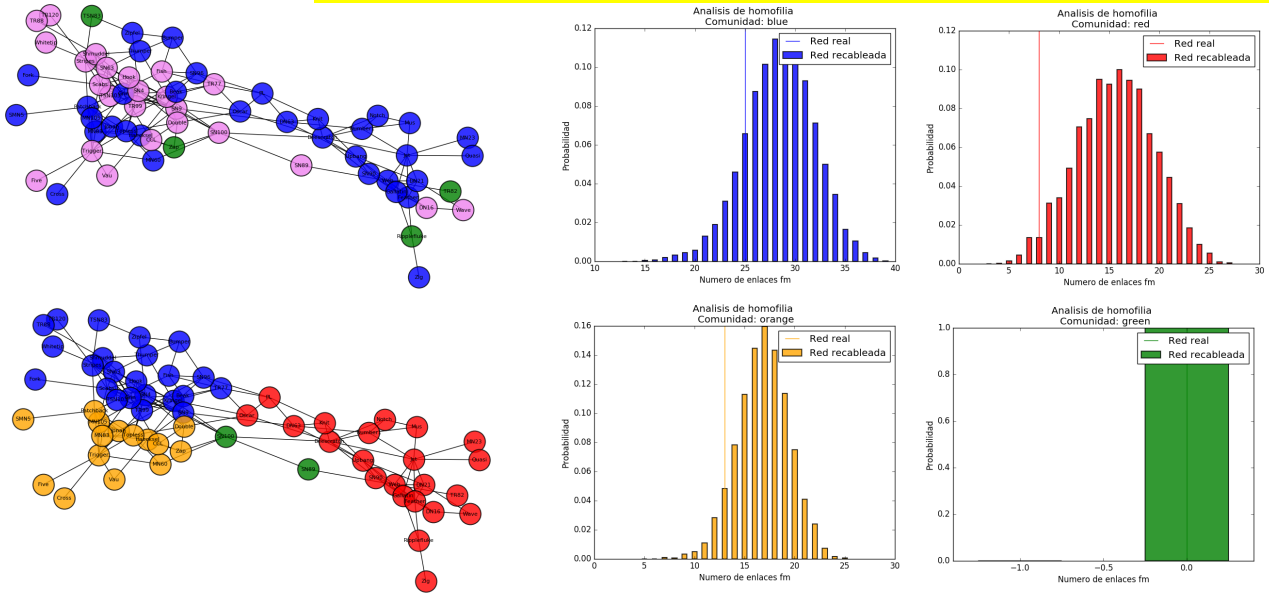


Figura 9: Histogramas para analizar la presencia de homofilia dentro de cada comunidad del método fast-greedy. Se realizaron 10000 reasignaciones aleatorias de género dentro de cada comunidad y se contaron enlaces f-m. La linea vertical corresponde a la cantidad de enlaces f-m dentro de cada comunidad de la red real.

Distribucion de enlaces fm (método fast-greedy)				
Comunidades	**Valor Medio (Hnull)**	**Desviacion Estandar**	**Valor Red Real**	**pValor**
Blue	26.64	3.49	23	0.1214
Red	15.67	3.9	8	0.0169
Orange	16.82	2.59	13	0.0514
Green	0	0	0	0

Cuadro 7: Análisis de homofilia dentro de cada comunidad del método fast-greedy. La primera columna corresponde al valor medio de las distribuciones de enlaces f-m para 10000 reasignaciones aleatorias de género.

8.4. Algoritmo de percolación de cliques 'using two bags':

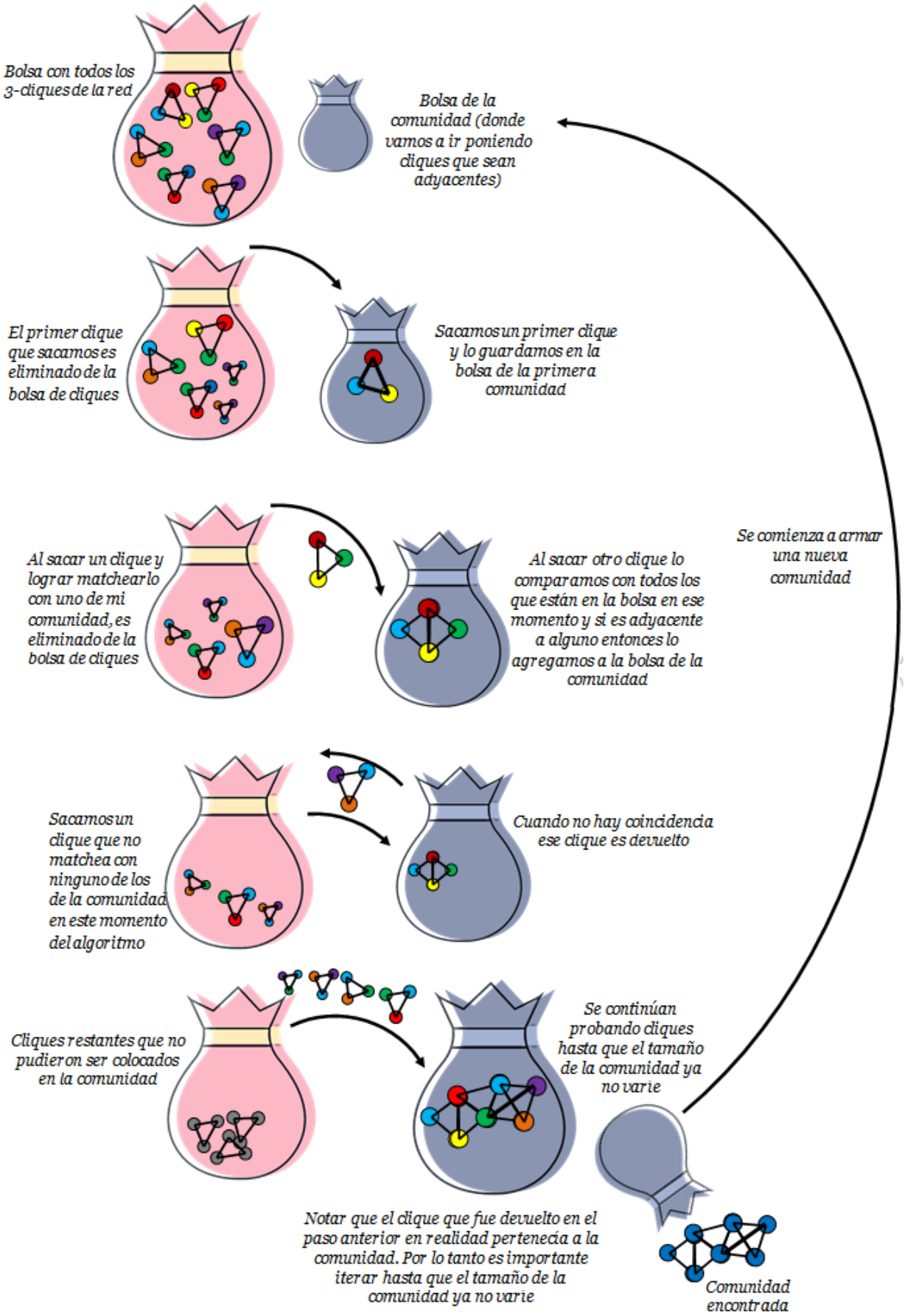


Figura 10: Algoritmo de percolación de cliques