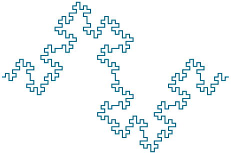
High-Throughput Sequencing Course

Gene Co-expression Network Analysis

Biostatistics and Bioinformatics



Summer 2018

Duke University School of Medicine

Duke Biostatistics & Bioinformatics

Section 1

Introduction

Gene Co-expression Network (GCN)

- GCN is a undirected graph.
- Each node represents a gene.
- Edge between nodes implies there is a significant co-expression relationship between them.

Gene Co-expression Network (GCN)

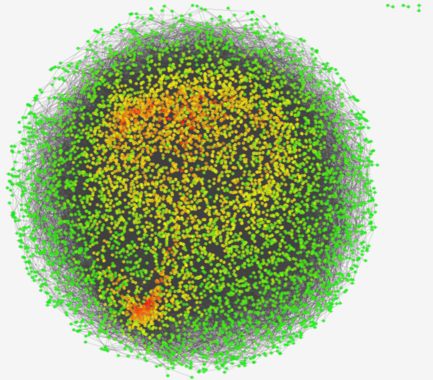


Figure: A gene co-expression network constructed from a microarray dataset containing gene expression profiles of 7221 genes for 18 gastric cancer patients (Created by S. Mohammad H. Oloomi).

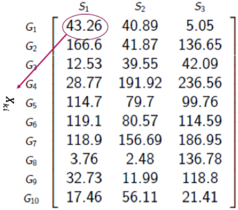
Section 2

GCN 101

Microarray and Linear Dependence

- Gene microarray data: X is an $n \times N$ data matrix, n subjects, N genes.
- Calculate the Pearson correlation matrix $\hat{\Sigma} = \widehat{\mathsf{Cor}}(X)$.
- Threshold the absolute value of Pearson correlations.

A Toy Example



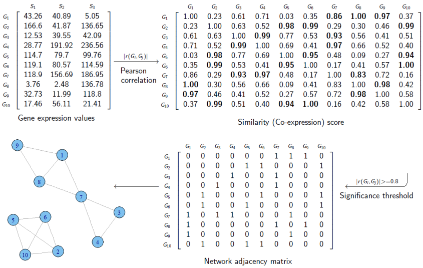
$$
\begin{array}{c|ccc}
 & S_1 & S_2 & S_3 \\
\hline
G_1 & 43.26 & 40.89 & 5.05 \\
G_2 & 166.6 & 41.87 & 136.65 \\
G_3 & 12.53 & 39.55 & 42.09 \\
G_4 & 28.77 & 191.92 & 236.56 \\
G_5 & 114.7 & 79.7 & 99.76 \\
G_6 & 119.1 & 80.57 & 114.59 \\
G_7 & 118.9 & 156.69 & 186.95 \\
G_8 & 3.76 & 2.48 & 136.78 \\
G_9 & 32.73 & 11.99 & 118.8 \\
G_{10} & 17.46 & 56.11 & 21.41
\end{array}
$$

Gene expression values

Pearson correlation:

$$r(G_i, G_j) = \frac{\frac{1}{n}\sum_{k=1}^{n}(X_{ki} - \bar{X}_i)(X_{kj} - \bar{X}_j)}{\{\frac{1}{n}\sum_{k=1}^{n}(X_{ki} - \bar{X}_i)^2\}^{1/2}\{\frac{1}{n}\sum_{k=1}^{n}(X_{kj} - \bar{X}_j)^2\}^{1/2}}$$

Illustration of Correlation Thresholding



	S_1	S_2	S_3
G_1	43.26	40.89	5.05
G_2	166.6	41.87	136.65
G_3	12.53	39.55	42.09
G_4	28.77	191.92	236.56
G_5	114.7	79.7	99.76
G_6	119.1	80.57	114.59
G_7	118.9	156.69	186.95
G_8	3.76	2.48	136.78
G_9	32.73	11.99	118.8
G_{10}	17.46	56.11	21.41

Gene expression values

$\xrightarrow{|r(G_i,G_j)|}$ Pearson correlation

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
G_1	1.00	0.23	0.61	0.71	0.03	0.35	**0.86**	**1.00**	**0.97**	0.37
G_2	0.23	1.00	0.63	0.52	**0.98**	**0.99**	0.29	0.30	0.46	**0.99**
G_3	0.61	0.63	1.00	**0.99**	0.77	0.53	**0.93**	0.56	0.41	0.51
G_4	0.71	0.52	**0.99**	1.00	0.69	0.41	**0.97**	0.66	0.52	0.40
G_5	0.03	**0.98**	0.77	0.69	1.00	**0.95**	0.48	0.09	0.27	**0.94**
G_6	0.35	**0.99**	0.53	0.41	**0.95**	1.00	0.17	0.41	0.57	**1.00**
G_7	0.86	0.29	**0.93**	**0.97**	0.48	0.17	1.00	**0.83**	0.72	0.16
G_8	**1.00**	0.30	0.56	0.66	0.09	0.41	0.83	1.00	**0.98**	0.42
G_9	**0.97**	0.46	0.41	0.52	0.27	0.57	0.72	**0.98**	1.00	0.58
G_{10}	0.37	**0.99**	0.51	0.40	**0.94**	**1.00**	0.16	0.42	0.58	1.00

Similarity (Co-expression) score

$|r(G_i,G_j)|>=0.8$

Significance threshold

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}
G_1	0	0	0	0	0	0	1	1	1	0
G_2	0	0	0	0	1	1	0	0	0	1
G_3	0	0	0	1	0	0	1	0	0	0
G_4	0	0	1	0	0	0	1	0	0	0
G_5	0	1	0	0	0	1	0	0	0	1
G_6	0	1	0	0	1	0	0	0	0	1
G_7	1	0	1	1	0	0	0	1	0	0
G_8	1	0	0	0	0	0	1	0	1	0
G_9	1	0	0	0	0	0	0	1	0	0
G_{10}	0	1	0	0	1	1	0	0	0	0

Network adjacency matrix

Figure: The two general steps for constructing a gene co-expression network: calculating co-expression score (*e.g.*, the absolute value of Pearson correlation coefficient) for each pair of genes, and selecting a significance threshold (*e.g.*, correlation > 0.8) (Created by S. Mohammad H. Oloomi).

Fisher Transformation

- Transform $r(G_i, G_j)$ to Z_{ij} so that
  - Z_{ij} is monotone with $r(G_i, G_j)$.
  - Z_{ij} asymptotically converges to Gaussian distribution.
- Fisher transformation: $Z_{ij} = \frac{1}{2} \ln \left(\frac{1+r(G_i,G_j)}{1-r(G_i,G_j)} \right)$.

How to Choose the Threshold?

$$|Z_{ij}| > \tau = \sqrt{2\ln\{p(p-1)\}/(n-3)}.$$

Here, p is the number of genes and n is the sample size.

Rationale:

- $m = p(p-1)/2$ is the total number of gene pairs
- If $Z_1, \ldots, Z_m$ (random errors) independently follows $\mathrm{N}(0, 1/(n-3))$, the largest among them is

$$\approx \sqrt{2\ln\{p(p-1)/2\}/(n-3)}.$$

In practice, this threshold is too conservative (too few edges!)

Type I Error Rate

$\mathrm{H}_{\mathrm{nul},ij}$: Gene i and Gene j are independent.

	Claim significant	Claim non-significant	Total
True nulls	N_{00}	N_{01}	m_0
False nulls	N_{10}	N_{11}	m_1
Total	R	$m - R$	m

- FDR $= \mathsf{E}(N_{00}/(R \vee 1))$.
- FWER $= \mathsf{P}(N_{00} \geq 1)$.

Benjamini and Hochberg (BH) Procedure (Benjamini and Hochberg, 1995)

- Let $T_{ij} = n^{1/2} Z_{ij}$
- Let P-values: $pv_{ij} = 2 - 2\Phi(|T_{ij}|)$.
- Let $m = p(p-1)/2$. Rank the P-values from the smallest to the largest, denoted by

$$\text{PV}_{(1)} \leq \text{PV}_{(2)} \leq \ldots \leq \text{PV}_{(m)}$$

- Let $k = \max\{j : \text{PV}_{(j)} \leq \alpha j/m\}$l
- Reject $\text{H}_{\text{nul},(j)}$, $1 \leq j \leq k$.

Practice: BH Procedure

P-values:
0.003, 0.012, 0.014, 0.1, 0.15, 0.34, 0.45, 0.78, 0.86, 0.91, 0.97

Section 3

GCN 201

RNA-Seq Data

- RNA-seq data: read counts mapping to the reference genome
- Two properties:
  - The presence of extreme values
  - The mean-variance dependence

Raw versus Expected Counts

Problem of using raw counts:

- The origin of some reads cannot always be uniquely determined.
- If two or more distinct transcripts in a particular sample share some common sequence (*e.g.*, if they are alternatively spliced mRNAs or mRNAs derived from paralogous genes), then sequence alignment may not be sufficient to discriminate the true origin of reads mapping to these transcripts.

Raw versus Expected Counts

Solutions:

- discarding these multiple-mapped reads (multireads for short) entirely
- partitioning and distributing portions of a multiread's expression value between all of the transcripts to which it maps ("rescue" method)
- RSEM (B. Li and Dewey, 2011) improves upon this approach, utilizing an Expectation-Maximization (EM) algorithm to estimate maximum likelihood expression levels.

Transform RNA-Seq Data

- Log transformation:
  - $X = \log_2(\text{Data} + 1)$
- Variance stabilization transformation (VST) (Anders and Huber, 2010)
  - Assume data follow negative binomial distribution
  - Estimate the dispersion parameter first
  - Transform the data so that the variance of the transformed data is independent of the mean.

Example: Scale I

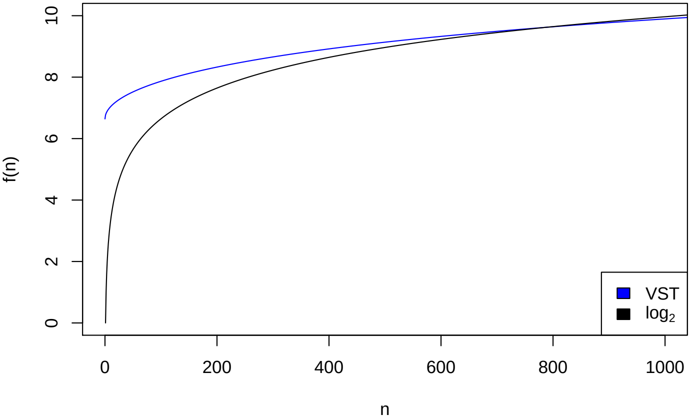
- Plot the VST and $\log_2$ transformation (x-axis shows the RSEM counts).
- Graphs of the variance stabilizing transformation for sample 1, in blue, and of the transformation $f(n) = \log_2(n/s1)$ in black, where n is the count and s_1 is the size factor for the first sample.

```r
library(DESeq)
vst <- function(countdata) {
    condition <- factor(rep("Tumour", ncol(countdata)))
    countdata <- newCountDataSet(countdata, condition)
    countdata <- estimateSizeFactors(countdata)
    cdsBlind <- DESeq::estimateDispersions(countdata, method = "blind")
    vstdata <- varianceStabilizingTransformation(cdsBlind)
    return(exprs(vstdata))
}

data <- read.csv("Data/rnaseq_lusc_example_SeqQC.csv", header = TRUE)
data.log2 <- log2(data + 1)
data.vst <- vst(data)
```

Example: Scale II

```r
condition <- factor(rep("Tumour", ncol(data)))
countdata <- newCountDataSet(data, condition)
countdata <- estimateSizeFactors(countdata)
px <- counts(countdata)[, 2]
ord <- order(px)

par(mfrow = c(1, 1))
matplot(px[ord], cbind(data.vst[, 2], log2(px))[ord, ], type = "l", lty = 1,
    col = c("blue", "black"), xlab = "n", ylab = "f(n)", xlim = c(0, 1000),
    ylim = c(0, 10))
legend("bottomright", legend = c(expression("VST"), expression(log[2])), fill = c("blue",
    "black"))
```

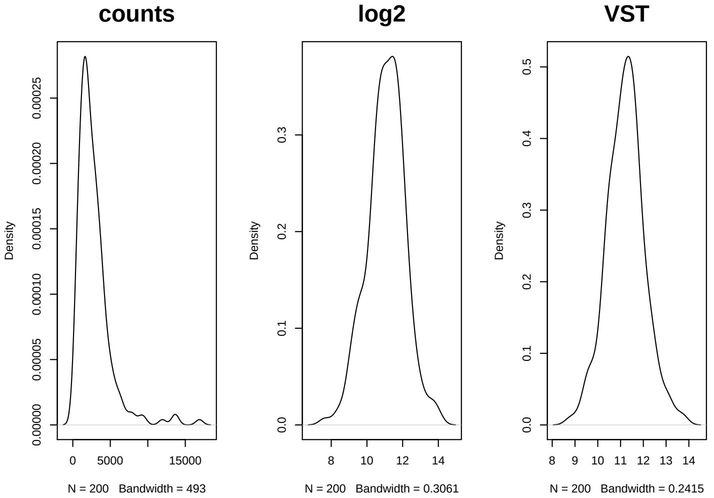
Example: Scale III



Example: Normality I

```
par(mfrow = c(1, 3))
plot(density(as.numeric(data[2, ])), main = "counts", cex.main = 2)
plot(density(as.numeric(data.log2[2, ])), main = "log2", cex.main = 2)
plot(density(as.numeric(data.vst[2, ])), main = "VST", cex.main = 2)
```

Example: Normality II



Heteroscedasticity

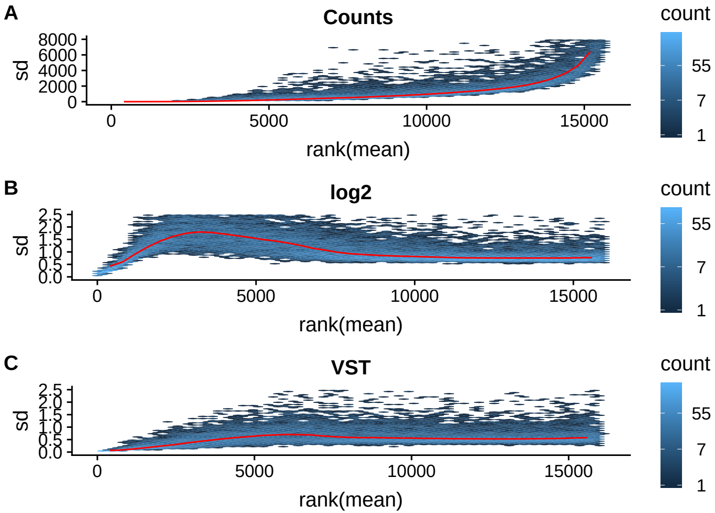
- Homoscedasticity: having the same scatter (variance)
- Heteroscedasticity: having the different scatter (variance)
  - In RNA-Seq data, genes with larger average expression have on average larger observed variance across samples, that is, they vary in expression from sample to sample more than other genes with lower average expression.

Example: Heteroscedasticity I

```r
# Mean-sd plot
library(ggplot2)
library(vsn)
p1 <- meanSdPlot(as.matrix(data))$gg + ylim(0, 8000) + ggtitle("Counts")
p2 <- meanSdPlot(as.matrix(data.log2))$gg + ylim(0, 2.5) + ggtitle("log2")
p3 <- meanSdPlot(as.matrix(data.vst))$gg + ylim(0, 2.5) + ggtitle("VST")
```

```r
library("gridExtra")
library("cowplot")
plot_grid(p1, p2, p3, labels = c("A", "B", "C"), ncol = 1, nrow = 3)
```

Example: Heteroscedasticity II



Log versus VST

A few things to consider:

- After the log transformation, there are less extreme values when compared to untransformed data, but there are still unequal variances.
- After VST, the per-gene standard deviation becomes more constant along the whole dynamic range, but note that the variance are still unequal for all genes.
- An additional problem of the log2 transformation is that $\log_2$ of zero is infinite! To avoid taking the logarithm of zero it is common to add a pseudo value of 1 prior taking the log. And, of course, we have to assume that adding 1 does not bias much the low non-zero counts.

Section 4

GCN 301

Problems form Transformation

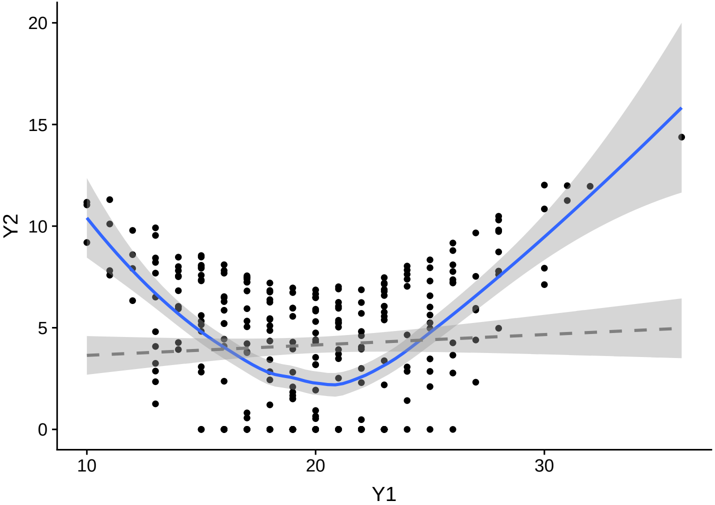
- Which one to choose?
- Transformation may introduce bias
- Transformation may cause loss of information

Is it possible to use the RSEM to infer the gene co-expression pattern?

Example: Non-linear Dependence I

```
set.seed(314)
n = 300
Y1 = rpois(n, lambda = 20)
Y2 = (Y1 - 20)^2 + runif(n, min = -50, max = 50)
Y2 = sqrt(Y2 * (Y2 >= 0))
r = cor(Y1, Y2)
r


## [1] 0.06641


fisher.z = log((1 + r)/(1 - r))/2
pv = 2 * (1 - pnorm(abs(fisher.z), mean = 0, sd = sqrt(1/(n - 3))))
pv


## [1] 0.2517


expr = data.frame(Y1, Y2)
p = ggplot(data = expr, aes(Y1, Y2))
p + geom_point() + geom_smooth(method = "loess") + geom_smooth(method = "lm",
    lty = "dashed", col = gray(0.5))
```

Example: Non-linear Dependence II



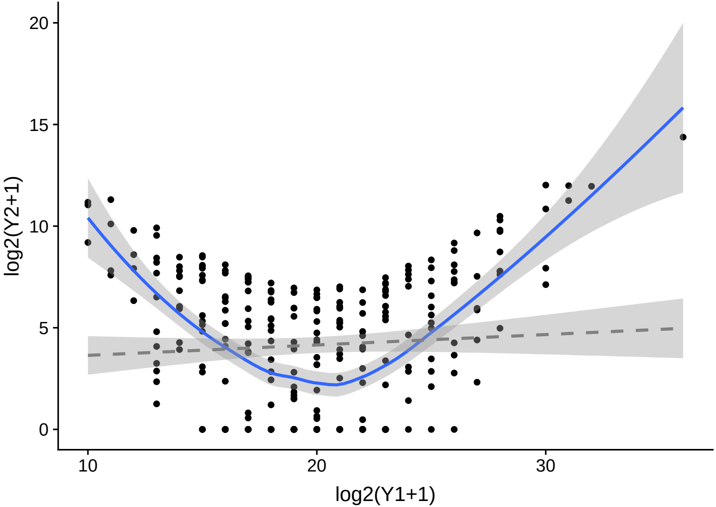
Example: Log transformation I

```r
expr.log2 = log2(expr + 1)
cor(expr.log2$Y1, expr.log2$Y2)
```

```
## [1] -0.04233
```

```r
p.log2 = ggplot(data = expr, aes(Y1, Y2))
p.log2 + geom_point() + geom_smooth(method = "loess") + geom_smooth(method = "lm",
    lty = "dashed", col = gray(0.5)) + xlab("log2(Y1+1)") + ylab("log2(Y2+1)")
```

Example: Log transformation II



Contigency Table I

```
library(arules)
Y1c = discretize(Y1, breaks = 3)
levels(Y1c) = c("low", "median", "high")
Y2c = discretize(Y2, breaks = 3)
levels(Y2c) = c("low", "median", "high")
expr = cbind(expr, Y1c, Y2c)
tbl = table(Y1c, Y2c)
tbl
```

```
##         Y2c
## Y1c      low median high
##   low     22     28   45
##   median  47     39   14
##   high    31     33   41
```

Chi-Square Test of Independence

For the cell in row r and column c,

- O_{rc}: the count
- E_{rc}: the expected number of count under independence $\frac{(\sum_{r=1}^{R} O_{rc})(\sum_{c=1}^{C} O_{rc})}{N}$.
- Discrepancy: $(O_{rc} - E_{rc})^2/E_{rc}$.

Chi-square test statistic:

$$T = \sum_{r=1}^{R} \sum_{c=1}^{C} \frac{(O_{rc} - E_{rc})^2}{E_{rc}}$$

Example: Chi-Square Test of Independence

```
chisq.test(tbl)


##
##  Pearson's Chi-squared test
##
## data:  tbl
## X-squared = 28, df = 4, p-value = 1e-05
```

More about Chi-square Test of Independence

- can extend to adjust library size and covariates
- can extend to more adaptively choose the quantile levels
- SQUAC method: Xie and R. Li (2018)

Section 5

References

- Anders, Simon and Wolfgang Huber (2010). “Differential expression analysis for sequence count data”. In: *Genome Biol* 11.10, R106. DOI: `10.1186/gb-2010-11-10-r106`.
- Benjamini, Y. and Y. Hochberg (1995). “Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1, pp. 289–300. ISSN: 00359246. URL: `http://www.jstor.org/stable/2346101`.
- Li, Bo and Colin N Dewey (Aug. 2011). “RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome”. In: *BMC Bioinformatics* 12, p. 323. DOI: `10.1186/1471-2105-12-323`.
- Xie, Jichun and Ruosha Li (July 2018). “False discovery rate control for high dimensional networks of quantile associations conditioning on covariates”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. ISSN:

1369-7412. DOI: 10.1111/rssb.12288. URL:
http://dx.doi.org/10.1111/rssb.12288.