フィールドインフォマティクス講義

「フィールドアグリオミクスにより有機農業を科学する」

２０１９月6月4日

理化学研究所バイオリソース研究センター
植物－微生物共生研究開発チーム
市橋 泰範 チームリーダー

講義内容

- 植物－微生物共生研究開発チーム紹介
- 研究例の紹介「フィールドアグリオミクスにより有機農業を科学する」
- 相関ネットワーク解析の実習

植物－微生物共生研究開発チーム紹介

植物の根圏は、地球上で最も微生物が豊富な生態系であり、多様な植物種と共生して土壌中の養分を根に供給する菌根菌などの有用な微生物が存在しています。植物－微生物の共生関係の理解が進めば、地球規模での持続可能な食料供給や環境負荷軽減への大きな貢献が期待できます。本チームは、植物－微生物共生研究に資する根圏微生物および実験植物のリソース開発と、これを活用した植物－微生物共生の実験系の確立、更には農業現場への応用に資する情報整備を行います。内外の研究コミュニティと緊密に連携することにより、共生現象の実態解明と産業利用につながる世界トップクラスの成果をめざします。

具体的な研究テーマ：

- 微小液滴による根圏微生物の新規培養技術の開発
- アーバスキュラー菌根菌のリソース研究
- 植物－微生物共生におけるモデル実験系の確立
- 次世代型農業を目指したフィールドアグリオミクス研究

◆ メンバーとして参加や共同研究にご興味があれば、ご連絡ください：
https://pms.brc.riken.jp/ja/contact

フィールドアグリオミクスにより有機農業を科学する

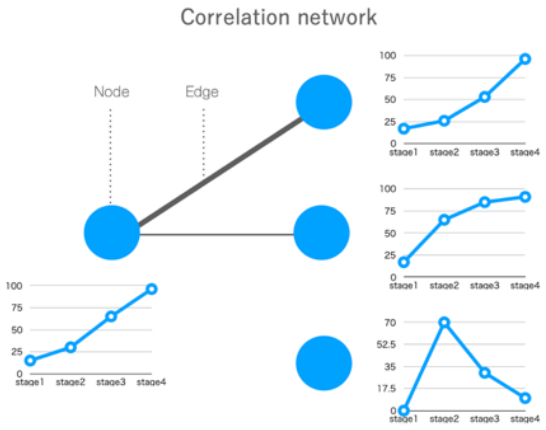
近年、有機農業は世界的な食料および環境問題解決に関して注目されています。有機農法の一つである太陽熱処理は、土壌中の植物病原体を防除する方法として開発されたが、植物病原体がいない場合でも植物の成長を促進する効果が知られています。その成長促進効果のメカニズムを明らかにするため、私たちは同じ圃場内において異なる農法でコマツナを栽培し、全代謝物解析および微生物プロファイル解析によるフィールドアグリオミクス研究を行いました。その結果、太陽熱処理が土壌全体の微生物でなく、コマツナの根圏に生息する微生物の種類に大きく影響することがわかりました。またコマツナや土壌、根圏微生物など全てのデータを統合した相関ネットワーク解析を行うと、個別の解析では明らかにできなかった収量に強く相関した有機態窒素（アミノ酸など）および根圏微生物が明らかになりました。得られた結果に基づき無菌培養実験により、栄養源や生理活性物質として収量を増加させる特定の有機態窒素があることを実証しました。本研究のデータは、慣行農法を支える無機栄養説とは異なり、有機態窒素を利用した農法の有効性を示唆するものでした。

相関ネットワーク解析の実習

上記の研究例で紹介した相関ネットワーク解析手法を実習します。上記のデータは未発表なので、すでに公開されているデータを用いて R の WGCNA と igraph パッケージを使用した相関ネットワーク解析の手法を説明します。

1）*ネットワークとは？*

ノード（Node）の集合とエッジ（Edge）の集合で構成されるグラフ。例えば、共発現遺伝子ネットワークでは、ノードが遺伝子、エッジが遺伝子間の共発現。



２）*使用するデータ*

フィールドでのオミクスデータとして Ichihashi et al., 2018 の研究から Supplementary Data にある”pcp-2017-e-00401-File011.csv”を使用します。

ここではフィールドに自生する寄生植物カナビキソウから、吸器（寄生する際に宿主植物の体内に侵入して栄養分を吸収する特殊な器官）と根（吸器は根からリプログラミングして形成するため対照サンプルとして）をサンプリングして取得した RNA-seq データについて、*de novo* assembly、mapping、normalization 等の前処理済みのデータを本解析に用います（詳細は論文を参照）。

◆ 論文 URL：
https://academic.oup.com/pcp/article/59/4/729/4769293#125131457
◆ データおよび R スクリプトのリンク：
https://briefcase.riken.jp/public/Kj44gA0s6gTASfE

３）*使用する R のパッケージ*

WGCNA：Weighted correlation network analysis (WGCNA)
◆ パッケージ URL：
https://horvath.genetics.ucla.edu/html/CoexpressionNetwork/Rpackages/WGCNA/

igraph：Network analysis and visualization
◆ パッケージ URL：
https://igraph.org

それぞれのパッケージのインストールは R/R Studio の R パッケージインストーラ/Install Packages の CRAN からインストールできます。

４）*相関ネットワーク解析の実行*

Working directory を指定し、パッケージを読み込みます。

```
> setwd("<working_directory>")
> library(WGCNA)
```

```
> library(igraph)
```

データを読み込み、必要なデータを抽出します。

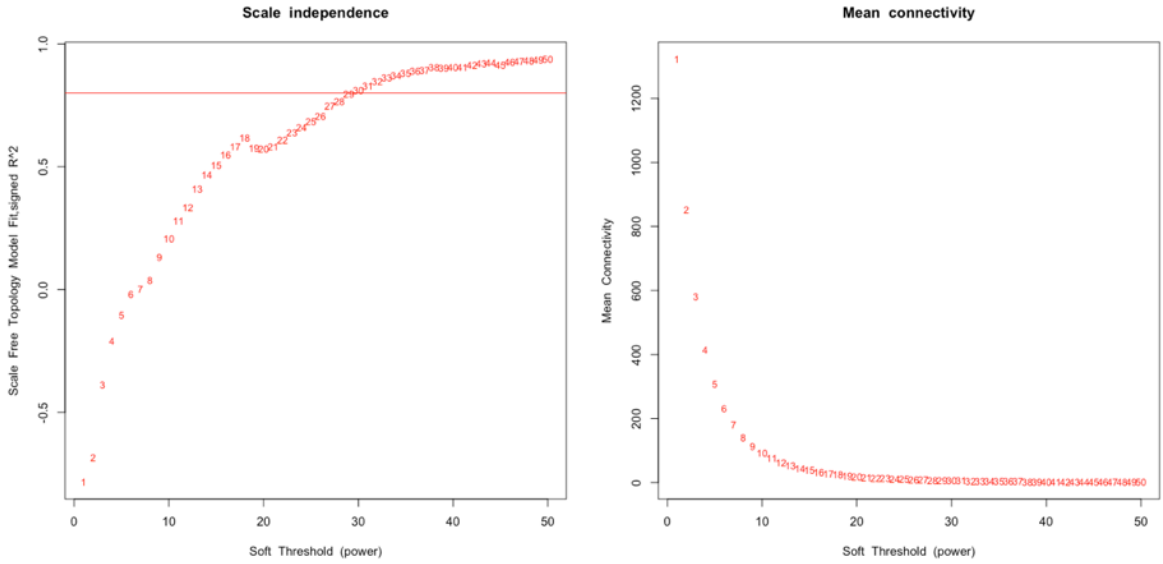
```
> data <- read.csv("pcp-2017-e-00401-File011.csv",row.names=1)
> data <- data[c(5:16)]
> dim(data)
[1] 2265   12
```

ネットワークのエッジは、生物学的なネットワークを反映させるために Soft-thresholding によって決めます。現実世界のネットワークが持つ第 1 の性質であるスケールフリー性 (次数分布のべき乗則、一部の頂点が他のたくさんの頂点と辺で繋がっており、大きな次数を持っている一方で、その他の大部分はわずかな頂点としか繋がっておらず、次数は小さいという性質)が適切になるような Soft-threshold power β を求めます。今回は、β を 1~50 の範囲から、model fitting index R^2 が 0.8 よりも高くなる値を探索します。

```
> powers  =  c(1:50)
> sft=pickSoftThreshold(t(data), powerVector=powers, networkType = "unsigned",
RsquaredCut=0.8, verbose=5)
pickSoftThreshold: will use block size 2265.
 pickSoftThreshold: calculating connectivity for given powers...
   ..working on genes 1 through 2265 of 2265
   Power SFT.R.sq   slope truncated.R.sq mean.k. median.k. max.k.
1      1  0.78400  5.6200          0.951 1320.00  1340.000 1690.0
2      2  0.68400  2.8500          0.955  852.00   855.000 1290.0
3      3  0.38900  1.4200          0.965  583.00   578.000  995.0
(省略)


> tiff("SoftThresholding.tif", width=16, height=8, unit="in",compression="lzw",res=100)
> par(mfrow  =  c(1,2))
> cex1  =  0.8
> # Scale-free topology fit index as a function of the soft-thresholding power
> plot(sft$fitIndices[,1],  -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
```

```
+      xlab="Soft  Threshold  (power)",ylab="Scale  Free  Topology  Model  Fit,signed
R^2",type="n",
+      main  =  paste("Scale  independence"));
> text(sft$fitIndices[,1],  -sign(sft$fitIndices[,3])*sft$fitIndices[,2],
+      labels=powers,cex=cex1,col="red");
> #  this  line  corresponds  to  using  an  R^2  cut-off  of  h
> abline(h=0.8,col="red")
> #  Mean  connectivity  as  a  function  of  the  soft-thresholding  power
> plot(sft$fitIndices[,1],  sft$fitIndices[,5],
+      xlab="Soft  Threshold  (power)",ylab="Mean  Connectivity",  type="n",
+      main  =  paste("Mean  connectivity"))
> text(sft$fitIndices[,1],  sft$fitIndices[,5],  labels=powers,  cex=cex1,col="red")
> dev.off()
null device
          1
> sft$powerEstimate
[1] 30
```

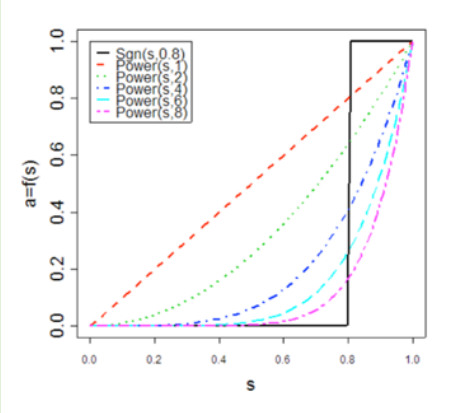
以下のような図が出力されます。β が 30 のときに model fitting index R^2 が 0.8 よりも高くなることがわかります。



補足：Soft-thresholding について

To transform the similarity matrix into an adjacency matrix, one needs to define an adjacency function. This choice determines whether the resulting network will be weighted (soft-thresholding) or unweighted (hard thresholding). Most biologists would be very suspicious of a gene co-expression network that does not satisfy scale-free topology at least approximately. Therefore, a soft threshold power β (in $a(i,j)=|cor(x(i),x(j))|^{\beta}$) that give rise to a network that does not satisfy approximate scale-free topology should not be considered. There is a natural trade-off between maximizing scale-free topology model fit (scale free fitting parameter R^2) and maintaining a high mean number of connections. High values of β often lead to high values of R^2. But the higher the power β, the lower is the mean connectivity of the network.

These considerations have motivated to propose the following scale-free topology criterion for choosing the power β: Only consider those powers that lead to a network satisfying scale-free topology at least approximately, e.g. R^2>0.80. In addition, the users can take the following additional considerations into account when choosing the adjacency function parameter. First, the mean connectivity should be high so that the network contains enough information (e.g. for module detection). Second, the slope of the regression line between log(p(k)) and log(k) should be negative (typically smaller than -2). In practice, the relationship between R^2 and β is characterized by a saturation curve. In most applications, it is goot to use the lowest power β where saturation is reached.



Power and signum adjacency functions. The value of the adjacency function (y-axis) is plotted as a function of the similarity (co-expression measure). Note that the adjacency function maps the interval [0,1] into [0,1].

上記で算出したβを使って、隣接行列を作ります。その際、topological overlap matrix (TOM)に変換することで、見せかけ・単独のエッジを除去します。

```
> adjacency  =  adjacency(t(data),power=sft$powerEstimate)

> TOM =  TOMsimilarity(adjacency)

..connectivity..

..matrix multiplication..

..normalization..

..done.

> id=rownames(data)

> colnames(TOM)=id

> rownames(TOM)=id
```

作成した隣接行列を可視化するために、igraph を使用します。その際、ネットワークの特徴的な構造のみに着目するために、TOM similarity > 0.1 でフィルタリングします。

```
> subTOM = (TOM>0.1)*TOM

> subnet=graph.adjacency(subTOM,mode="undirected",weighted=TRUE,diag=FALSE)

> summary(subnet)

IGRAPH e6c71dd UNW- 2265 12528 --

+ attr: name (v/c), weight (e/n)
```

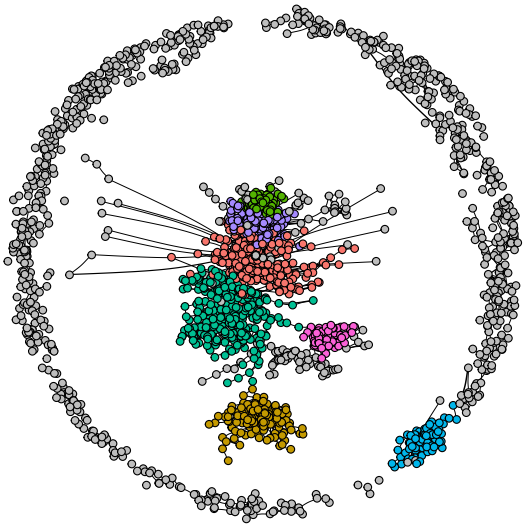
ネットワークのモジュール構造を Fast greedy アルゴリズムによって計算します。モジュール 1 から順にどれくらいのノード（遺伝子）が入っているか表示させることができます。

```
> community.fastgreedy=fastgreedy.community(subnet)

> table(community.fastgreedy$membership)

  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17

223 138 160 281  78 197  90  48  18   7   2   3   5   2   2   2   4

（省略）
```

今回は 50 ノード（遺伝子）以上入っているモジュールにそれぞれ異なる色をつけてネットワークを可視化します。

```
> # ggplot color
> ggplotColours <- function(n=6, h=c(0, 360) +15){
+   if ((diff(h)%%360) < 1) h[2] <- h[2] - 360/n
+   hcl(h = (seq(h[1], h[2], length = n)), c = 100, l = 65)
+ }
> ggplotColours(7)
[1] "#F8766D" "#C49A00" "#53B400" "#00C094" "#00B6EB" "#A58AFF"
[7] "#FB61D7"
> # visualization
> set.seed(123)
> V(subnet)$color <- "grey"
> V(subnet)[community.fastgreedy$membership==1]$color <- ggplotColours(7)[1]
> V(subnet)[community.fastgreedy$membership==2]$color <- ggplotColours(7)[2]
> V(subnet)[community.fastgreedy$membership==3]$color <- ggplotColours(7)[3]
> V(subnet)[community.fastgreedy$membership==4]$color <- ggplotColours(7)[4]
> V(subnet)[community.fastgreedy$membership==5]$color <- ggplotColours(7)[5]
> V(subnet)[community.fastgreedy$membership==6]$color <- ggplotColours(7)[6]
> V(subnet)[community.fastgreedy$membership==7]$color <- ggplotColours(7)[7]
> v.label=rep("",length(V(subnet)))
> v.label=V(subnet)$name
> v.size=rep(3,length(V(subnet)))
> V(subnet)$shape <- "circle"
> l <- layout_with_kk(subnet)
> l <- layout.norm(l, ymin=-1, ymax=1, xmin=-1, xmax=1)
> pdf("Network.pdf", useDingbats=FALSE)
> plot.igraph(subnet, layout=l,rescale=F,
edge.curved=.1,vertex.size=v.size,vertex.label=v.label,
vertex.label.cex=0.05,edge.color="black", edge.width=E(subnet)$weight*4)
> dev.off()
null device
          1
```

以下のような図が出力されます（v.label=V(subnet)$name を#によりコメントアウトして、遺伝子名を表示させずに作図してあります)。それぞれのモジュールにどういったノード（遺伝子）が入っているか、その機能からモジュールの特徴付けをすることで（例えば、共発現遺伝子ネットワーク解析であれば、Gene ontology 解析や Gene set enrichment 解析等)、ネットワーク全体でどのようなモジュール構造があるか機能的な側面を議論することができます。



続いて、個々のモジュール内のノード間の関係性をより詳細に調べるために、モジュールを抽出して、いくつかネットワーク統計量で色分けをすることができます。今回はモジュール 1 を抜き出して strength（個々のノードに対して近接するエッジを積算）をノードの大きさと色で可視化します。

```
> m1 <- induced_subgraph(subnet,community.fastgreedy$membership==1)
> fine = 500 # this will adjust the resolving power.
> palette = colorRampPalette(c('magenta','green'))
> graphCol = palette(fine)[as.numeric(cut(strength(m1),breaks = fine))]


> V(m1)$color <- graphCol
> v.label=rep("",length(V(m1)))
> v.label=V(m1)$name
> v.size=log(strength(m1)*100)
> V(m1)$shape <- "circle"
> l <- layout_with_kk(m1) #, repulserad=vcount(m1)^3, area=vcount(m1)^2.4)
```

```
> l <- layout.norm(l, ymin=-1, ymax=1, xmin=-1, xmax=1)
> pdf("Module1.pdf", useDingbats=FALSE)
> plot.igraph(m1, layout=l*1, rescale=F, edge.curved=.2,
vertex.size=v.size,vertex.label=v.label, vertex.label.cex=0.05,edge.color="black",
edge.width=E(m1)$weight*4)
> dev.off()
null device
          1
```

以下のような図が出力されます。Strength を可視化させることでどのノードがより多くのノードとつながっているのか、すなわちハブとしての機能を持つか議論することができます。