

# Relatório de Estágio - MS877

## MODULAR: um software para computar a modularidade em grandes conjuntos de redes

**Aluna:** Flávia Maria Darcie Marquitti\*

**Orientador:** Paulo Roberto Guimarães Jr.

\* E-mail: flamarquitti@gmail.com

## Resumo

Os sistemas biológicos podem ser vistos como redes de interações entre as espécies ou indivíduos. Uma característica essencial de muitas redes biológicas é a sua organização em módulos, que são subconjuntos dos elementos mais ligados um ao outro do que a outros elementos da rede. Apresento MODULAR, um programa computacional desenvolvido para realizar o cálculo rápido e autônomo da modularidade em grandes conjuntos de redes. MODULAR lê um conjunto de arquivos texto contendo as matrizes que representam tanto redes unipartidas como bipartidas, assim como ele também lê arquivo do tipo UCINET de listas de adjacência. Isso permite que pesquisadores não familiarizados com o cálculo possam estimar modularidade em uma ampla gama de redes complexas ecológicas. O software pode identificar e definir os módulos por meio de duas métricas de modularidade diferentes e amplamente usadas em estudos de redes ecológicas. A fim de encontrar a partição de rede que maximiza a modularidade, o software oferece cinco métodos de otimização para o usuário: *simulated annealing*, *fast greedy*, decomposição espectral, além de híbridos de *fast greedy* e *simulated annealing*, e do híbrido de decomposição espectral e *simulated annealing*. Foram incluídos dois dos modelos nulos mais comuns utilizados em estudos de redes ecológicas para verificar como a modularidade encontrada pela maximização de cada medida difere de um referencial teórico. Exemplifico o uso do MODULAR com duas redes ecológicas. As redes diferem em seu nível de intimidade interação entre as espécies. Elas apresentam diferentes graus de modularidade e a rede com alto valor de modularidade é mais modular do que o esperado por interações que acontecem ao acaso entre as espécies.

## Introdução

Sistemas ecológicos podem ser vistos como redes, onde os elementos (espécies ou indivíduos) são representadas por nós e suas interações ecológicas descritas como arestas (ou *links*), que ligam os nós (Pascual & Dunne 2006). A forma com que essas redes são organizadas, ou seja, a topologia da rede, influencia a dinâmica da rede (Stouffer & Bascompte 2011) e como elas respondem a alterações como a perda de espécies (Dunne *et al.* 2002). Vários descritores topológicos têm sido usados para descrever a organização de redes ecológicas (Pascual & Dunne 2006, Bascompte & Jordano 2007). Uma das características mais importantes da topologia de rede é a modularidade, também denominada compartimentalização, *clustering*, ou estrutura da comunidade (Boccaletti *et al.* 2006). Módulos são grupos coesos de nós altamente conectados e que são frouxamente ligados a outros nós na rede (Newman & Girvan 2004, Boccaletti *et al.* 2006, Olesen *et al.* 2007) (ver Figura 2).

Em redes ecológicas, modularidade ocorre se grupos de espécies (ou indivíduos) compartilham recursos similares uns com os outros, mas com pouca sobreposição com o conjunto de recursos usados por outros grupos de espécies (ou de indivíduos) dentro da rede (Guimarães *et al.* 2007, Tinker *et al.* 2012). A organização modular foi encontrada em diferentes tipos de redes ecológicas: de redes indivíduo-recursos (Araújo *et al.* 2008), a teias alimentares que retratam quem se alimenta de quem em uma comunidade ecológica (Krause *et al.* 2003, Allesina & Pascual 2009); de redes mutualísticas que descrevem as interações entre espécies de plantas e seus polinizadores (Olesen *et al.* 2007), a redes antagonicas que descrevem as interações entre parasitas e seus hospedeiros (Fortuna *et al.* 2010). Visto que a organização modular determina como os elementos estão altamente conectados em uma rede ecológica, a modularidade influencia a dinâmica ecológica e evo-

lutiva nos sistemas ecológicos (Guimarães *et al.* 2007, Olesen *et al.* 2007, Thébault & Fontaine 2010). Além disso, como a modularidade influencia a forma como os efeitos de perturbações se espalham entre os elementos (espécies ou indivíduos) em uma rede ecológica, ela está intimamente relacionada à robustez da rede e à persistência de espécies e indivíduos (Stouffer & Bascompte 2011).

Impulsionado pela relevância de modularidade para o funcionamento do ecossistema, o problema da detecção de modularidade tornou-se uma parte importante dos estudos sobre a estrutura e dinâmica de redes ecológicas. A modularidade pode ser medida por diferentes métricas e uma das mais populares no estudo de redes biológicas é a modularidade ( $Q$ , às vezes referida como  $M$ ) (Newman & Girvan 2004), que tem sido utilizada em vários estudos em redes ecológicas e outras redes complexas (Guimerà & Amaral 2005, Olesen *et al.* 2007, Fortuna *et al.* 2010). Para uma dada partição da rede em módulos,  $Q$  mede a proporção de arestas que ligam os nós no mesmo módulo. Como buscar e encontrar a modularidade máxima ( $Q$  máximo) em uma rede é um problema NP-difícil (Ruan & Zhang 2007), portanto não há qualquer algoritmo conhecido para encontrar o valor máximo de  $Q$  em tempo polinomial. Por isso é necessário usar abordagens sub-ótimas de otimização para executar o cálculo em redes. Algoritmos heurísticos podem ser usados para uma aproximação de partição em módulos de forma a distribuir o maior número possível de arestas dentro dos módulos e o menor número possível de arestas entre os módulos, sendo uma forma de maximizar a métrica de modularidade.

Como o número de combinações possíveis de  $N$  nós em módulos segue o número de Bell de  $N$ , encontrar a partição com o maior modularidade é muitas vezes demorado para redes muito grandes. Além disso, muitas vezes os biólogos analisam grandes conjuntos de dados e testam seus resultados contra referenciais teóricos representados por conjuntos formados por milhares de réplicas (Gotelli & Graves 1996, Gotelli 2001). Neste sentido,

um dos principais entraves para a análise da modularidade em redes biológicas é a falta de programas que permitam a computação rápida e automática da modularidade de grandes conjuntos de redes por pesquisadores não familiarizados com programação. Neste trabalho, apresento o software MODULAR que calcula modularidade e identifica os módulos em várias redes complexas. Embora muitos algoritmos tenham sido propostos para encontrar a partição que maximiza o valor  $Q$  e alguns deles estejam disponíveis ao público, em meu conhecimento, não há um software disponível que permita a utilização de diferentes métricas e algoritmos de otimização de um modo que simplifique o trabalho de ecólogos. O programa MODULAR foi desenvolvido para permitir que se calcule a modularidade (Newman & Girvan 2004) automaticamente de vários arquivos de entrada e para permitir a escolha do algoritmo de otimização que supram as necessidades do usuário. Ilustro a utilidade deste novo software através da análise de duas redes ecológicas com um dos algoritmos de otimização disponível no MODULAR.

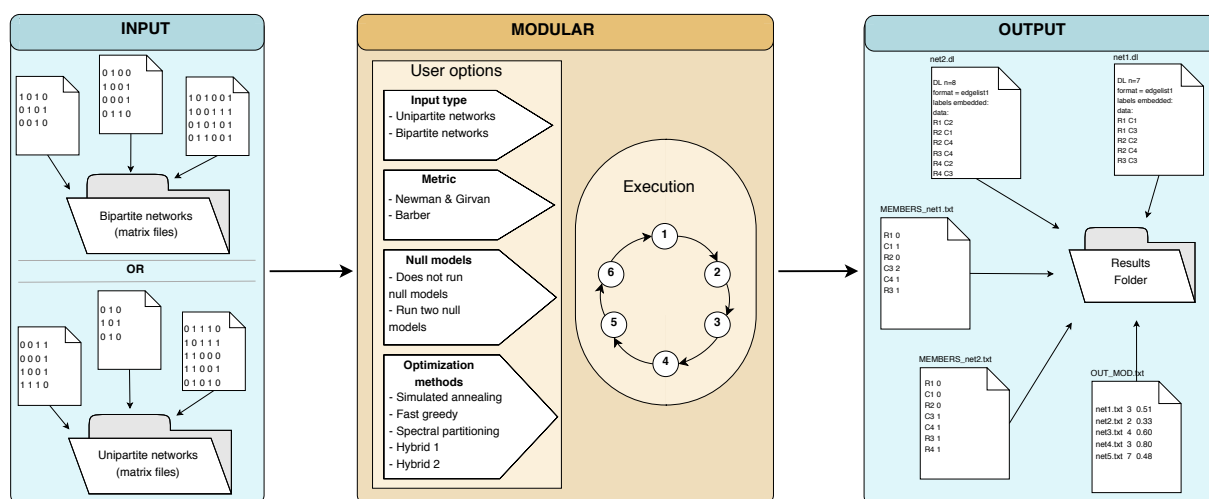
## Descrição

Nesta seção apresento uma visão geral do software MODULAR. Primeiramente, descrevo o funcionamento do software em geral, e, em seguida, descrevo como executá-lo. Por último, informo sobre as bibliotecas necessárias e detalhes dos métodos de otimização.

## Funcionamento do MODULAR

MODULAR foi desenvolvido em linguagem  $C$  e usa recursos da biblioteca IGRAPH-0.6 (Csardi & Nepusz 2006) e da biblioteca GNU Scientific Library (GSL) (Gough 2009). O funcionamento básico MODULAR está representado na Figura 1. MODULAR foi concebido para facilitar a detecção de módulos de múltiplas redes. Para conseguir isso,

a maximização da modularidade é automaticamente executada ao longo de um conjunto de arquivos de entrada contendo representações de redes bipartidas, tais como aquelas que descrevem as interações entre as plantas e frugívoros (Donatti *et al.* 2011), ou redes unipartidas, tais como as redes que descrevem como os indivíduos se sobrepõem em suas dietas (Araújo *et al.* 2008). O usuário pode executar o programa de uma vez só para tantos arquivos quanto ele queira, incluindo matrizes teóricas geradas por outros modelos, diferentes dos modelos nulos incluídos no programa.



**Figura 1.** Design do software modular. A concepção do programa pode ser dividida em três passos. O primeiro é a **preparação do arquivo de entrada**. O usuário pode escolher entre as representações de redes bipartidos ou unipartidas. O segundo passo inclui **rodar o MODULAR**. O usuário precisa responder a algumas perguntas e MODULAR que vai executar seis etapas (veja a explicação da execução no texto). Finalmente, MODULAR produzirá **arquivos de saída** com os resultados.

O conjunto de arquivos de entrada pode ser de dois tipos: matrizes em arquivos de texto (extensão .txt) ou lista de adjacência em arquivos UCINET (extensão .dl). Qualquer um dos dois tipos de arquivos de entrada pode conter representações de redes bipartidas e de redes unipartidas. Se os dados de entrada são matrizes em arquivos de texto, cada arquivo .txt é uma matriz binária em que as colunas são separadas por tabulação ou

espaço. O usuário não deve incluir rótulos de linhas e colunas no arquivo de entrada. A representação de redes unipartidas deve conter matrizes quadradas  $N \times N$  em que a linha  $i$  e coluna  $i$  representam o mesmo elemento. Estas matrizes são simétricas, e o programa lerá apenas os dados de entrada a partir da triangular superior da matriz. Por outro lado, se as matrizes em arquivos de texto representam redes bipartidas em matriz  $N \times M$ , linhas e colunas representam diferentes conjuntos de elementos, e as arestas existem apenas entre os elementos dos diferentes conjuntos. Por exemplo, as linhas podem representar animais enquanto colunas pode representar plantas em uma rede de interação animal-plantas (Jordano *et al.* 2003). Diferentemente, se os dados de entrada são arquivos UCINET, a representação das redes unipartidas deve seguir os formatos de arquivo UCINET com o campo *formato* indicando *edgelist1* seguido por uma lista de adjacência composta de pares de nós. Por outro lado, a representação das redes bipartidas deve definir no campo *formato* uma *edgelist2*, seguido por uma lista de adjacência de pares de nós. O primeiro nó do par é um elemento de um conjunto e o segundo nó é elemento do outro conjunto. Portanto, a primeira opção que o MODULAR oferece ao usuário é a seleção entre arquivos de entrada diferentes representando redes bipartidas e unipartidas.

A segunda opção dada ao usuário é em relação à métrica de modularidade. Se os arquivos de dados de entrada são a representação de redes bipartidas, o usuário pode escolher entre duas medidas diferentes: a modularidade de Newman e Girvan -  $Q$  (Newman & Girvan 2004) ou a modularidade de Barber -  $Q_B$  (Barber 2007). A medida de Barber é uma modificação da medida  $Q$  para redes de bipartidas. Ao executar MODULAR sobre representação das redes de unipartidas, apenas a medida  $Q$  está disponível (Newman & Girvan 2004). No entanto, para a análise das redes bipartidas ambas as métricas estão disponíveis, já que na literatura ecológica, redes bipartidas foram analisadas com índices para redes unipartidas (Olesen *et al.* 2007, Fortuna *et al.* 2010).

Se o usuário escolhe a tradicional medida  $Q$ , há cinco algoritmos de otimização para realizar a busca pela partição da rede em módulos que maximiza o índice de modularidade: (i) *fast greedy* (FG) (Clauset *et al.* 2004), (ii) o *simulated annealing* (SA) (Guimerà & Amaral 2005), (iii) a decomposição espectral (SP) (Newman 2006), (iv) um híbrido de *simulated annealing* e decomposição espectral (Hyb-SP), e (v) um híbrido de *simulated annealing* e *fast greedy* (Hyb-FG). Os algoritmos de otimização diferem no método de pesquisa pela partição que maximiza a medida de modularidade (que é a função objetivo). O método de pesquisa FG busca a máxima modularidade aceitando apenas novas partições apresentam maior modularidade do que a partição atual. É um método de otimização guloso e foi aprimorado por Clauset *et. al* (2004) para o caso de modularidade.

#### **Otimização FG:**

- 1) Calcula a modularidade para uma determinada partição da rede.
- 2) Muda o conjunto de nós entre as partições e re-calcula a modularidade.
- 3) Aceita a nova partição se ela fornece um valor maior de modularidade.
- 4) Pára quando não encontrar qualquer solução melhor ou quando um número máximo de iterações escolhido para executar o FG for atingido.

O método SA pesquisa o valor máximo da modularidade tentando evitar máximos locais (Kirkpatrick *et al.* 1983). Este método inicializa os módulos de rede com um nó em cada módulo, gerando assim uma configuração inicial de  $N$  módulos de tamanho 1. Depois disso, o método desloca nós aleatoriamente entre os módulos, calculando o do valor de modularidade resultante após a configuração dos novos módulos. Se a nova modularidade é maior do que na configuração de módulos anterior, o algoritmo toma a nova configuração de módulos como solução atual. Por outro lado, se a nova modularidade é menor do

que na configuração de módulos anterior, o algoritmo aceita a nova configuração com uma dada probabilidade (Metropolis *et al.* 1953). Deste forma, o programa simula um sistema de partículas, em que os níveis de energia seguem uma distribuição de Boltzmann. Com esta técnica, o algoritmo pode evitar armadilhas para máximos locais movendo através de soluções sub-ótimas e potencialmente convergindo para um ponto máximo. Evitar ficar preso em ótimos locais é possível por causa do parâmetro da temperatura, o qual está diretamente relacionado com a possibilidade de aceitar novas soluções. A temperatura diminui gradualmente ao longo do processo, diminuindo a possibilidade de aceitar soluções piores (menores valores de modularidade). Os passos são repetidos até que a temperatura atinja um limiar SA ou ocorra um número de iterações sem quaisquer alterações na solução. O método é resumido a seguir:

**Otimização SA:**

- 1) Calcula a modularidade para uma determinada partição da rede.
- 2) Gera uma nova partição trocando nós ao acaso e fundindo/ dividindo módulos; calcula a modularidade.
- 3) Aceita a nova partição com uma dada probabilidade, de acordo com a diferença entre os novos valores e os valores antigos de modularidade e de acordo com a temperatura atual.
- 4) Para se a solução não mudar depois de um número pré-determinado de iterações, ou quando o número máximo de iterações escolhido para executar o SA é atingido. Caso contrário, reduz a temperatura utilizando um fator de redução de temperatura do SA e repete os passos.

O método SP baseia-se nos espectros de matriz. Este método divide a rede em



seqüências de grupos de acordo com as entradas do autovetor dominante (Clauset *et al.* 2004).

### **Otimização SP:**

- 1) Estabelece a matriz de modularidade (matriz B, veja Newman 2006).
- 2) Divide a rede em dois grupos, de acordo com as entradas no autovetor dominante.
- 3) Repete o processo até que todas as sub-matrizes não tenham nenhum autovalor positivo.

Apresento em MODULAR mais dois métodos híbridos, combinando o baixo tempo de execução das abordagens SP e FG com o maior espaço de busca do SA. Nas abordagens híbridas, a entrada do SA é a saída do método mais rápido (SP ou FG). Ao fazer isso, a otimização pode alcançar melhores resultados. Se o usuário optar por maximizar a métrica de Barber, apenas o método de SA está disponível na versão atual.

A fim de verificar como a modularidade encontrada pela maximização de cada medida difere de um referencial teórico, o usuário tem a opção de executar dois modelos nulos diferentes, especificando quantas repetições cada modelo nulo irá gerar. Foram incluídos dois dos modelos nulos mais comuns utilizados em estudos de redes ecológicas: (i) o modelo Erdős e Rényi (Erdős & Rényi 1959), e (ii) o “modelo nulo 2” (Bascompte *et al.* 2003). O tempo de execução pode variar consideravelmente e, conseqüentemente, a escolha do algoritmo de otimização torna-se especialmente importante. Como MODULAR pode executar um grande conjunto de dados de entrada, o usuário pode testar a modularidade de redes teóricas geradas em outros programas e por modelos nulos distintos, utilizando-as como dado de entrada.

Com base nas opções acima, MODULAR executa iterativamente até seis etapas de execução (“Execution”, na Figura 1 para cada arquivo de entrada *.txt* ou *.dl* presentes

nas pasta especificada pelo usuário:

Para cada *.txt* ou *.dl* arquivo na pasta especificada, faz:

1. Lê a matriz  $A$  ou a lista de adjacência do arquivo de entrada de acordo com o tipo especificado (representação de uma rede bipartida ou unipartida);
2. Se o arquivo de entrada é uma matriz, remove todas as linhas e colunas nulas presentes na matriz, gerando uma nova matriz  $A'$ ;
3. Se o arquivo de entrada é uma matriz, gera um arquivo *it.dl* do *UCINET* (Borgatti *et al.* 2002) com a rede representada pela matriz  $A'$ ;
4. Lê o arquivo *.dl*, carregando a rede para a memória do computador em estruturas de dados que podem ser manipuladas pelas funções da biblioteca *IGRAPH*;
5. Maximiza a modularidade da rede, executando o método escolhido pelo usuário (SA, SP, FG, Hyb-SP, ou Hyb-FG);
6. Executa os modelos nulos para a rede, se solicitado pelo usuário.

Após essas etapas, um conjunto de arquivos de saída é gerado. Há três arquivos de saída principais. Se os arquivos de entrada são matrizes e *.txt*, e cada arquivo de entrada é chamado *MatrixName.txt*, o programa MODULAR gera um arquivo com a lista de adjacência chamado *MatrixName.dl* no formato UCINET. A primeira linha tem o número de nós na rede. Se MODULAR está executando para representações de redes bipartidas, as linhas serão nomeadas pela letra  $R$  seguida de um número (por exemplo,  $R2$  para a segunda linha do arquivo de entrada) e colunas pela letra  $C$  seguida por um número (por exemplo,  $C3$  para a terceira coluna do arquivo de entrada). Se o conjunto de dados é composto por representações de rede unipartidas, as linhas serão nomeados

pela letra  $L$  seguida de um número (por exemplo,  $L2$  para a segunda linha do arquivo de entrada). O número refere-se à ordem no arquivo de entrada, em qualquer caso.

A principal saída é o arquivo *OUT\_MOD.txt*. Ele contém, para cada arquivo de entrada, o valor de modularidade e número de módulos encontrados pela métrica e algoritmo escolhidos. Se o usuário optar por executar os modelos nulos, o arquivo *OUT\_MOD.txt* conterà também a proporção de matrizes nulas com valor maior do que o valor da modularidade do arquivo de entrada. Um arquivo de saída chamado *MEMBERS\_MatrixName.txt* é gerado para cada arquivo de entrada *MatrixName.txt*. Ele contém duas colunas: a primeira coluna apresenta os rótulos das linhas da matriz a partir do arquivo *.dl*, enquanto que a segunda coluna tem o módulo ao qual cada linha pertence. Se a análise de modelo nulo é executada, haverá ainda duas saídas adicionais: *OUT\_1\_MatrixName.txt*, com todos os valores de modularidade para o modelo nulo 1, e *OUT\_2\_MatrixName.txt*, com todos os valores de modularidade do modelo nulo 2. Ambos os arquivos têm tantas linhas quanto o número de modelos nulos escolhidos pelo usuário.

## Linguagem, Bibliotecas e Algoritmos

Implantei o software MODULAR em linguagem  $C$ . O código depende de duas bibliotecas: IGRAPH e o GNU Scientific Library (GSL). A biblioteca IGRAPH proporciona estruturas de dados para representação de grafos e funções de manipulação, bem como as funções de detecção da estrutura da comunidade. A biblioteca GSL é usada principalmente para gerar números aleatórios necessários para os métodos de otimização. Com a implementação de um conjunto de funções que utilizam essas duas bibliotecas, foi possível fazer um software fácil de usar para a detecção de módulos através da maximização de modularidade. A biblioteca IGRAPH utilizada é a versão 0.6, que inclui suporte para arquivo UCINET do formato DL. Usei a função *IGRAPH\_read\_graph\_dl* para ler a rede em uma estrutura de

dados de grafos. Além disso, usei estrutura de dados chamada *membership\_vectors* para representar os módulos da rede. O método SA (Kirkpatrick *et al.* 1983) foi implementado utilizando uma estrutura de dados da GSL, que contém os parâmetros de funcionamento, tais como a temperatura inicial e o seu fator de redução da temperatura. O método *fast greedy* (Clauset *et al.* 2004) e a *decomposição espectral* (Newman 2006) são funções chamadas a partir da biblioteca IGRAPH. Estes dois métodos são mais rápidos do que o *simulated annealing* mas as suas pesquisas baseiam-se as decisões locais, conduzindo a valores de modularidade inferiores ao do SA. Em MODULAR, combino a velocidade desses dois métodos para encontrar uma solução razoável, com a pesquisa mais ampla fornecida pelo método SA, criando dois métodos híbridos. A primeira abordagem híbrida (*Hyb-SP*) executa primeiro o método SP, e sua configuração de módulo de saída é utilizada como entrada do método SA, mas com uma temperatura inicial reduzida. A mesma técnica é aplicada no segundo método híbrido, utilizando como primeiro algoritmo o método FG (*Hyb-FG*).

## Exemplo

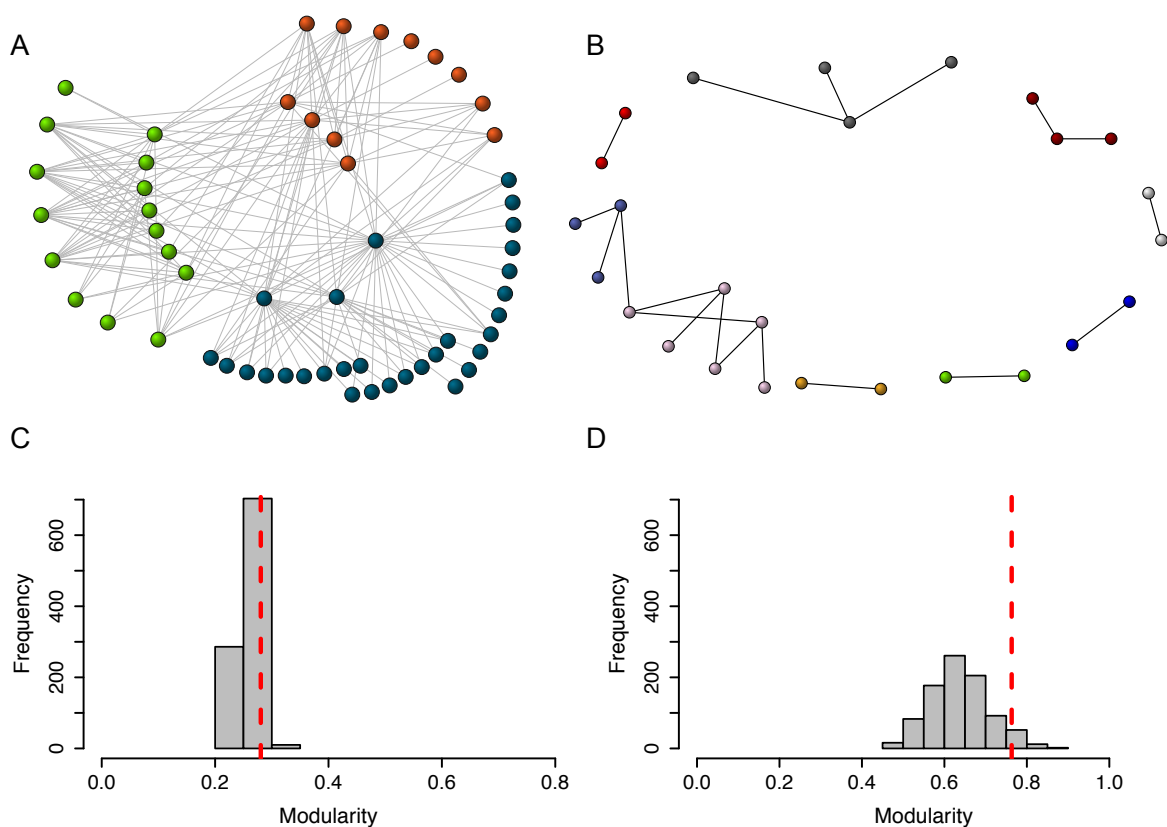
Ilustro o uso de MODULAR analisando duas redes ecológicas. As redes descrevem antagonismos, que são interações ecológicas que têm conseqüências positivas para a aptidão de um dos indivíduos que interagem e um efeito negativo sobre os outros indivíduos que interagem. Apesar de ambas as redes descreverem antagonismos, eles variam em seu grau de intimidade interação, isto é, o grau de integração biológica entre indivíduos que interagem. Há evidências disponíveis na literatura apontando que interações ecológicas com diferentes graus de intimidade interação variam em seu grau de modularidade (Guimarães *et al.* 2007, Fontaine *et al.* 2011, Pires & Guimarães 2012). A primeira rede descreve as

interações entre espécies gafanhotos e as plantas que consomem (Ueckert & Hansen 1971). É uma interação de baixa intimidade já que os consumidores são capazes de alimentar-se muitas plantas diferentes ao longo de suas vidas. A segunda rede descreve as interações entre insetos galhadores e plantas (Tscharrntke *et al.* 2001). Esta é uma interação com um elevado grau de intimidade, uma vez que existe uma dependência trófica dos consumidores em hospedeiros individuais, pelo menos em parte do ciclo de vida do consumidor. Em ambos os casos, os arquivos de entrada foram matrizes binárias representando redes bipartidas.

Calculei a modularidade usando a métrica  $Q$ . Executei MODULAR usando o algoritmo de otimização SA. Gerei 1.000 redes utilizando o “modelo nulo 2” para testar se as redes são mais modular do que o esperado para redes aleatórias com heterogeneidade similar nas interações entre espécies. Tal como previsto pelas diferenças biológicas entre as redes, a rede de insetos galhadores foi mais modular ( $Q = 0,76$ ) do que as redes teóricas geradas pelo modelo nulo ( $p = 0,05$ ). O grau de modularidade encontrado para a rede gafanhoto-planta foi semelhante ao encontrado para as redes teóricas. ( $Q = 0,28, p = 0,11$ , Figura 2).

## Estudos futuros

Como direções futuras, posso incluir no programa MODULAR novos algoritmos e otimizações para calcular a modularidade, assim como métricas de modularidade relacionadas aos nós das redes.



**Figura 2.** A modularidade de interações antagonísticas. Redes que descrevem (A) interações entre as plantas e gafanhotos (Ueckert & Hansen 1971) e (B) insetos galhadores e plantas (Tschardt *et al.* 2001). Os nós de diferentes cores são atribuídos a diferentes módulos. (C-D) A significância do grau de modularidade encontrado para cada rede pode ser testada comparando o valor de modularidade observado, linha vermelha, e a distribuição dos valores de modularidade para redes geradas por um modo de nulo.

## Agradecimento

Agradeço ao meu orientador Paulo Roberto Guimarães Jr. por todo suporte e dedicação durante meu período de estudos na graduação em Matemática Aplicada. Sem sua compreensão e paciência teria sido muito difícil conseguir completar meus estudos e desenvolver

minha pesquisa.

## Referências

- Allesina, S. & Pascual, M. (2009) Food web models: a plea for groups. *Ecology letters*, **12**, 652–662.
- Araújo, M.S., Guimarães Jr, P.R., Svanbäck, R., Pinheiro, A., Guimarães, P., Reis, S.F. & Bolnick, D.I. (2008) Network analysis reveals contrasting effects of intraspecific competition on individual vs. population diets. *Ecology*, **89**, 1981–1993.
- Barber, M.J. (2007) Modularity and community detection in bipartite networks. *Physical Review E*, **76**, 066102.
- Bascompte, J. & Jordano, P. (2007) Plant-animal mutualistic networks: the architecture of biodiversity. *Annual Review of Ecology, Evolution, and Systematics*, **38**, 567–593.
- Bascompte, J., Jordano, P., Melián, C.J. & Olesen, J.M. (2003) The nested assembly of plant-animal mutualistic networks. *Proceedings of the National Academy of Sciences of the United States of America*, **100**, 9383–9387.
- Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D.U. (2006) Complex networks: Structure and dynamics. *Physics Reports*, **424**, 175–308.
- Borgatti, S.P., Everett, M.G. & Freeman, L.C. (2002) Ucinet for windows: Software for social network analysis. Technical report, Analytic Technologies, Harvard, MA.
- Clauset, A., Newman, M.E.J. & Moore, C. (2004) Finding community structure in very large networks. *Physical Review E*, **70**, 066111.

- Csardi, G. & Nepusz, T. (2006) The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695.
- Donatti, C.I., Guimarães, P.R., Galetti, M., Pizo, M.A., Marquitti, F.M.D. & Dirzo, R. (2011) Analysis of a hyper-diverse seed dispersal network: modularity and underlying mechanisms. *Ecology letters*, **14**, 773–781.
- Dunne, J.A., Williams, R.J. & Martinez, N.D. (2002) Network structure and biodiversity loss in food webs: robustness increases with connectance. *Ecology Letters*, **5**, 558–567.
- Erdős, P. & Rényi, A. (1959) On random graphs. *Publicationes Mathematicae Debrecen*, **6**, 290–297.
- Fontaine, C., Guimarães, P.R., Kéfi, S., Loeuille, N., Memmott, J., Van Der Putten, W.H., Van Veen, F.J.F. & Thébault, E. (2011) The ecological and evolutionary implications of merging different types of networks. *Ecology letters*, **14**, 1170–1181.
- Fortuna, M.A., Stouffer, D.B., Olesen, J.M., Jordano, P., Mouillot, D., Krasnov, B.R., Poulin, R. & Bascompte, J. (2010) Nestedness versus modularity in ecological networks: two sides of the same coin? *Journal of Animal Ecology*, **79**, 811–817.
- Gotelli, N.J. (2001) Research frontiers in null model analysis. *Global Ecology and Biogeography*, **10**, 337–343.
- Gotelli, N.J. & Graves, G.R. (1996) *Null models in ecology*. Smithsonian Institution Press.
- Gough, B. (2009) *GNU Scientific Library Reference Manual - Third Edition*. Network Theory Ltd., 3rd edition.



- Guimarães, P.R., Rico-Gray, V., Oliveira, P.S., Izzo, T.J., dos Reis, S.F. & Thompson, J.N. (2007) Interaction intimacy affects structure and coevolutionary dynamics in mutualistic networks. *Current Biology*, **17**, 1797–803.
- Guimerà, R. & Amaral, L.A.N. (2005) Functional cartography of complex metabolic networks. *Nature*, **433**, 895–900.
- Jordano, P., Bascompte, J. & Olesen, J.M. (2003) Invariant properties in coevolutionary networks of plant–animal interactions. *Ecology Letters*, **6**, 69–81.
- Kirkpatrick, S., Gelatt, C.D. & Vecchi, M.P. (1983) Optimization by simulated annealing. *Science*, **220**, 671–680.
- Krause, A.E., Frank, K.A., Mason, D.M., Ulanowicz, R.E. & Taylor, W.W. (2003) Compartments revealed in food-web structure. *Nature*, **426**, 282–285.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. & Teller, E. (1953) Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**, 1087–1093.
- Newman, M.E.J. (2006) Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, **74**, 036104.
- Newman, M.E.J. & Girvan, M. (2004) Finding and evaluating community structure in networks. *Physical Review E*, **69**, 026113.
- Olesen, J.M., Bascompte, J., Dupont, Y.L. & Jordano, P. (2007) The modularity of pollination networks. *Proceedings of the National Academy of Sciences of the United States of America*, **104**, 19891–19896.

- Pascual, M. & Dunne, J.A. (2006) *Ecological networks: linking structure to dynamics in food webs*. Oxford University Press, USA.
- Pires, M.M. & Guimarães, P.R. (2012) Interaction intimacy organizes networks of antagonistic interactions in different ways. *Journal of The Royal Society Interface*.
- Ruan, J. & Zhang, W. (2007) An efficient spectral algorithm for network community discovery and its applications to biological and social networks. *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pp. 643–648. Ieee.
- Stouffer, D.B. & Bascompte, J. (2011) Compartmentalization increases food-web persistence. *Proceedings of the National Academy of Sciences*, **108**, 3648–3652.
- Thébault, E. & Fontaine, C. (2010) Stability of ecological communities and the architecture of mutualistic and trophic networks. *Science*, **329**, 853–856.
- Tinker, M.T., Guimarães Jr, P.R., Novak, M., Marquitti, F.M.D., Bodkin, J.L., Staedler, M., Bentall, G. & Estes, J.A. (2012) Structure and mechanism of diet specialisation: testing models of individual variation in resource use with sea otters. *Ecology Letters*, **15**, 475–483.
- Tscharntke, T., Vidal, S. & Hawkins, B.A. (2001) Parasitoids of grass-feeding chalcid wasps: a comparison of german and british communities. *Oecologia*, **129**, 445–451.
- Ueckert, D.N. & Hansen, R.M. (1971) Dietary overlap of grasshoppers on sandhill rangeland in northeastern colorado. *Oecologia*, **8**, 276–295.