

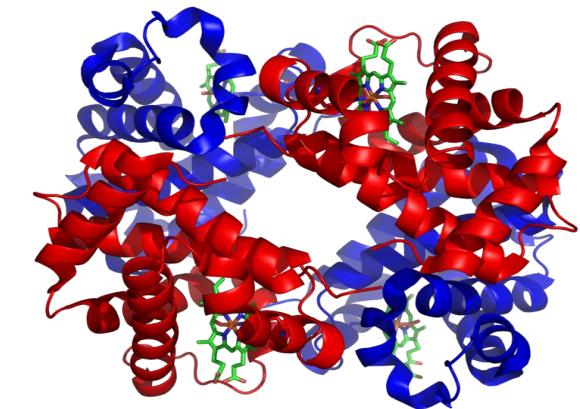
Probabilistic Network Inference

Guillermo de Anda Jáuregui
INMEGEN / CONAHCYT / C3-UNAM
[@gdeandajauregui](https://twitter.com/gdeandajauregui)

Today

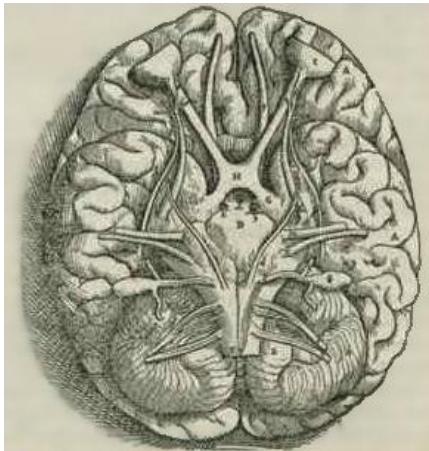
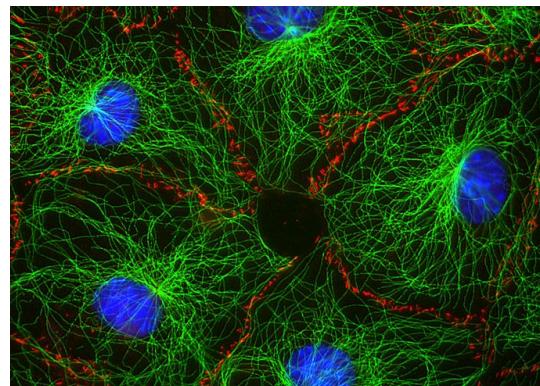
- Why use networks to model correlated data
- How to generate networks from data
- Things we can learn from probabilistically inferred networks

Life is complex...

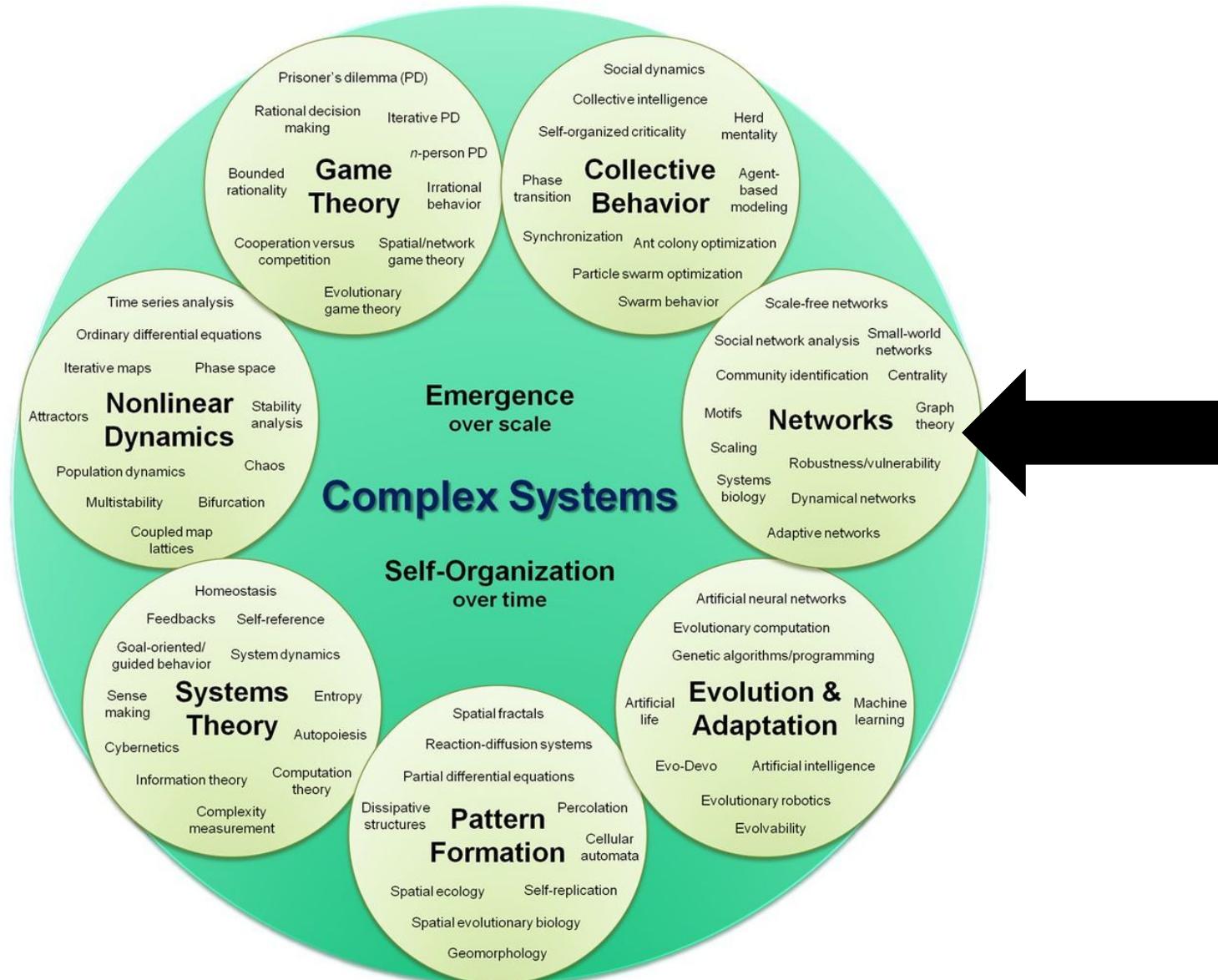


"There's no love in a carbon atom,
No hurricane in a water molecule,
No financial collapse in a dollar bill."

Peter Dodds

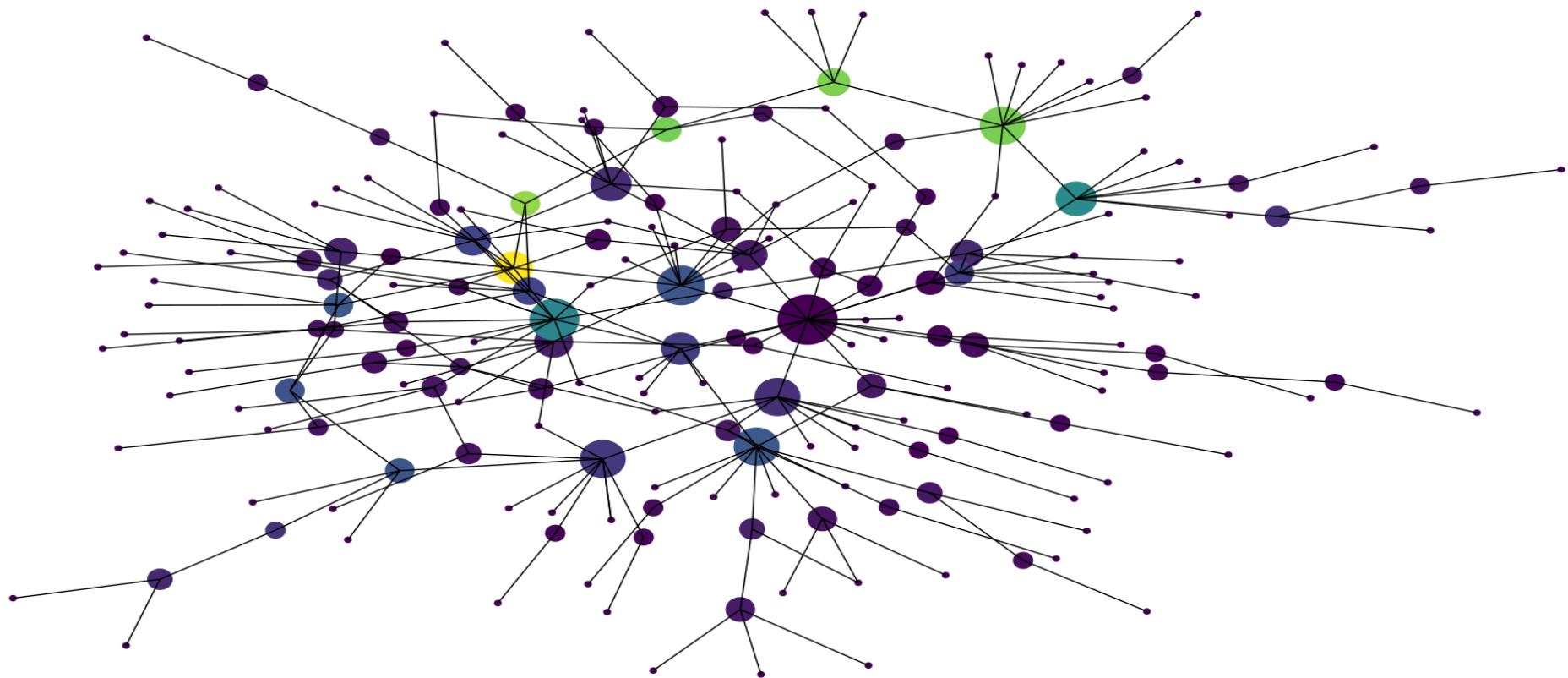




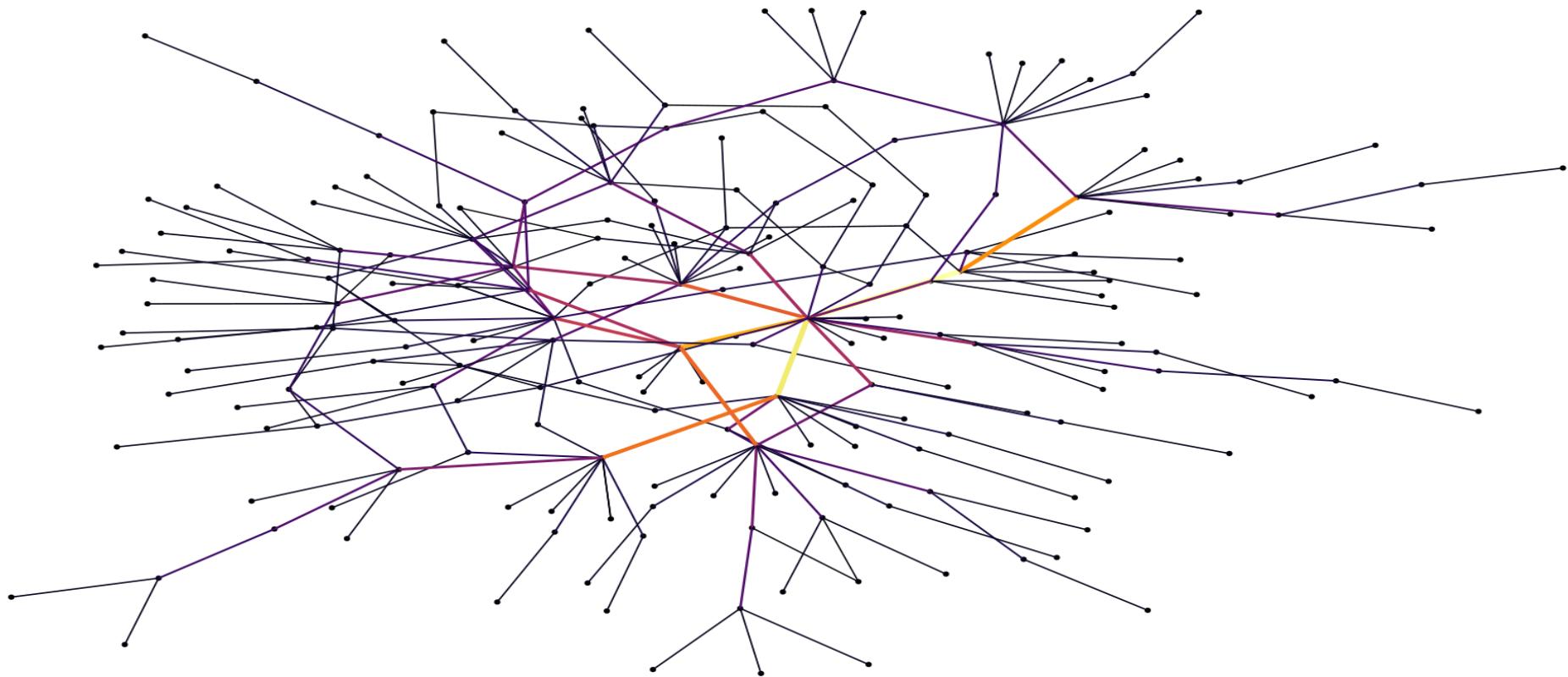


Networks are cool

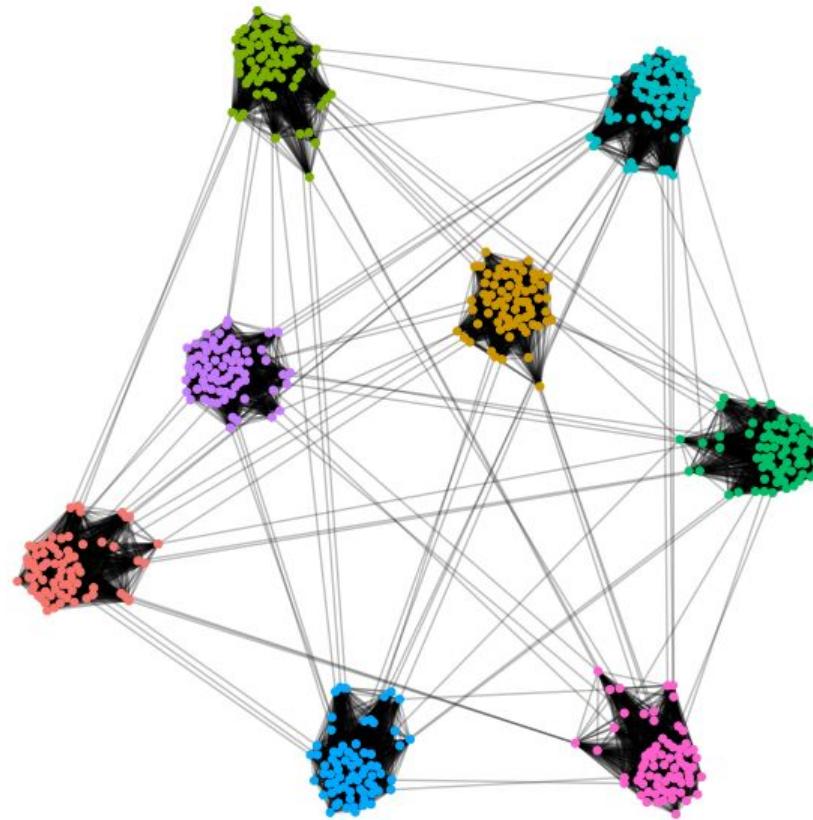
We may learn about the elements of a system...



...and how they are connected



Plus, about the whole system, itself



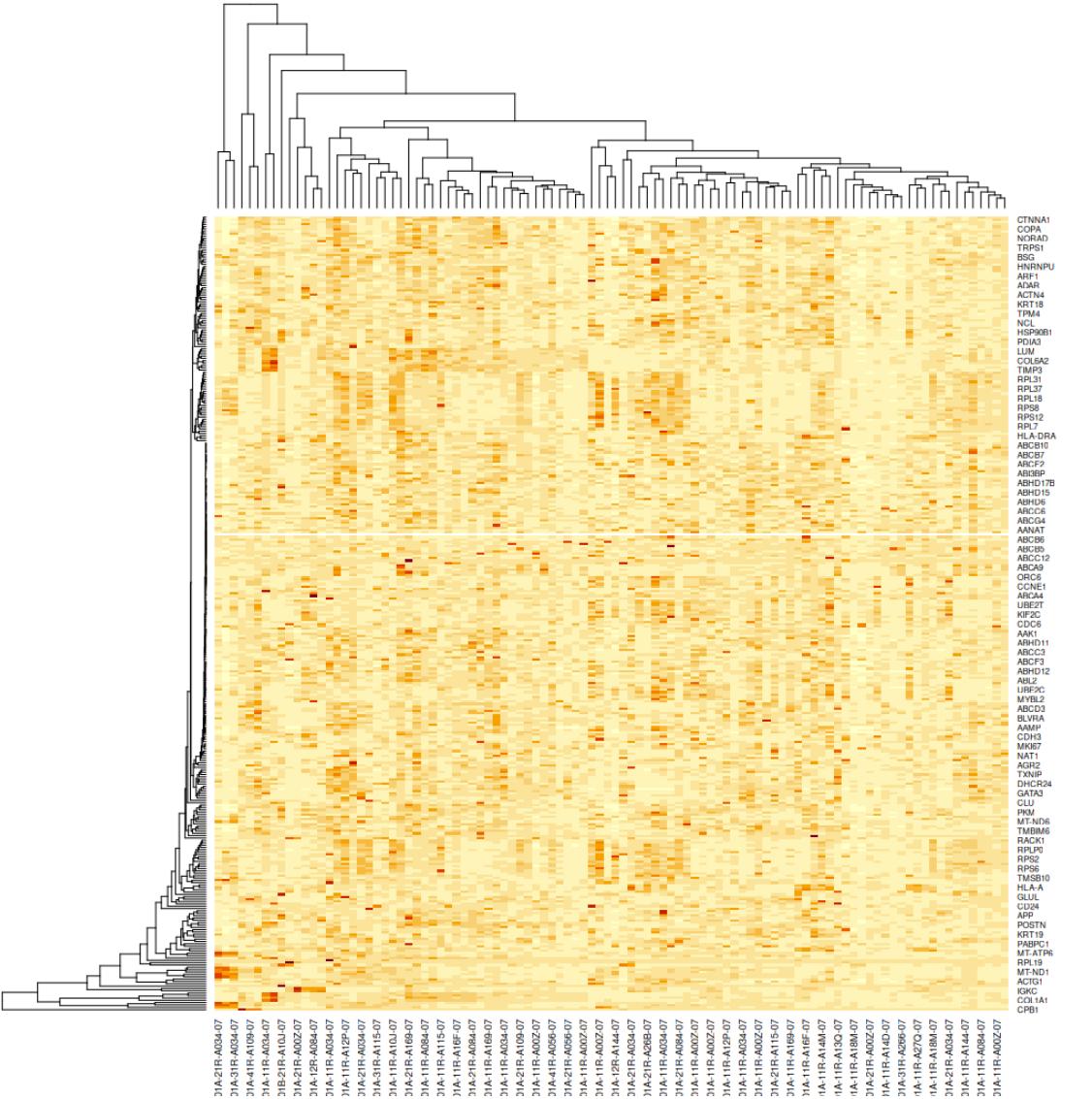
Networks in my data?

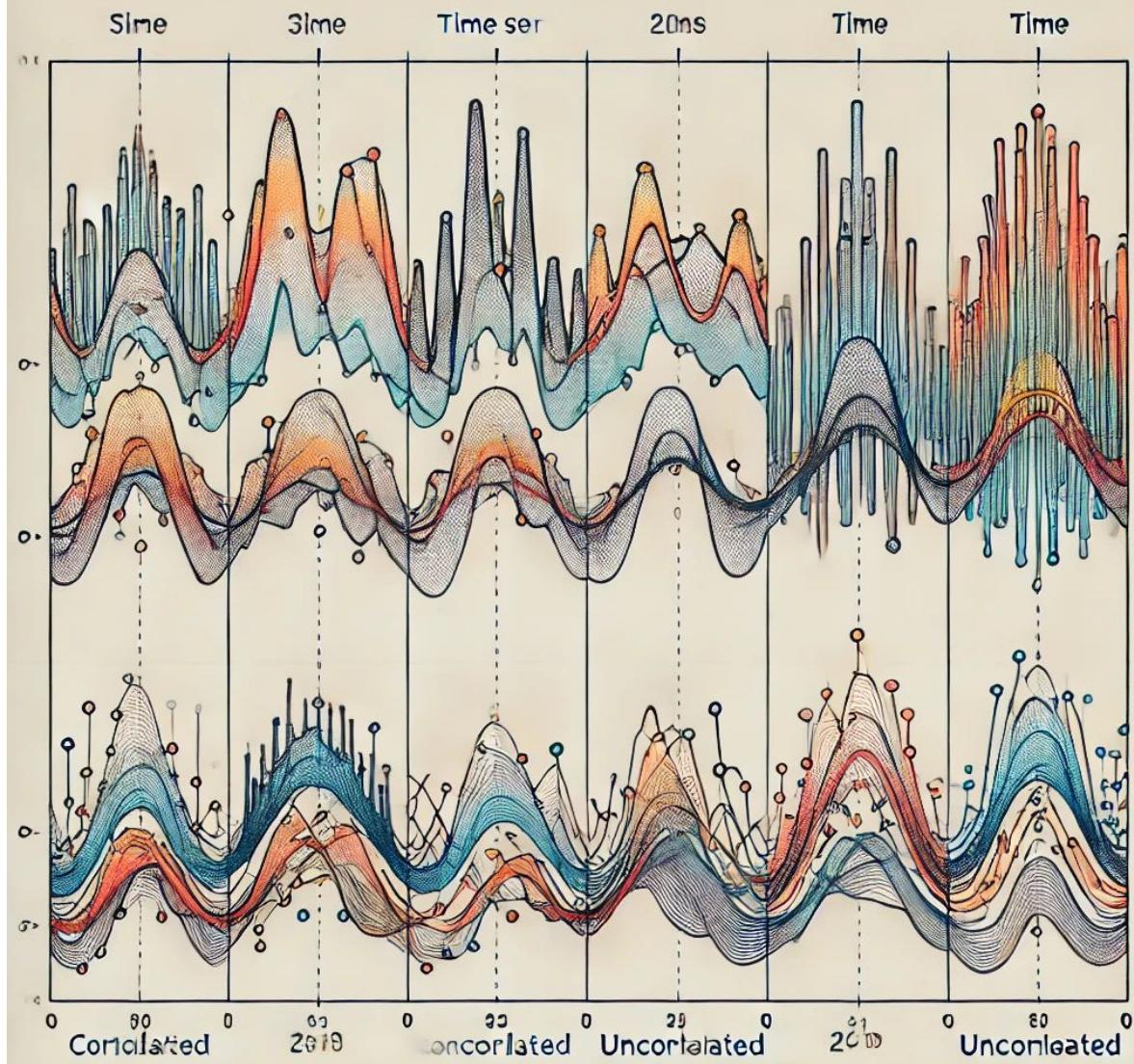




facebook®

But...





Identify statistical dependencies...
And model as networks

What's the best measure of statistical dependency?

Pearson correlation

$$\rho_{X,Y} = \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

Spearman correlation

$$r_s = \rho [R[X], R[Y]] = \frac{\text{cov} [R[X], R[Y]]}{\sigma_{R[X]} \sigma_{R[Y]}},$$

Mutual information

$$I(x_i^h, x_j^k) = \sum_{\Omega} \sum_{\Omega'} p(x_i^h, x_j^k) \log \frac{p(x_i^h, x_j^k)}{p(x_i^h) p(x_j^k)}$$

Markov Random Fields

Unknown real probability density functions?

Use Empirical Probability Distributions (Big Data)

Glivenko-Cantelli theorem (uniform convergence)

$$\|F_n - F\|_{\infty} = \sup_{x \in \mathbb{R}} |F_n(x) - F(x)| \rightarrow 0 \quad a.s.$$

Glivenko 1933, Cantelli 1933



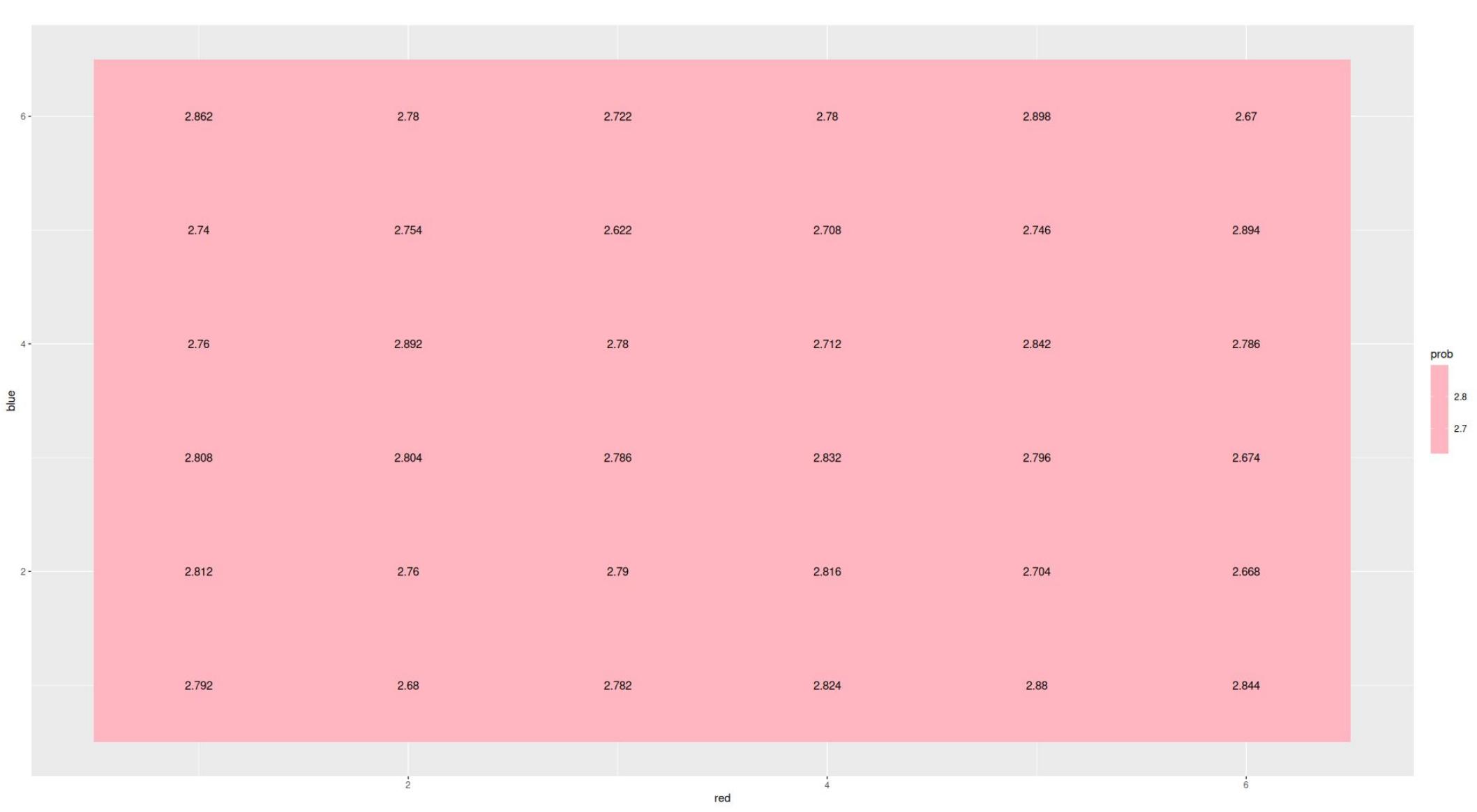
```
# excercise 1: random variables (dice) ----

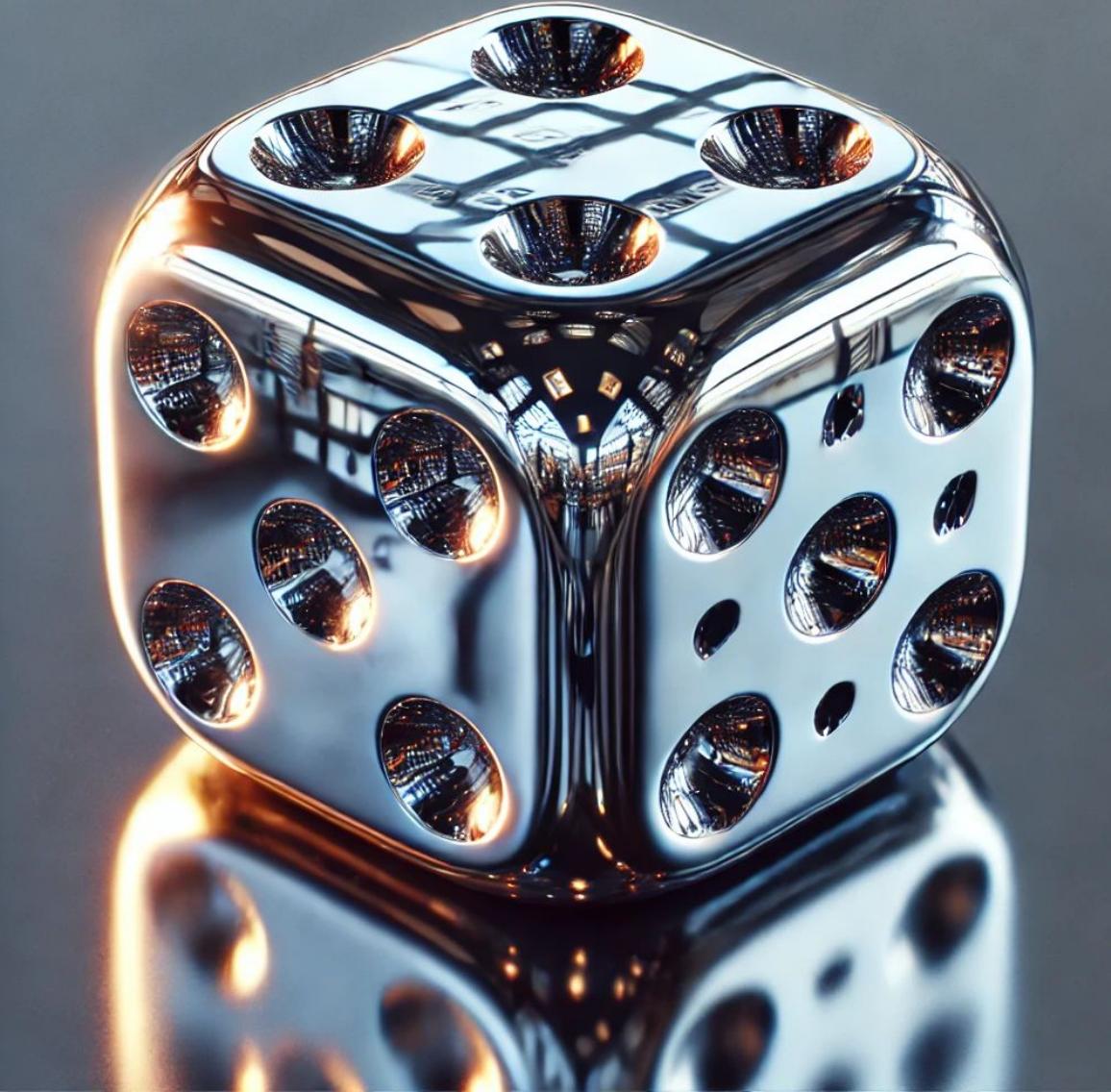
library(tidyverse)
library(infotheo)

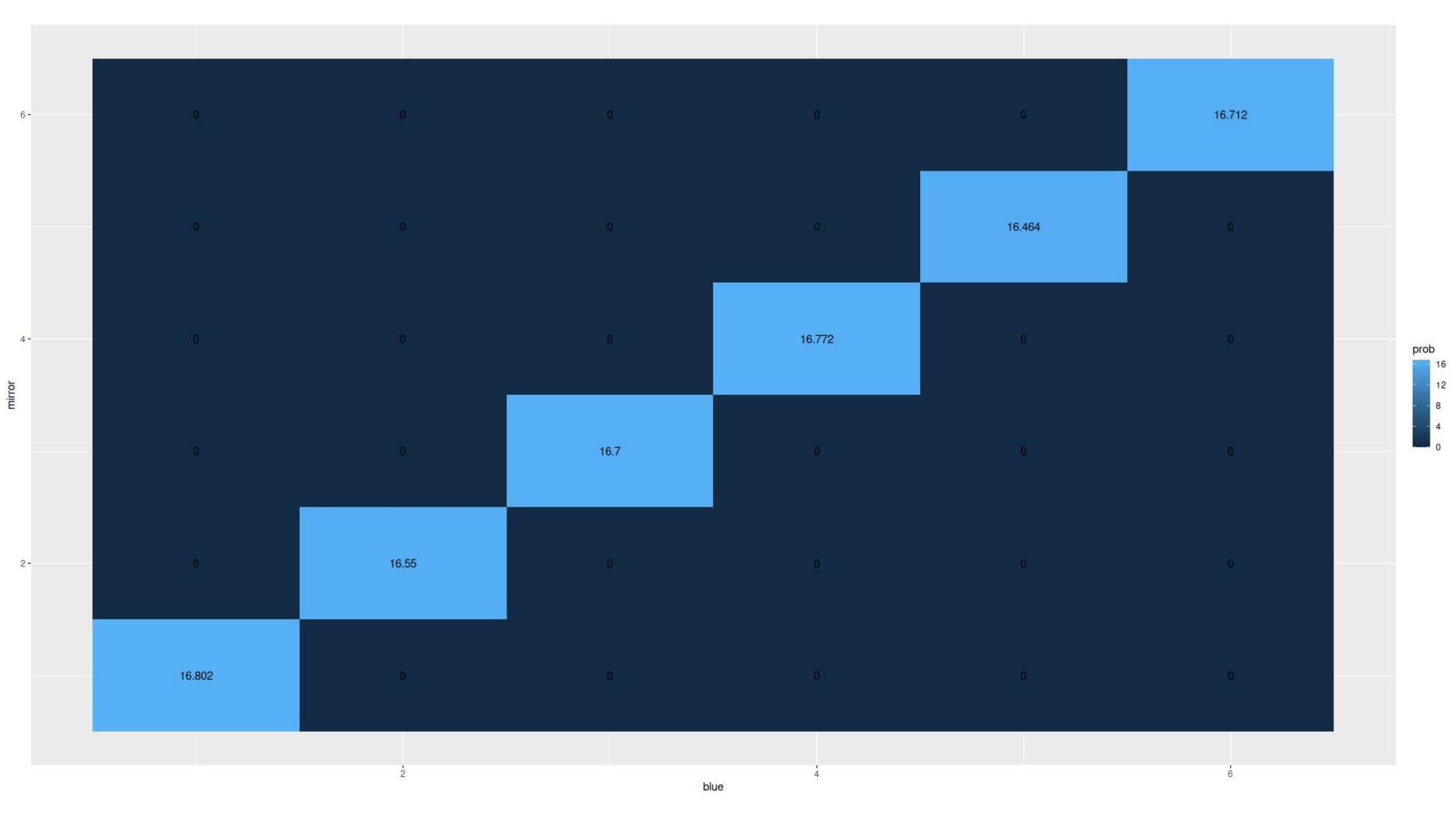
# make two fair dice (red and blue) ----
# Each die has outcomes from 1 to 6

set.seed(725)
{red_dice <- sample(1:6, size = 50000, replace = TRUE)
blue_dice <- sample(1:6, size = 50000, replace = TRUE)
}

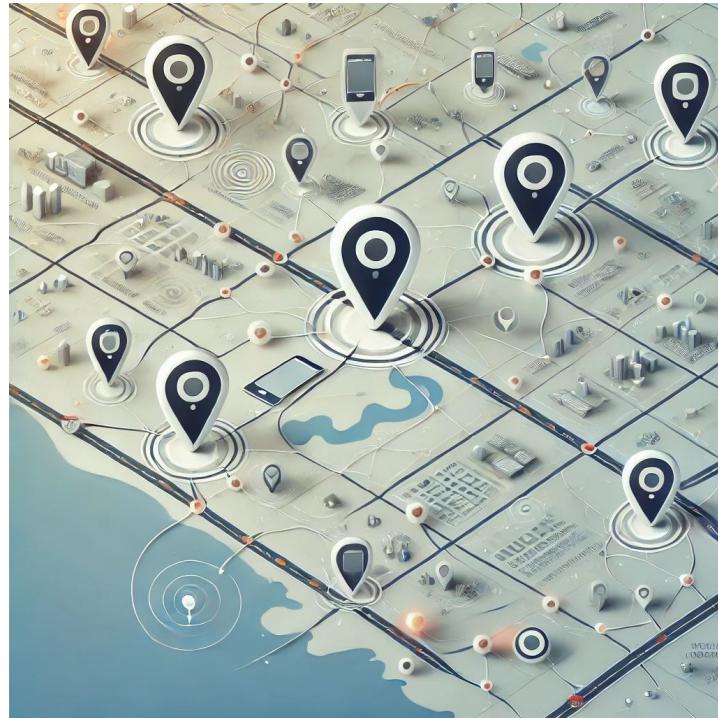
dice_data <- data.frame(red = red_dice, blue = blue_dice) |> as_tibble()
# plot probability density function (uniform) ----
```



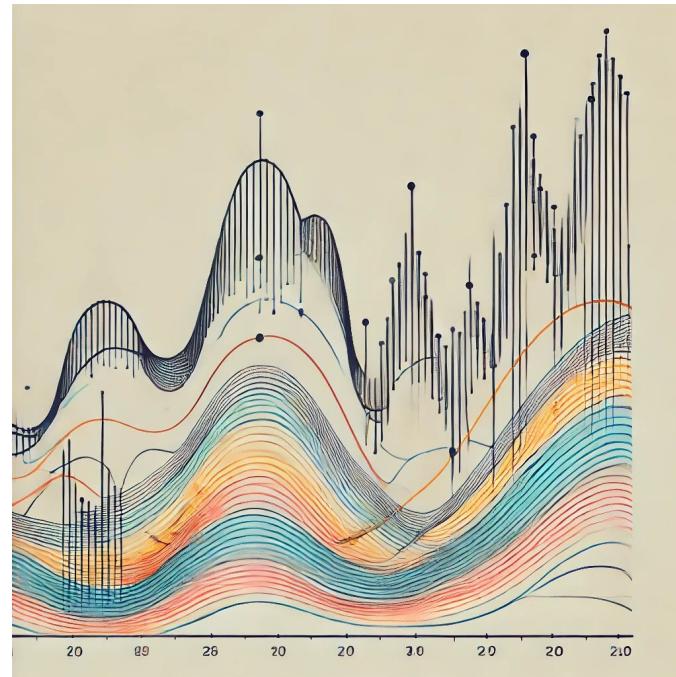




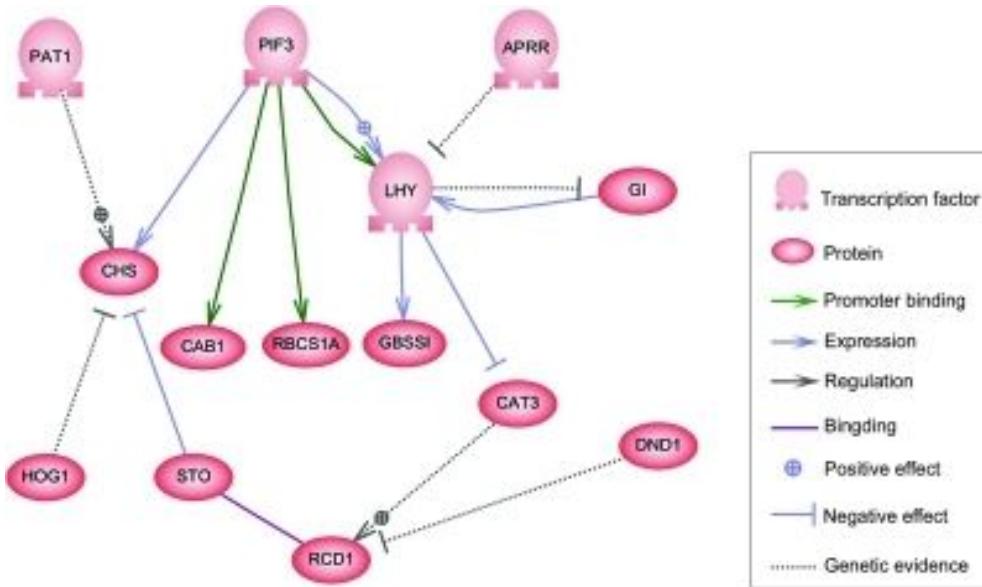
Some use cases



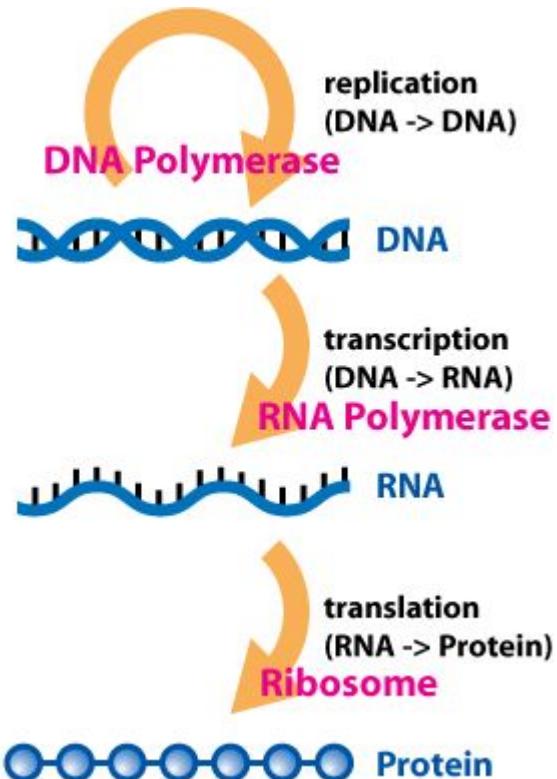
Some use cases

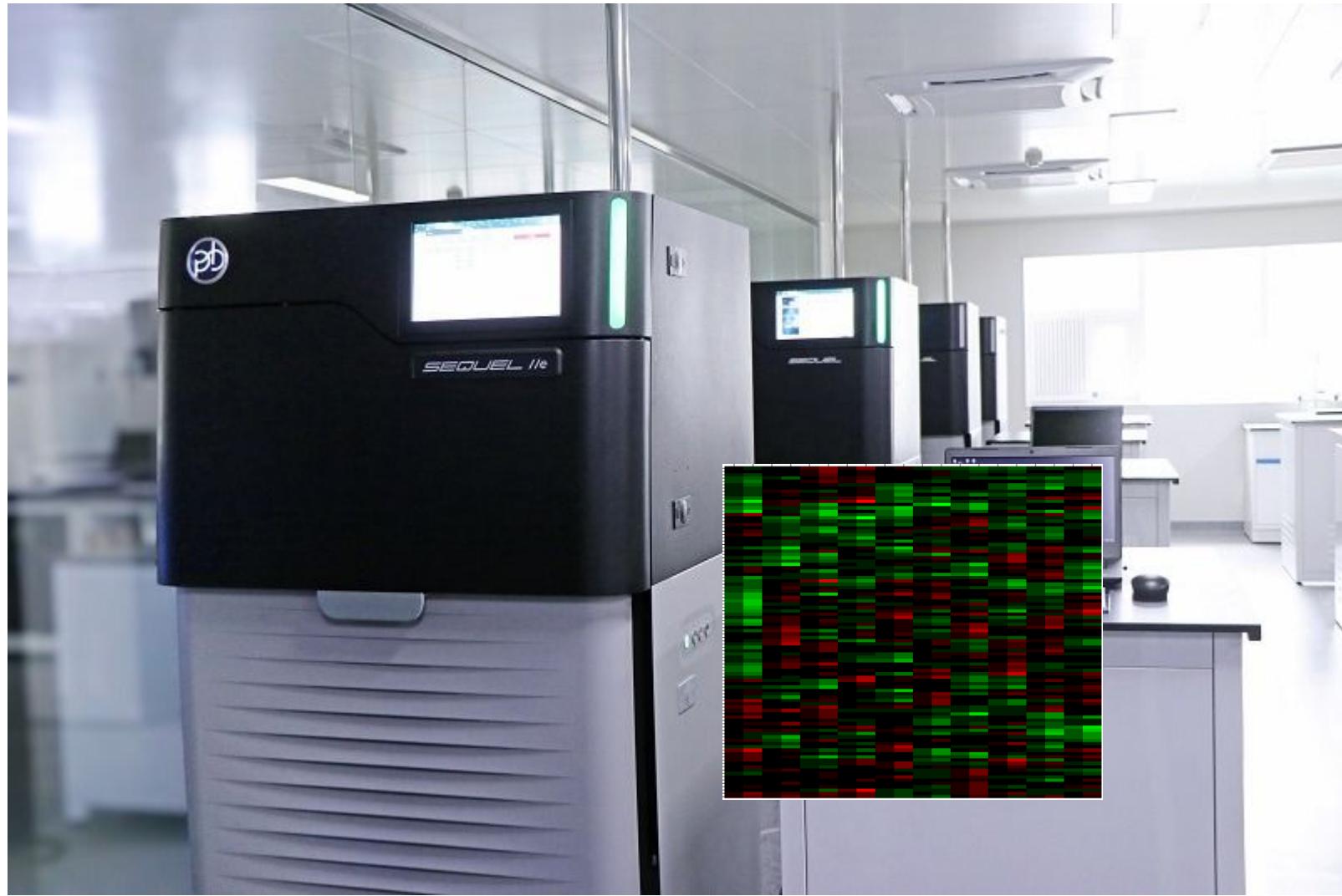


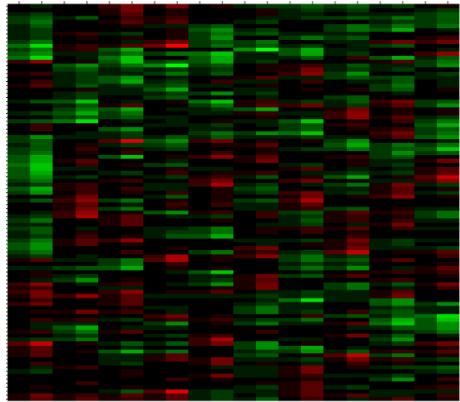
Some use cases



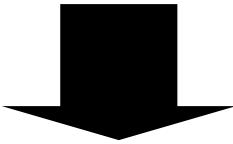
My (our) use case: genomics



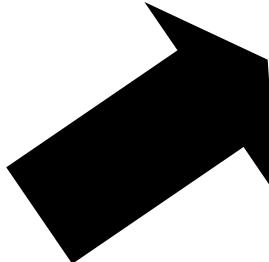




similarity



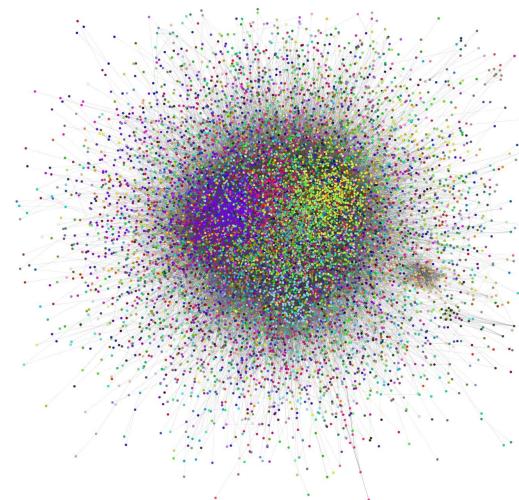
filtering

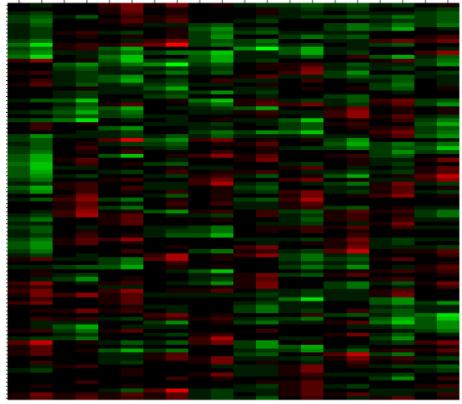


	g_1	g_2	...	g_n
g_1	1	0.7	...	0.9
g_2	0.7	1	...	0.2
...	1	...
g_n	0.9	0.2	...	1

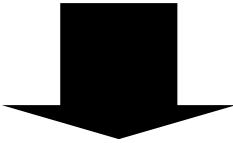
	g_1	g_2	...	g_n
g_1	0	1	...	1
g_2	1	0	...	0
...	0	...
g_n	1	0	...	0

Network

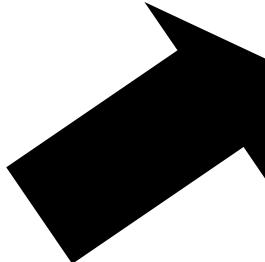




similarity



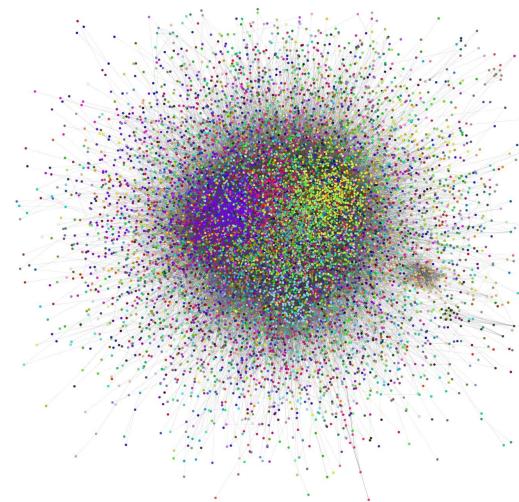
filtering

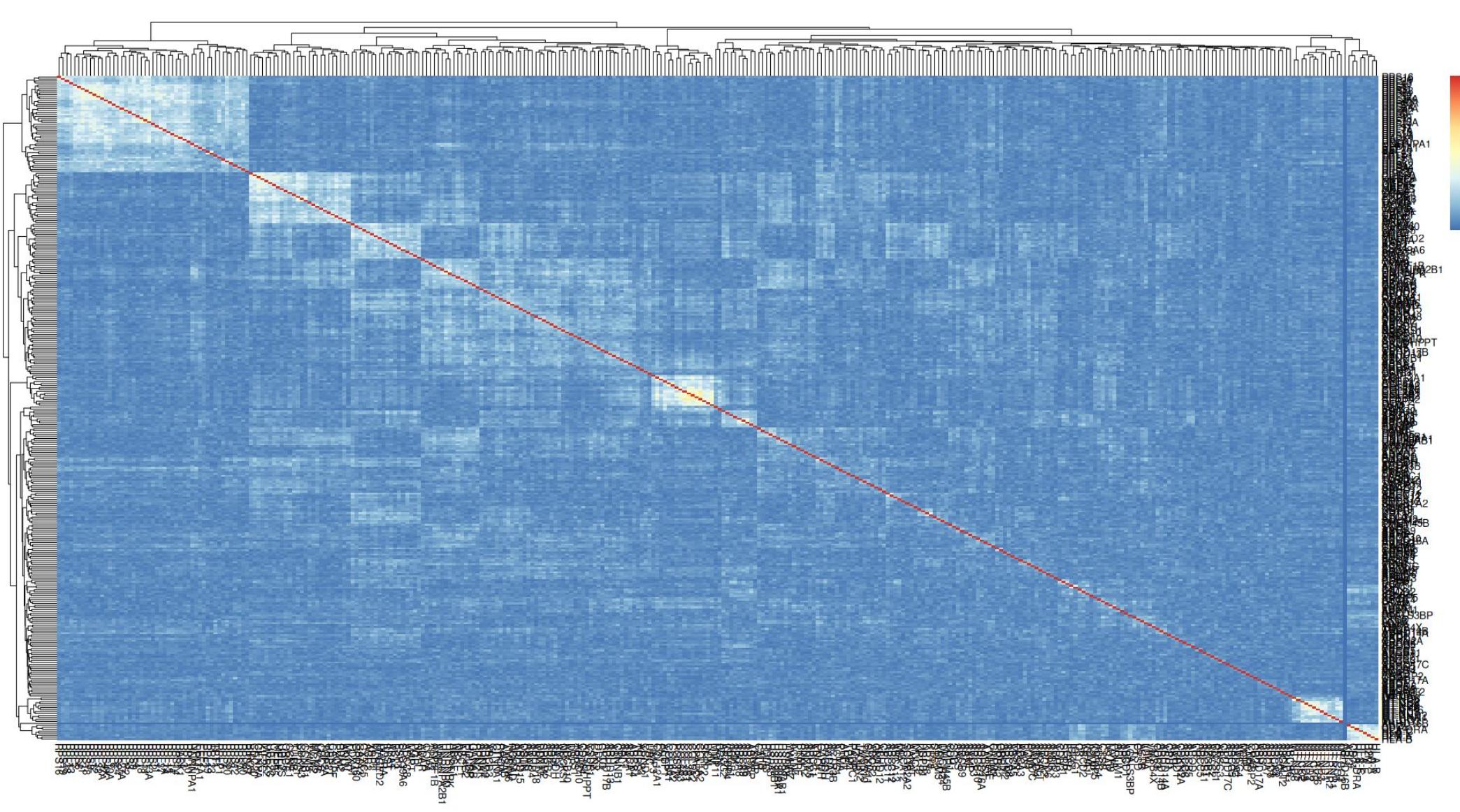


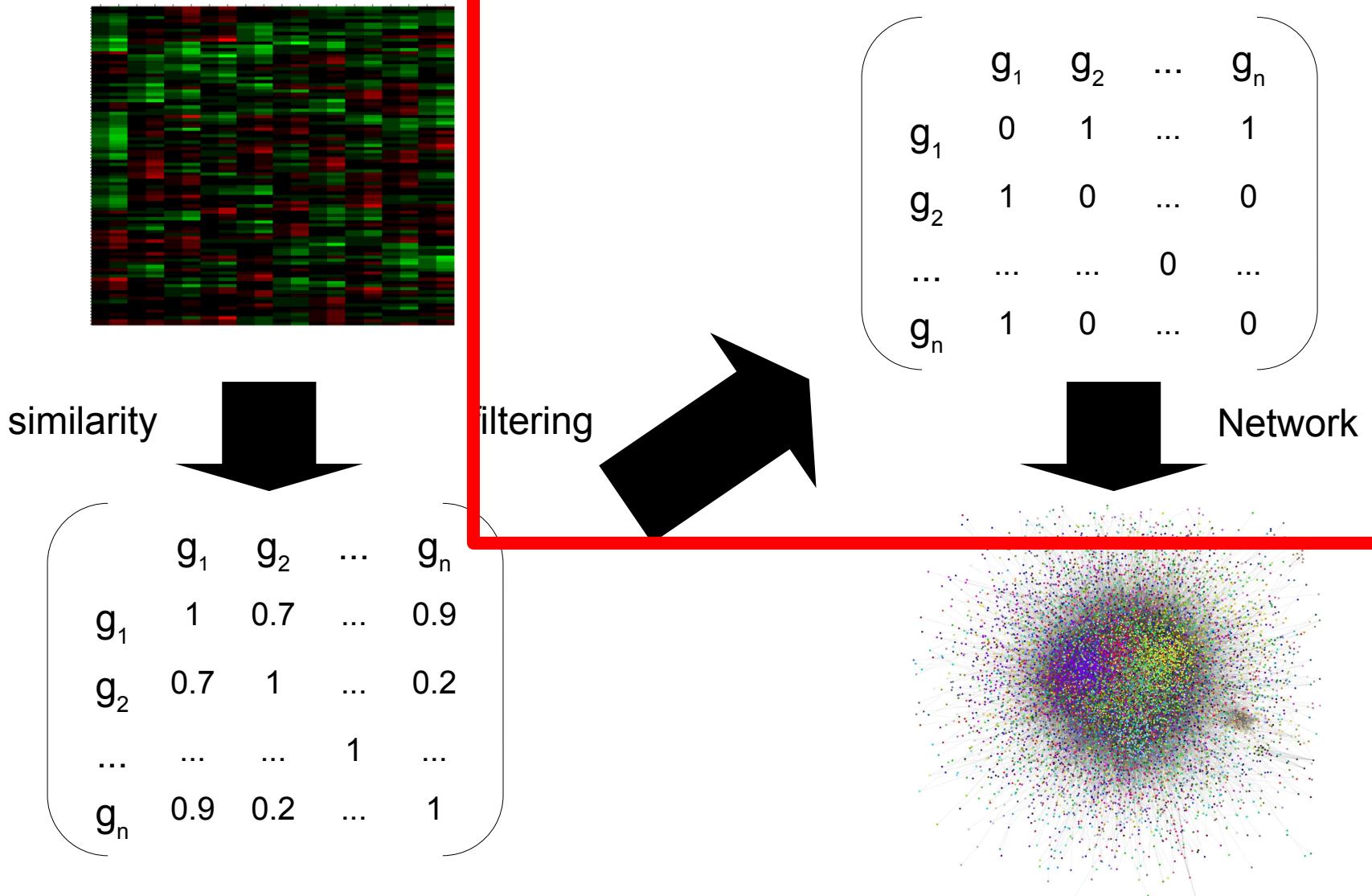
	g_1	g_2	...	g_n
g_1	1	0.7	...	0.9
g_2	0.7	1	...	0.2
...	1	...
g_n	0.9	0.2	...	1

	g_1	g_2	...	g_n
g_1	0	1	...	1
g_2	1	0	...	0
...	0	...
g_n	1	0	...	0

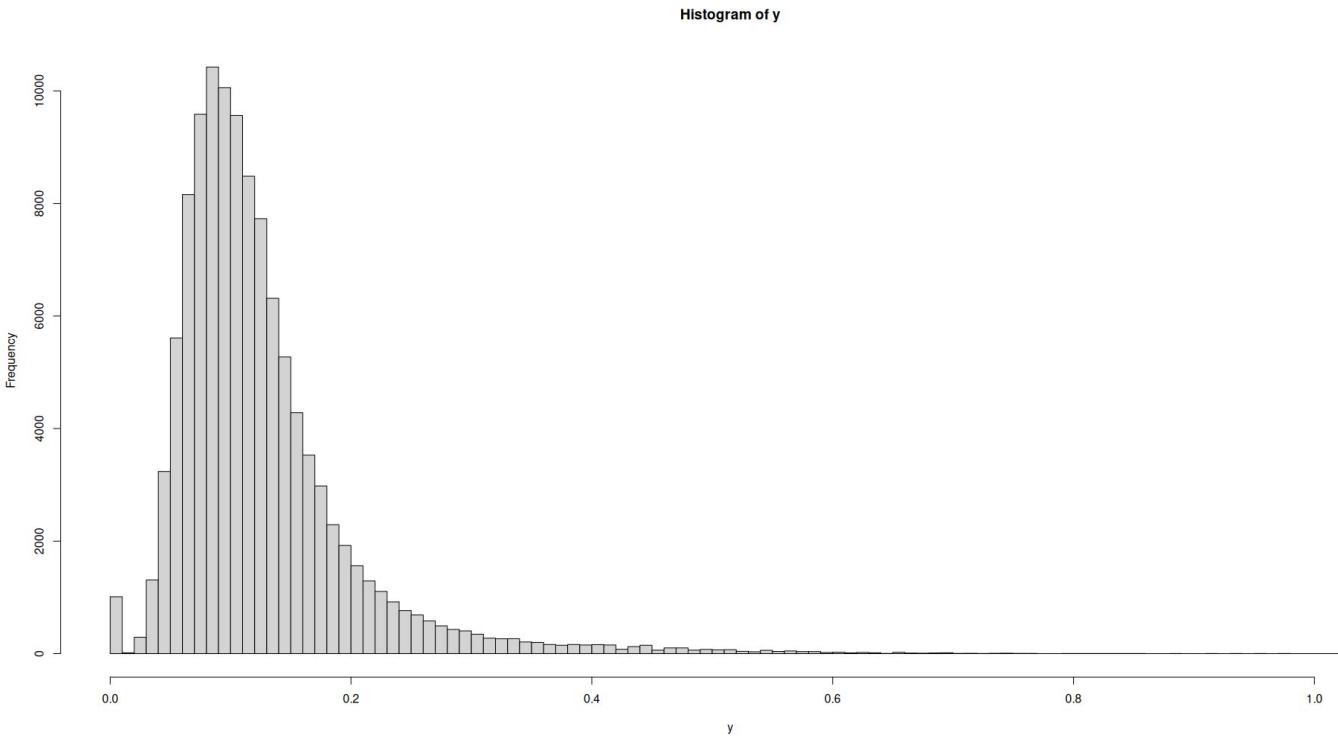
Network



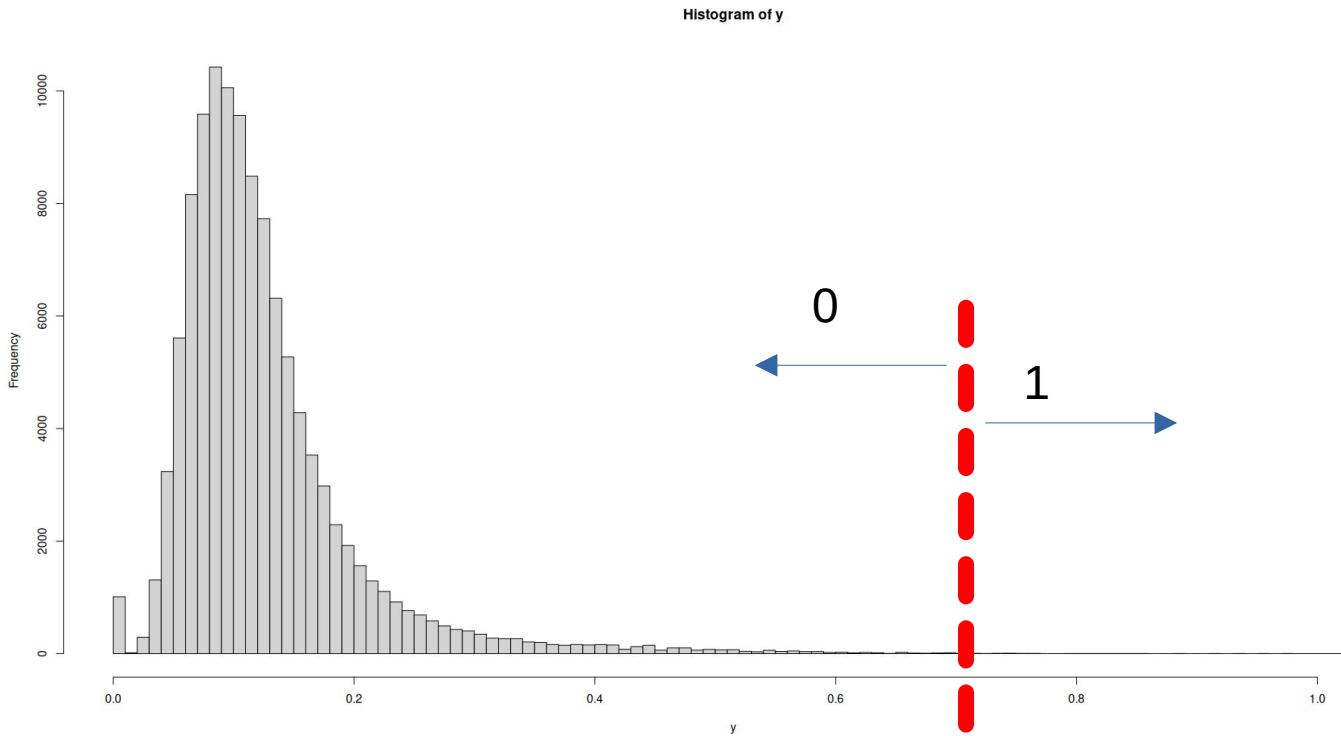




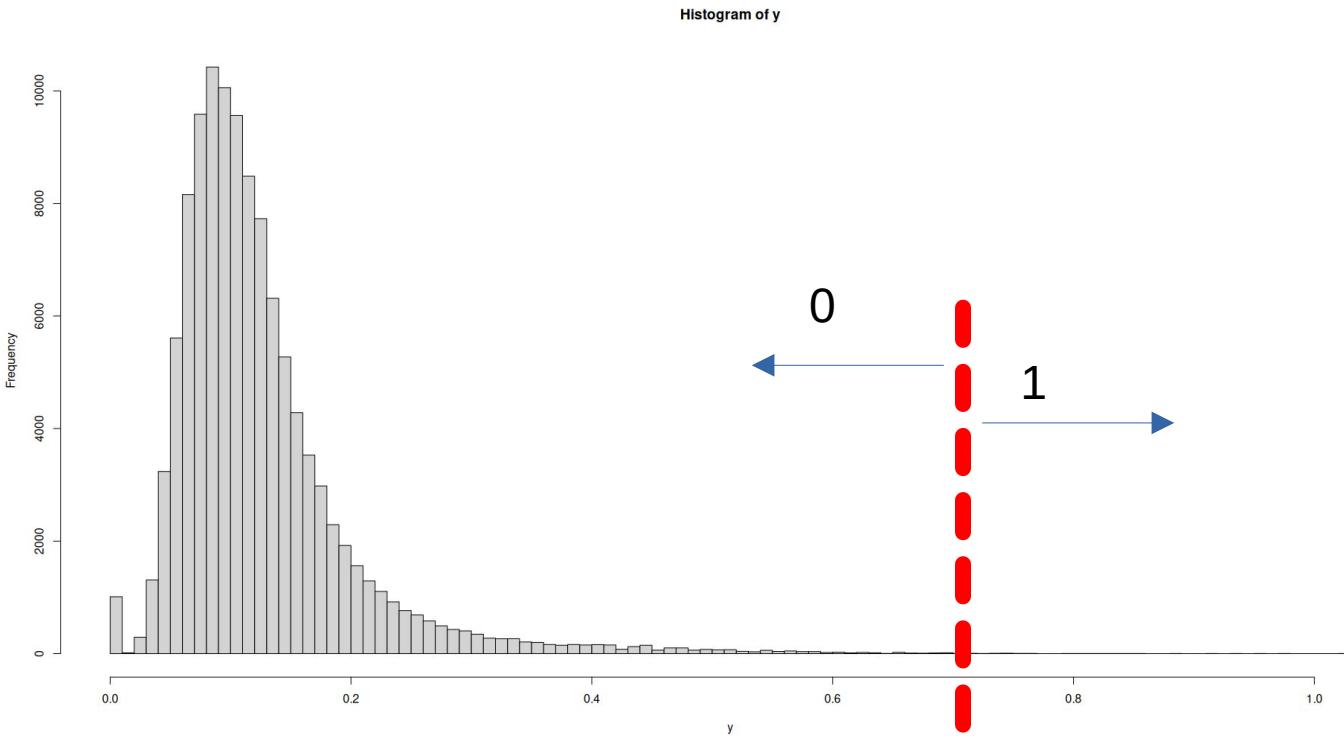
Basic: thresholding (aka: “relevance networks”)



Basic: thresholding (aka: “relevance networks”)



Basic: thresholding (aka: “relevance networks”)



Rationale:

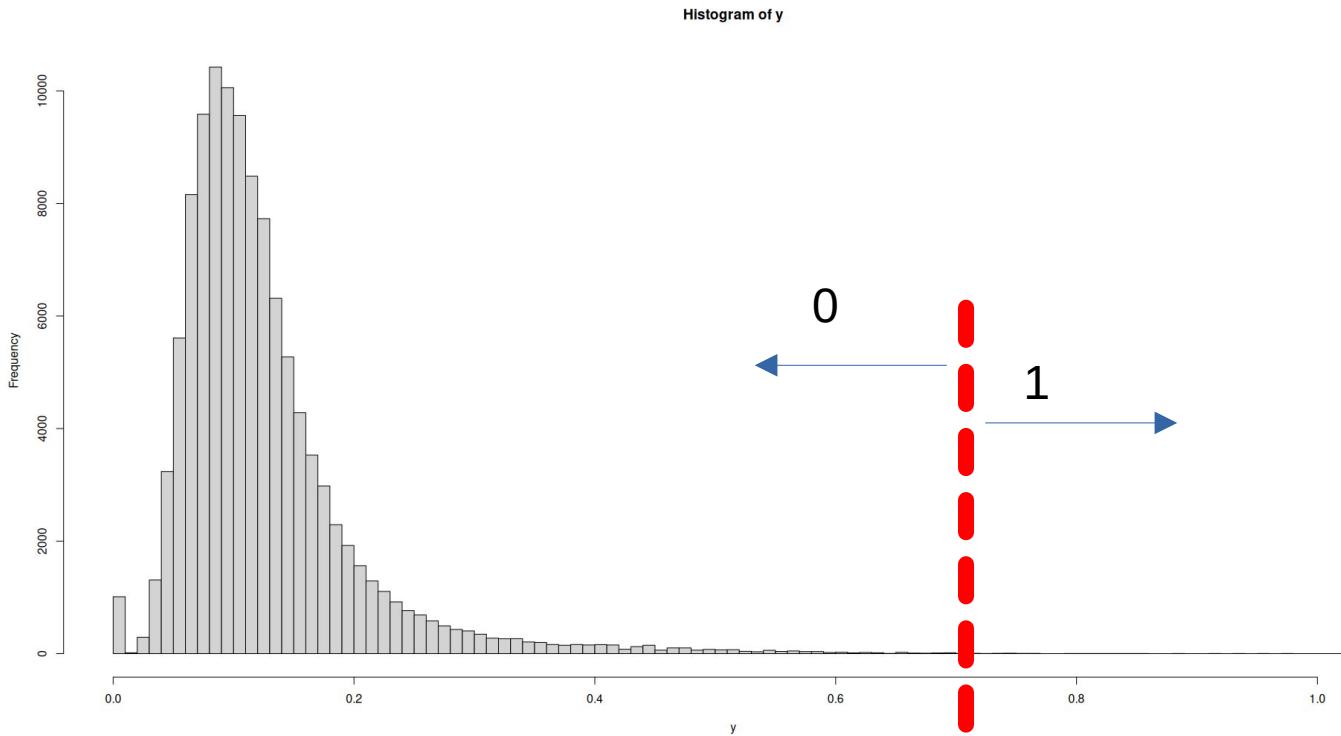
MI \sim permutation p.value

$$p(I \geq I_0 | \bar{I} = 0) \propto e^{-\alpha MI_0}$$

See:

[https://doi.org/
10.1186/1471-2105-7-s1-
s7](https://doi.org/10.1186/1471-2105-7-s1-s7)

Basic: thresholding (aka: “relevance networks”)



Cool... so where do I set the threshold?

... need phenomenological input

**Open Science
Free Software
Collaborative Culture**

gdeanda@inmegen.edu.mx
@gdeandajauregui
github.com/
guillermodeandajauregui
guillermodeandajauregui.github.io