Networks integration

Student: Peter Leontev,
Scientific advisor: Son Pham, UCSD

Project goals

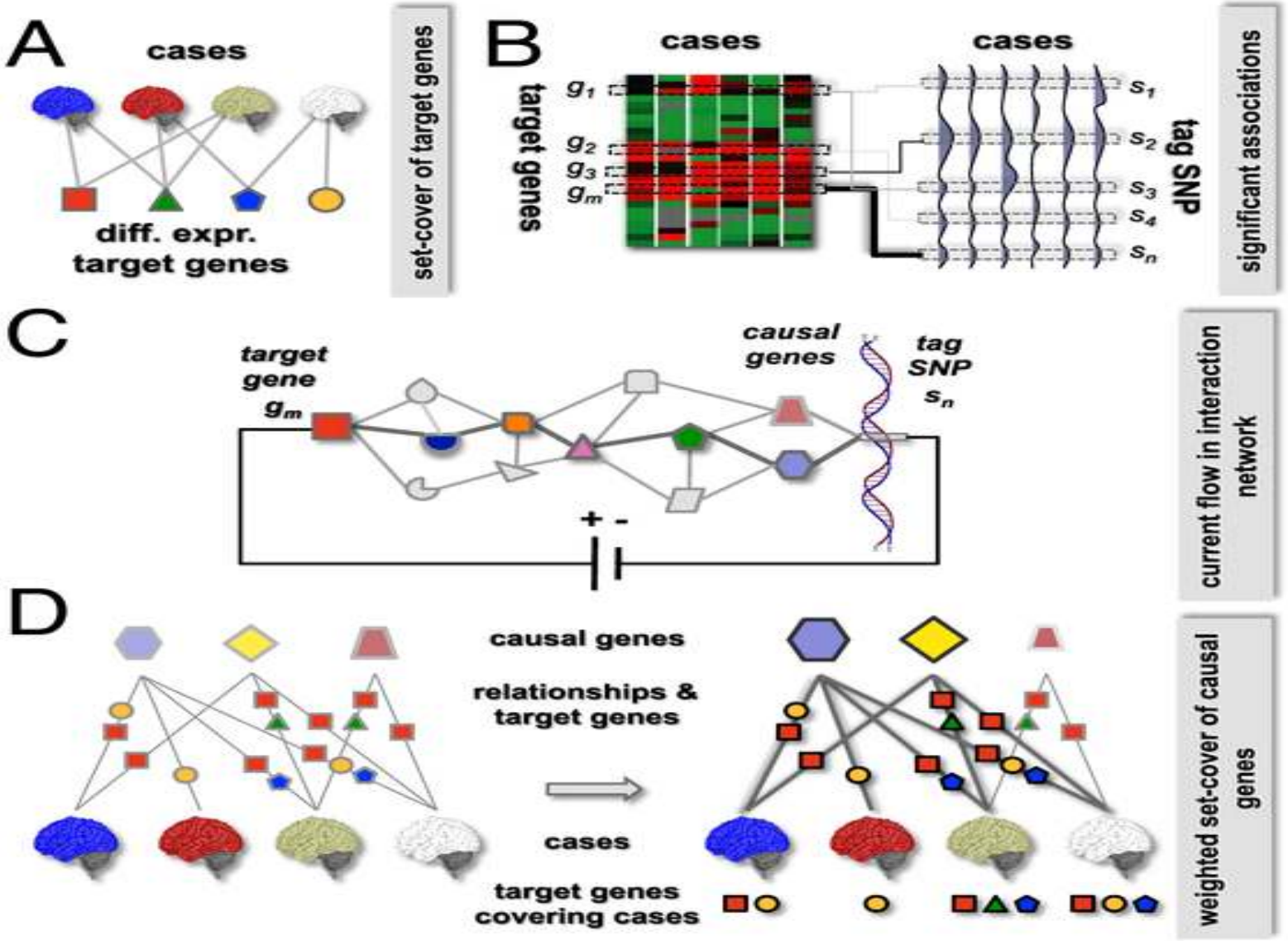
- Review an algorithm from the article: “Identifying Causal Genes and Dysregulated Pathways in Complex Diseases” (Kim et. al., 2011);
- Implement the algorithm;
- Detect and fix it bottlenecks;
- Find a way to integrate genomic data into output to reveal mechanisms of complex diseases;

ICGDP algorithm

Idea:

- take profiles of genes expression from the disease and control samples;
- take set of loci with CNV alterations from different chromosomes;
- take a network of molecular interactions containing as much as possible genes.

Result: get causal genes and dysregulated pathways.

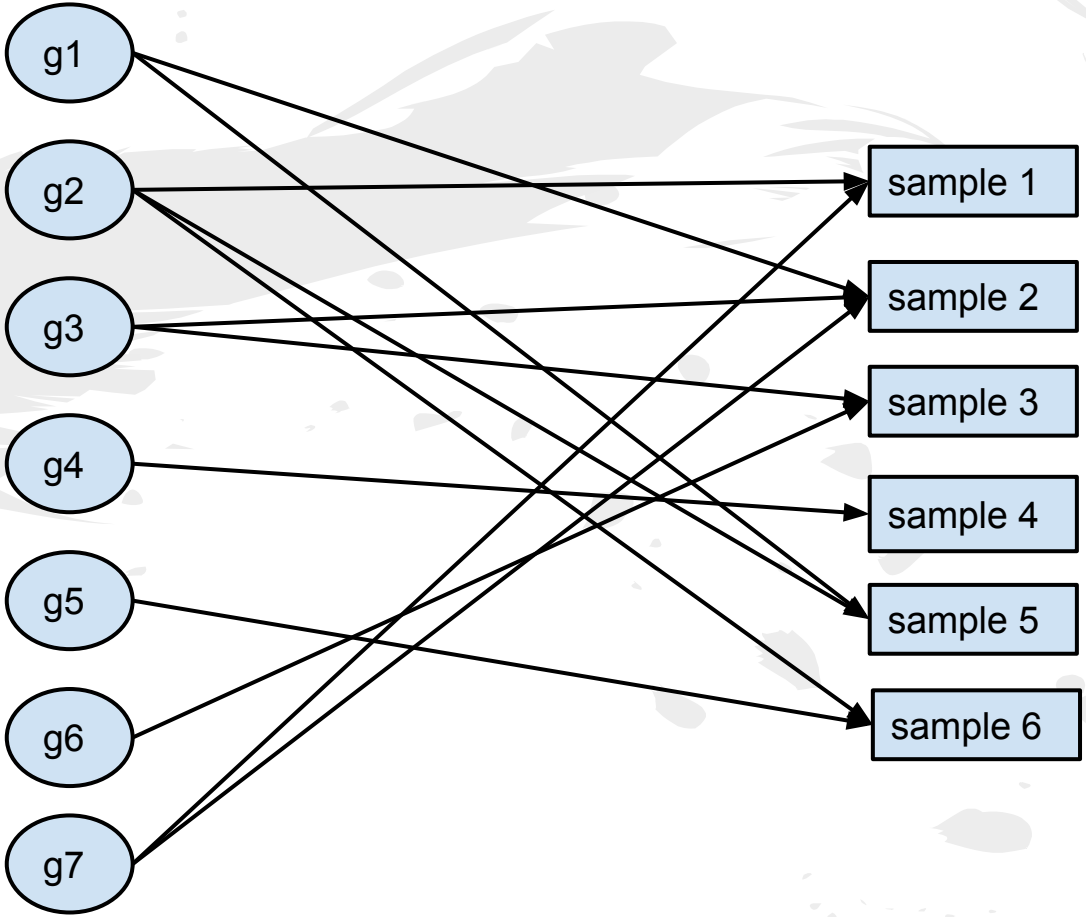

A
cases
diff. expr.
target genes
set-cover of target genes
B
cases
cases
target genes
g_1
g_2
g_3
g_m
tag SNP
s_1
s_2
s_3
s_4
s_n
significant associations
C
target
gene
g_m
causal
genes
tag
SNP
s_n
+ -
current flow in interaction network
D
causal genes
relationships &
target genes
cases
target genes
covering cases
weighted set-cover of causal genes

Overview

Selecting target genes

We stop if coverage is more or equal to A. That must be true for all samples except B.

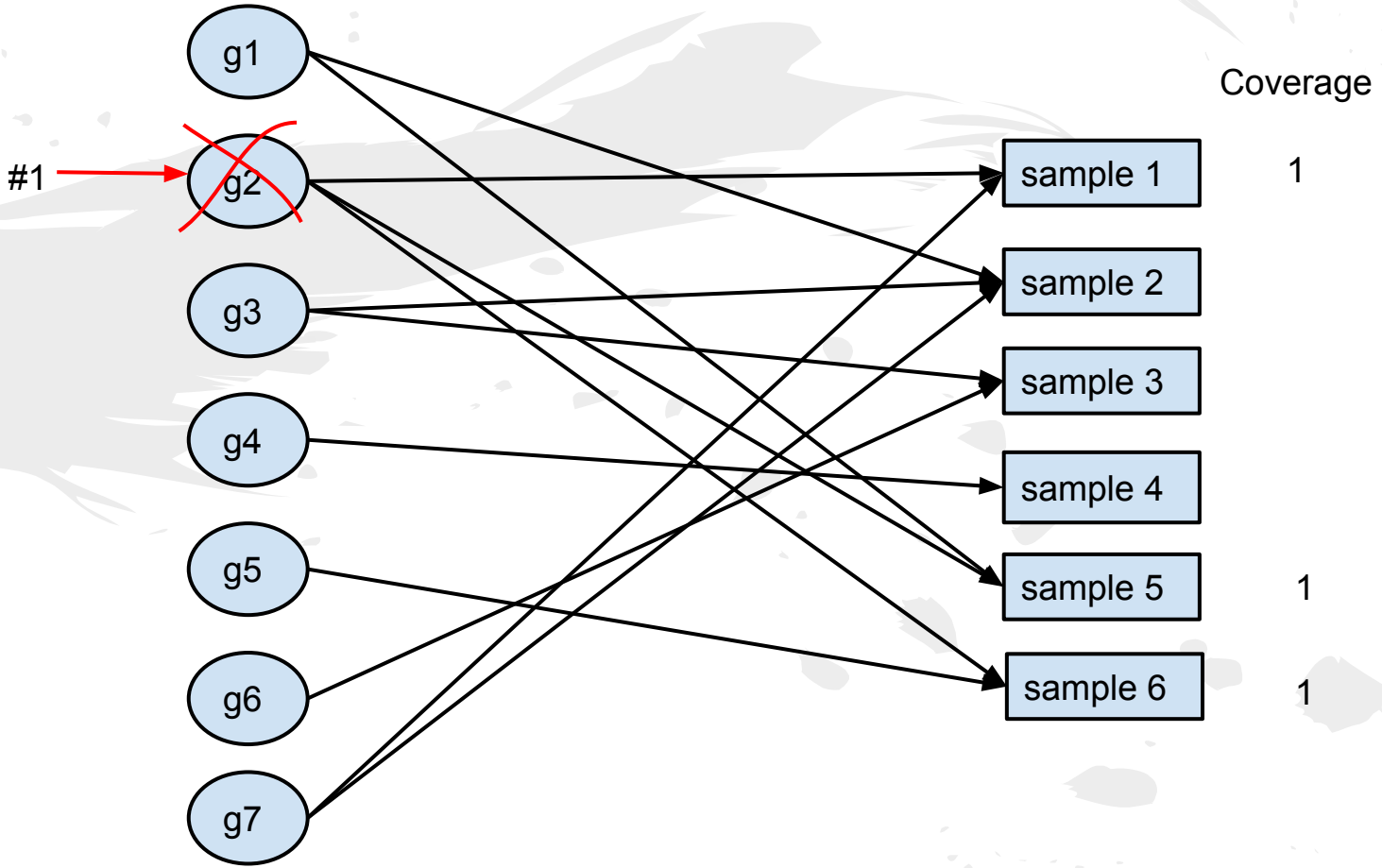
Let's consider:
A = 2, B = 2



Selecting target genes

We stop if coverage is more or equal to A. That must be true for all samples except B.

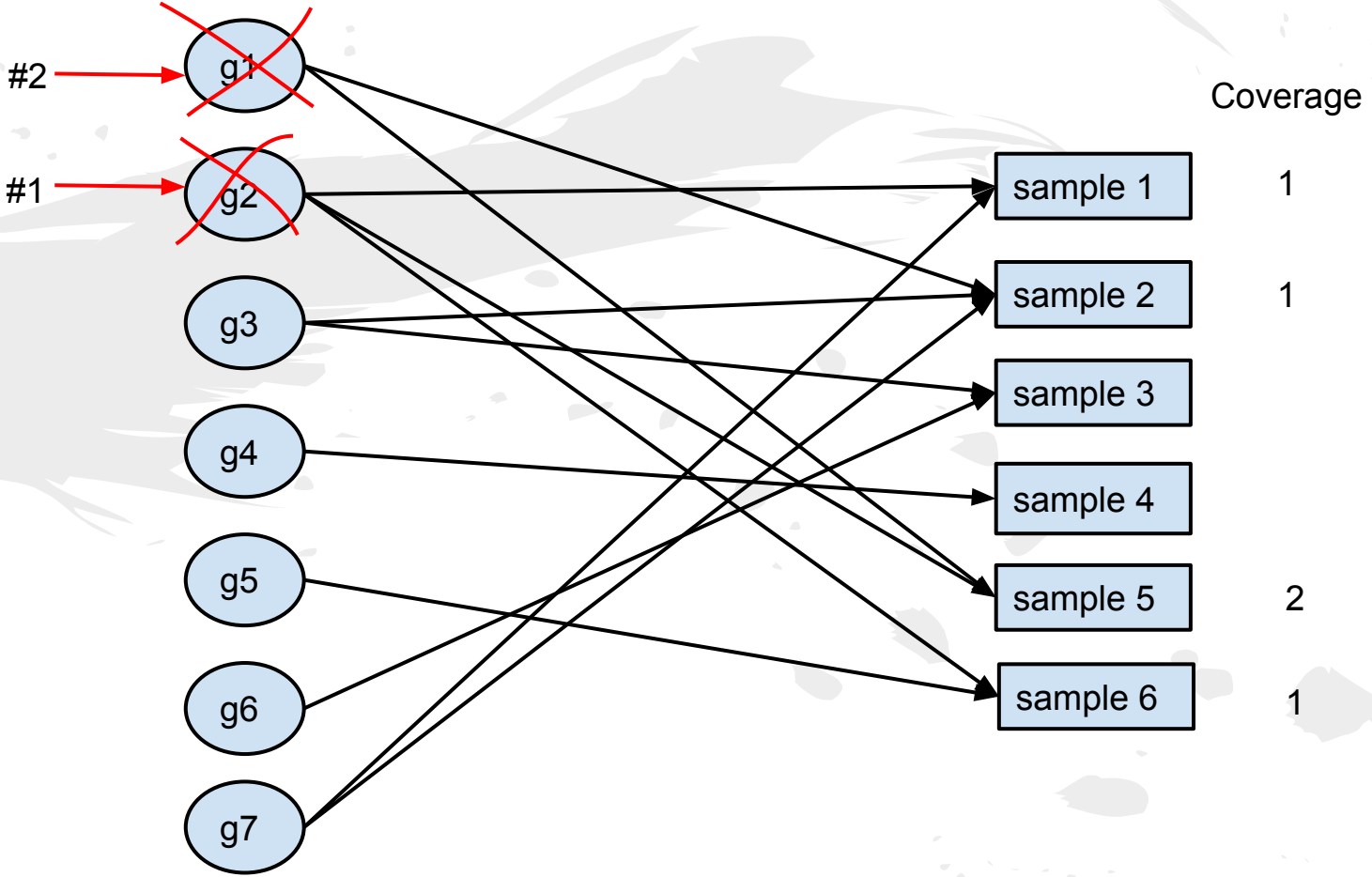
Let's consider:
A = 2, B = 2



Selecting target genes

We stop if coverage is more or equal to A. That must be true for all samples except B.

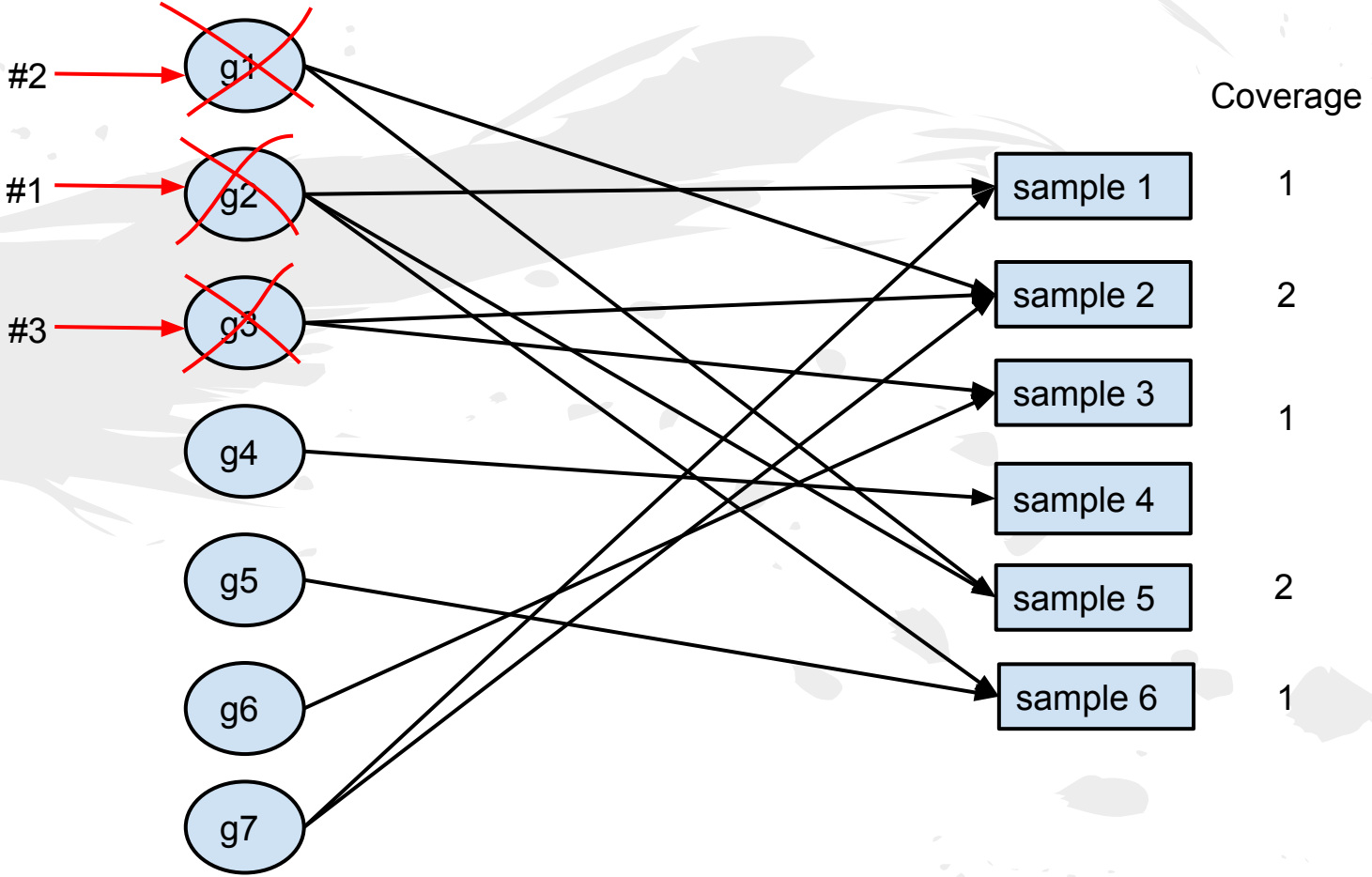
Let's consider:
A = 2, B = 2



Selecting target genes

We stop if coverage is more or equal to A. That must be true for all samples except B.

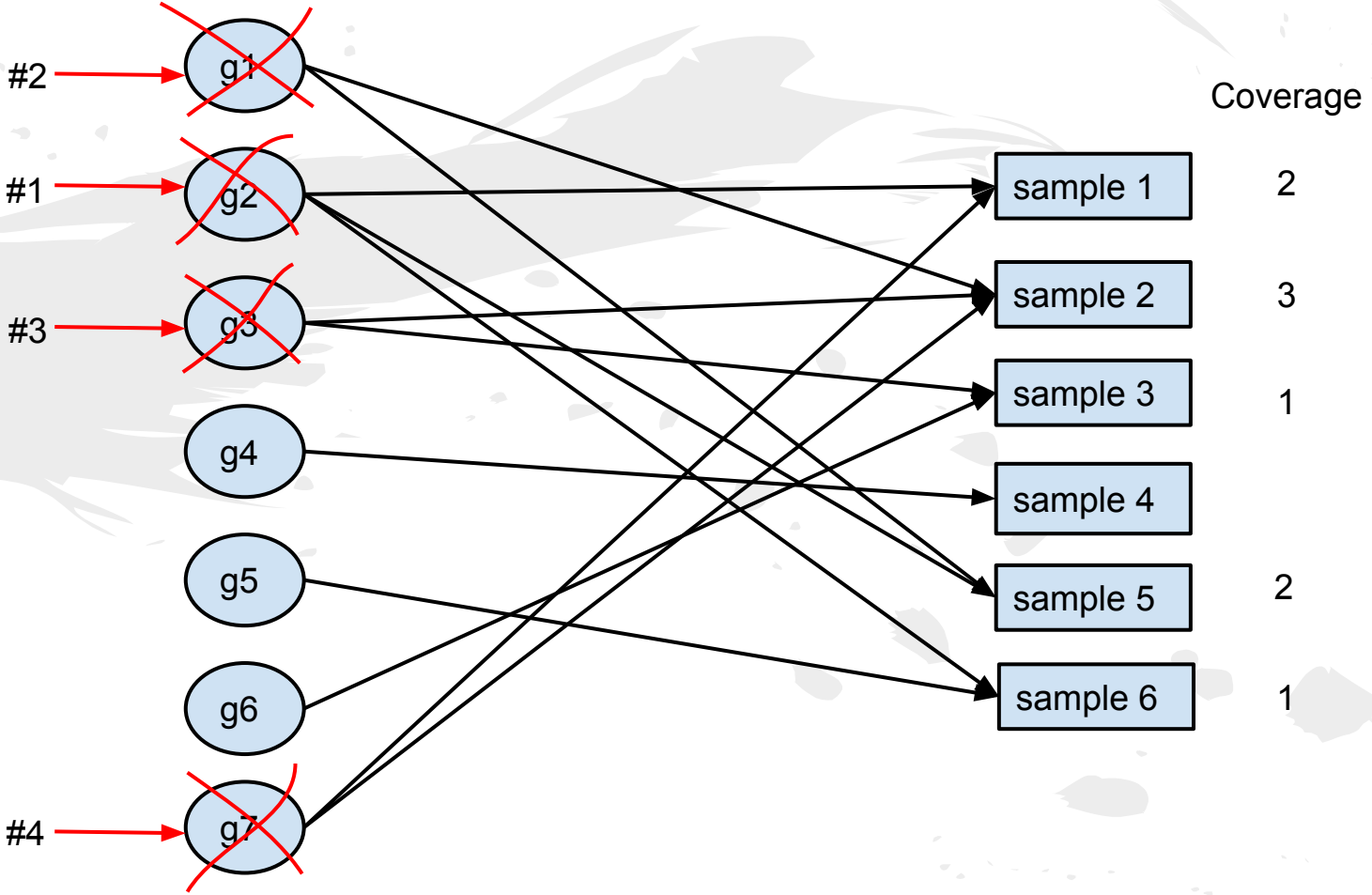
Let's consider:
A = 2, B = 2



Selecting target genes

We stop if coverage is more or equal to A. That must be true for all samples except B.

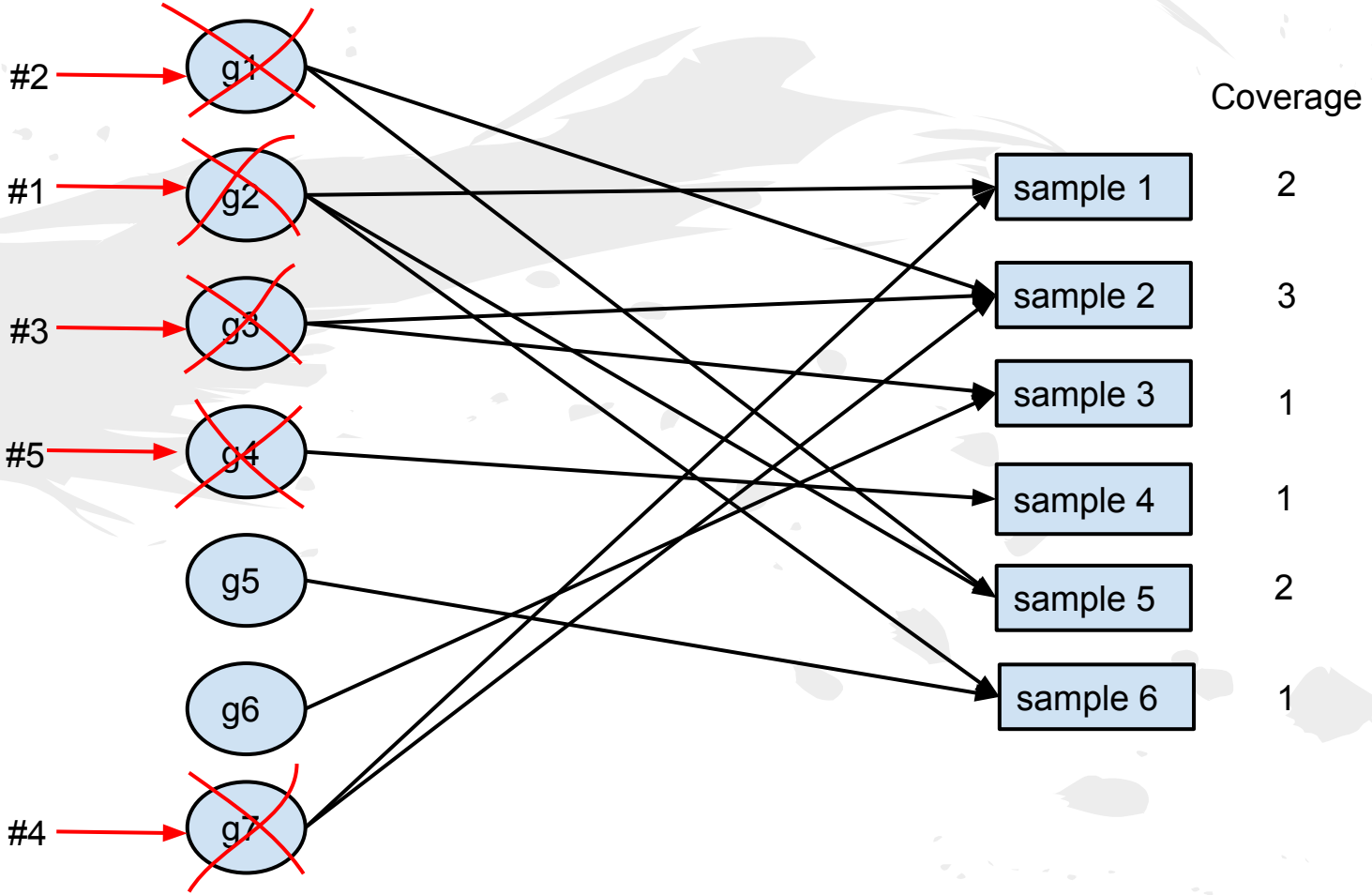
Let's consider:
A = 2, B = 2



Selecting target genes

We stop if coverage is more or equal to A. That must be true for all samples except B.

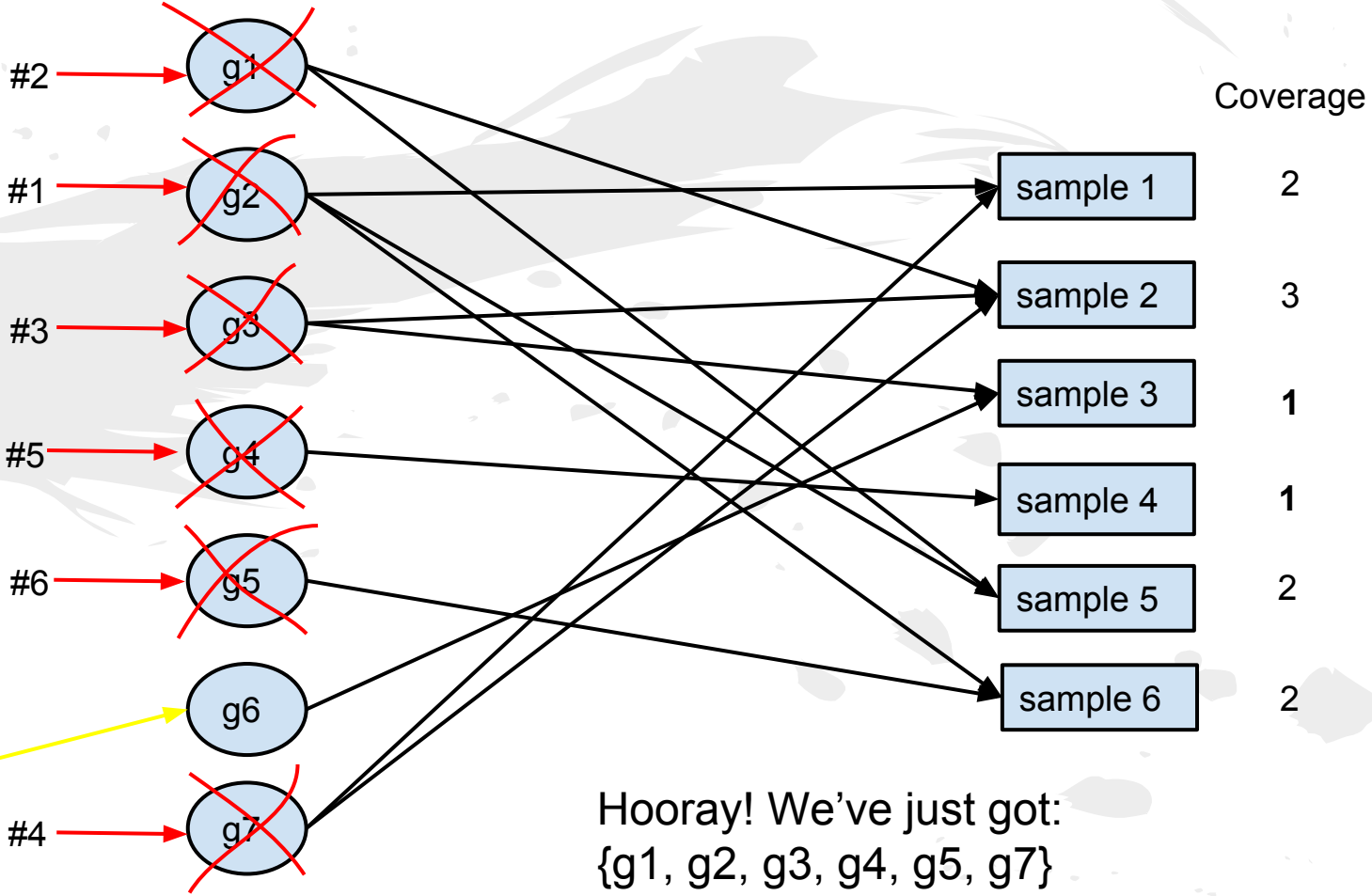
Let's consider:
A = 2, B = 2



Selecting target genes

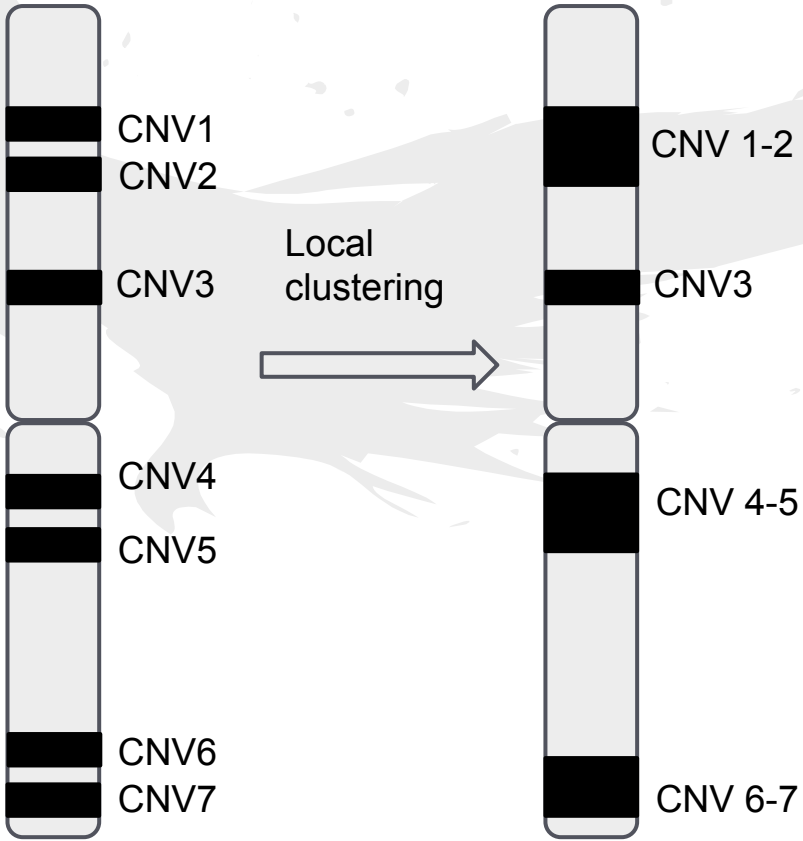
We stop if coverage is more or equal to A. That must be true for all samples except B.

Let's consider:
A = 2, B = 2



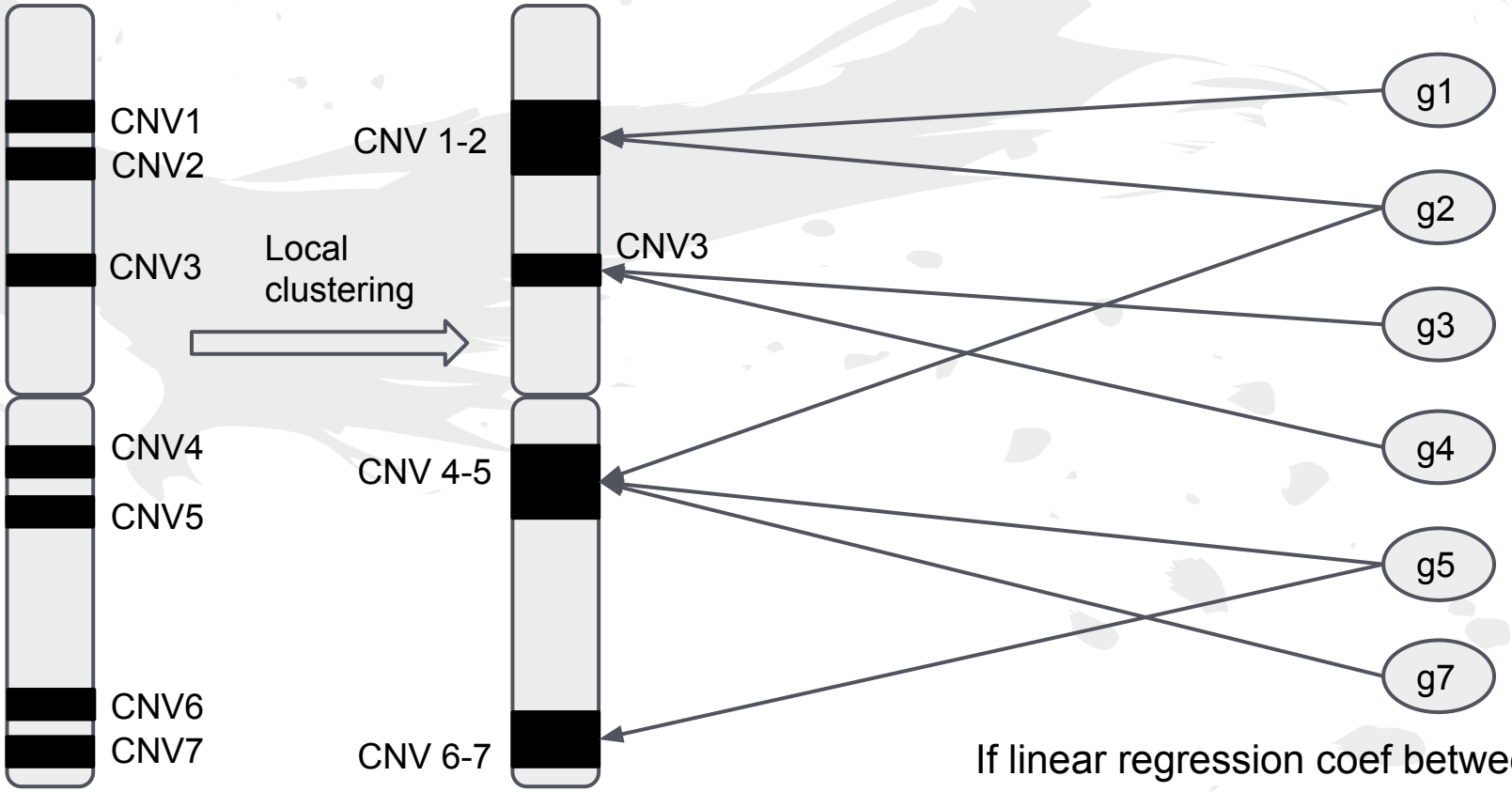
Hooray! We've just got:
{g1, g2, g3, g4, g5, g7}

Getting associated loci



CNV = copy number variations for all disease samples

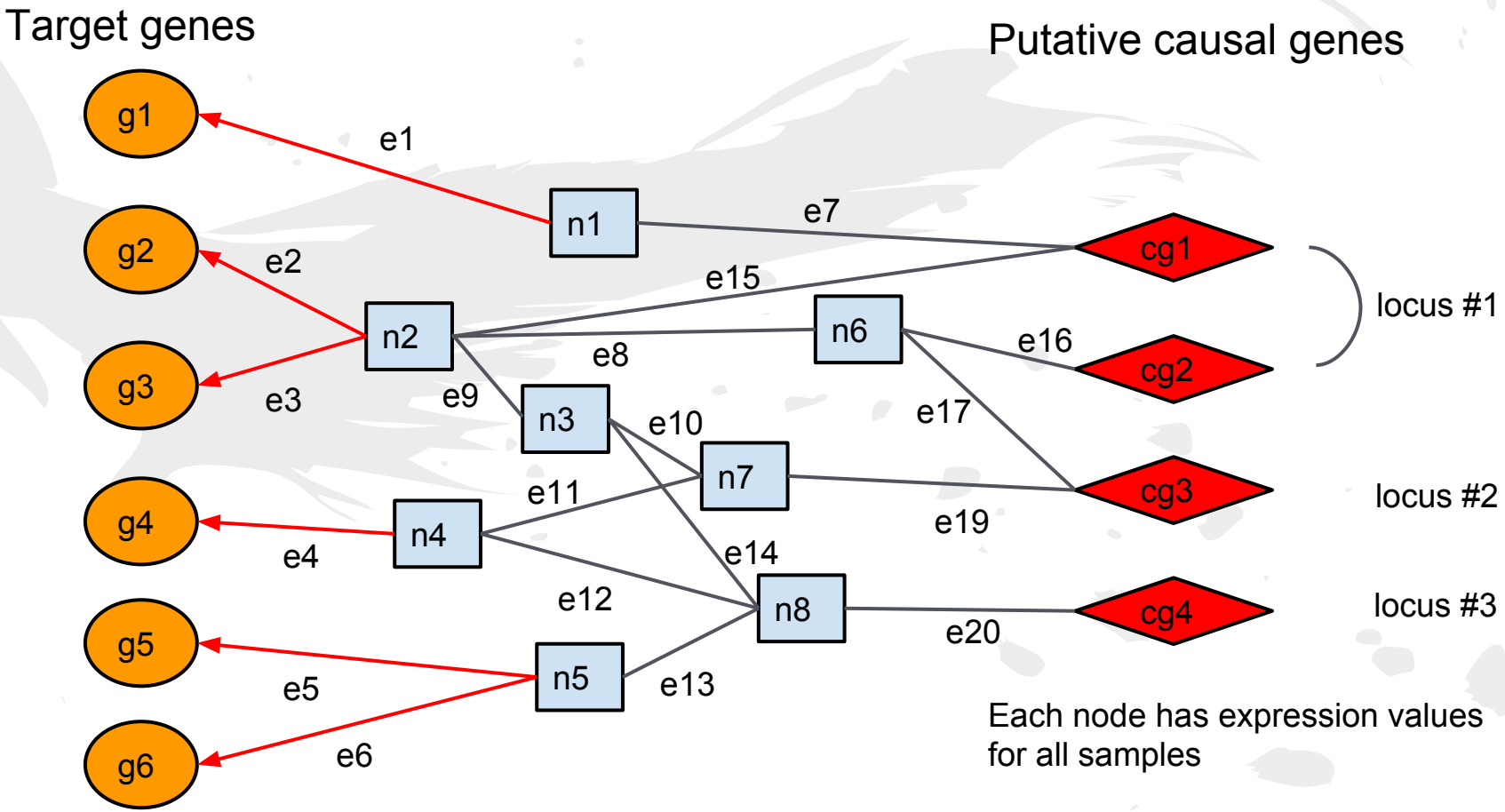
Getting associated loci



CNV = copy number variations for all disease samples

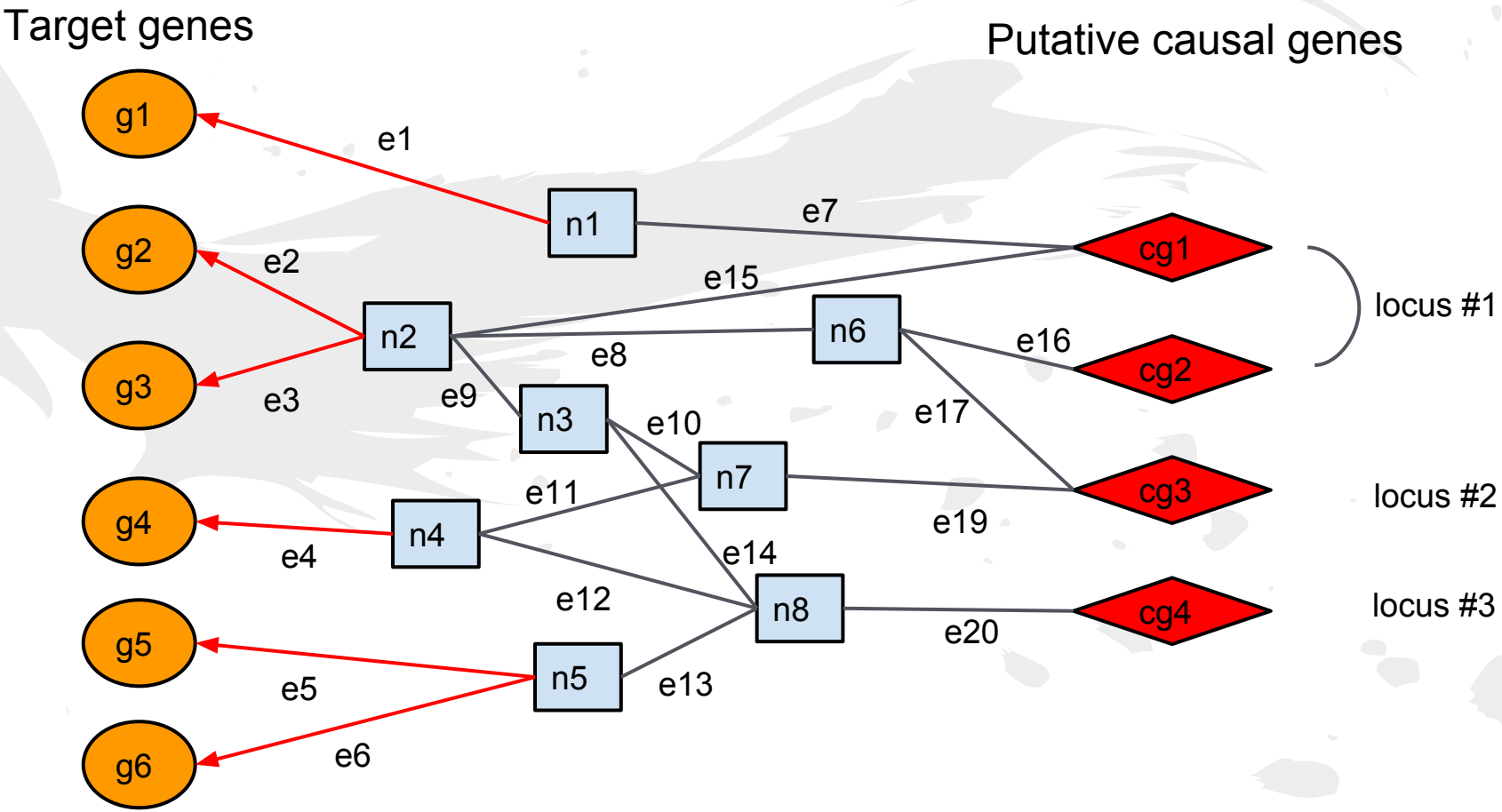
If linear regression coef between gene expression and CNV values > THRESHOLD then we take such locus

Selecting causal genes



Network of interactions consists of ppi, pdi and pho events

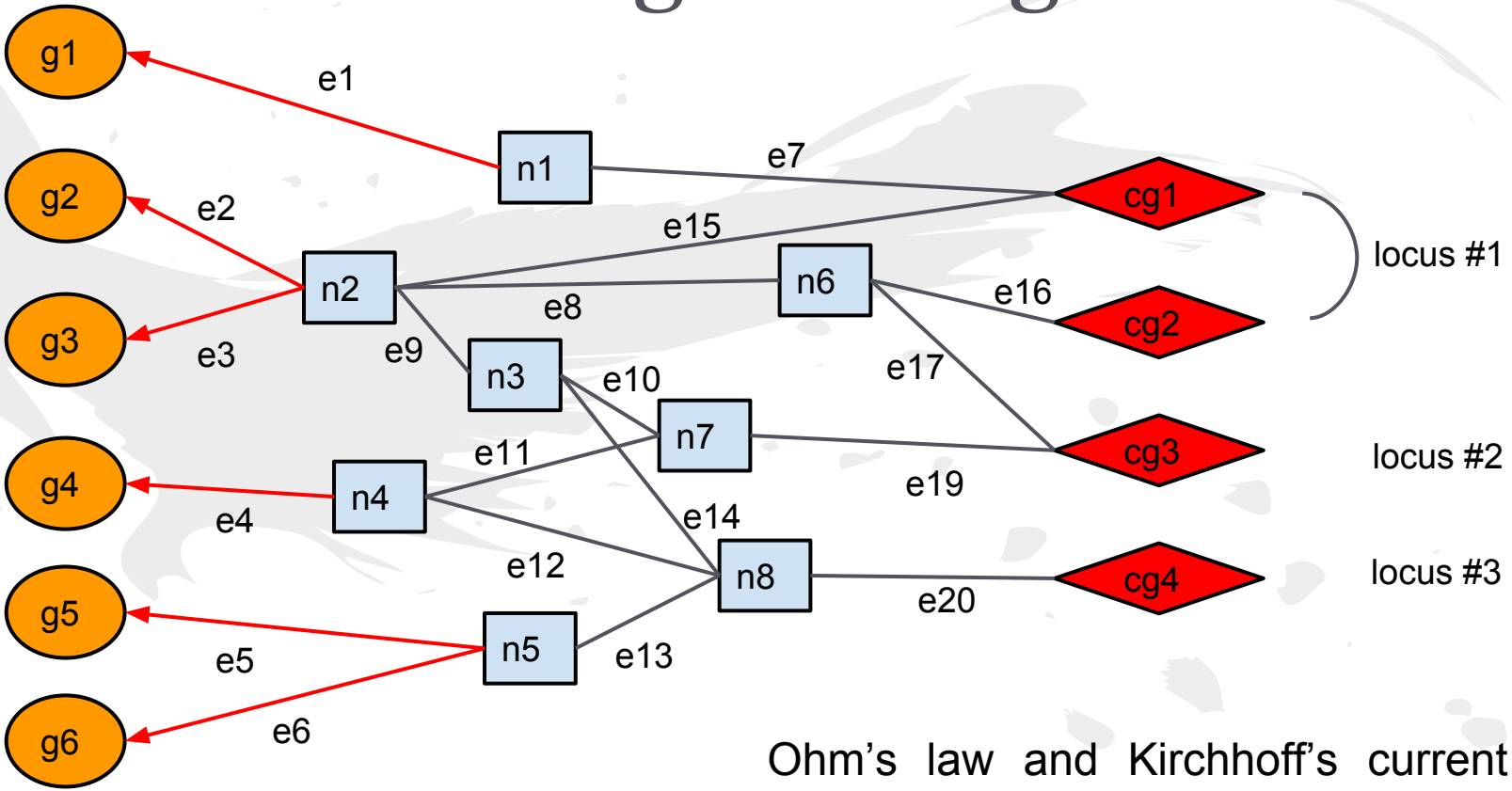
Selecting causal genes



Selecting causal genes

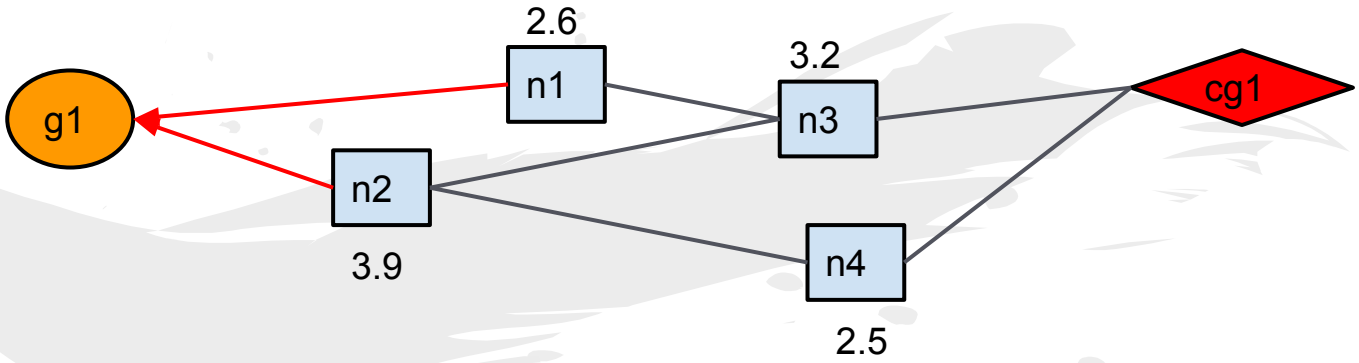
- Modelling flow of information requires set of rules for both nodes and edges;
- Need to set values for each node using correlation between this node and target gene node and each edge as mean of its nodes;
- Flow passes easily those nodes that are most correlated with target node;
- Physics saves the day;

Selecting causal genes

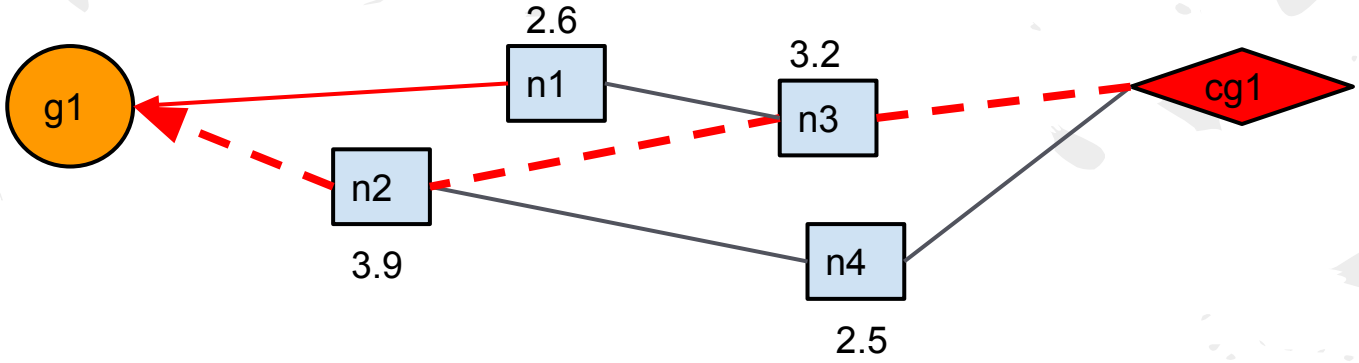


Ohm's law and Kirchhoff's current law allow to compute current to each causal gene (and also for each node).

Identifying dysregulated pathways



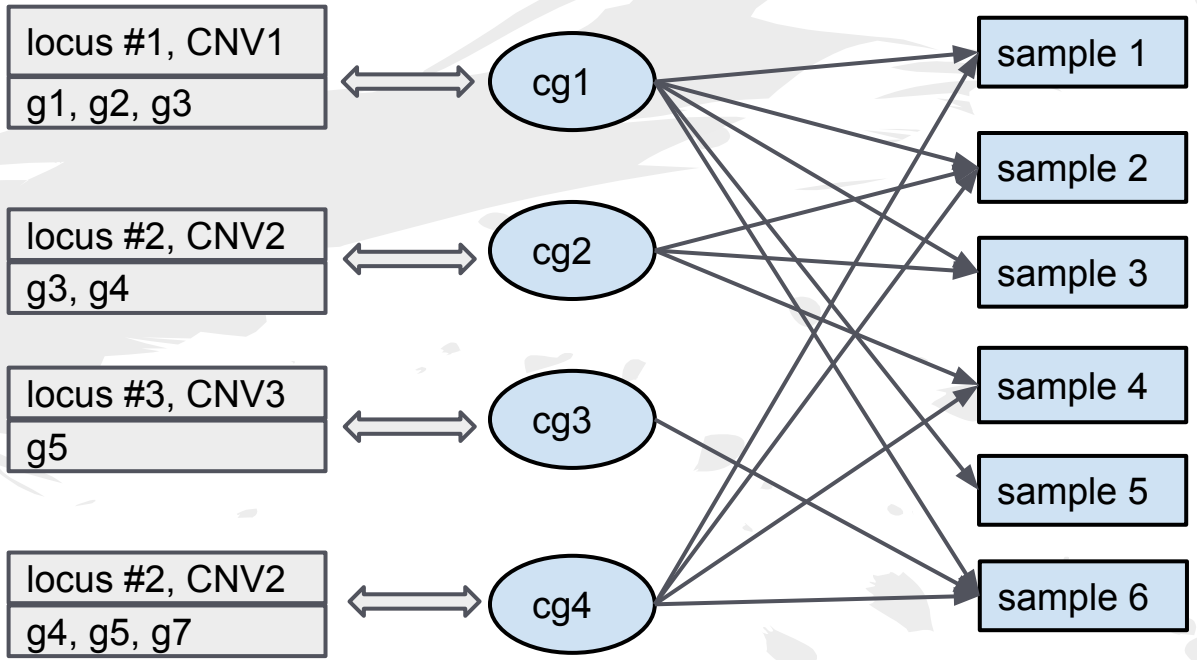
Choose a path to get max value of current for all its genes:



Selecting final causal genes

We stop if coverage weight is more or equal to W. That must be true for all samples except B.

Let's consider:
W = 2, B = 2

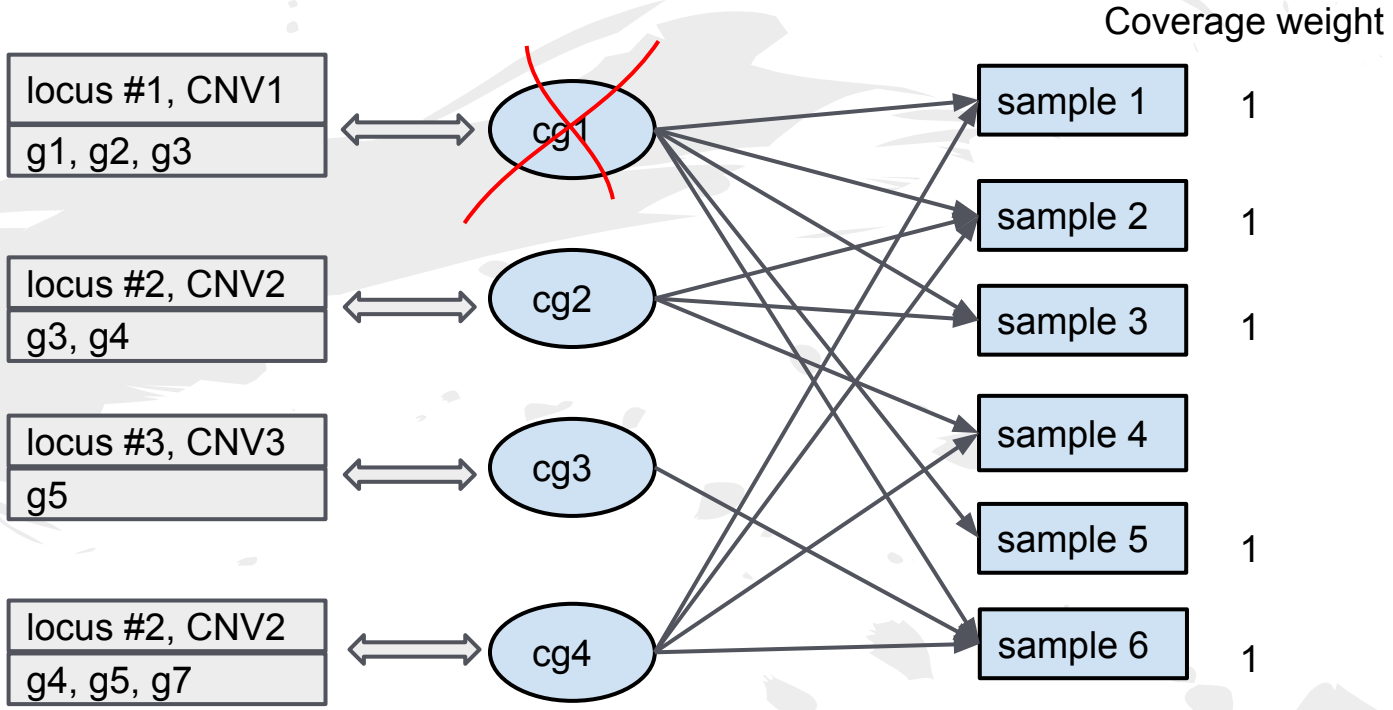


We want to select minimum number of relevant genes connected with samples

Selecting final causal genes

We stop if coverage weight is more or equal to W. That must be true for all samples except B.

Let's consider:
W = 2, B = 2

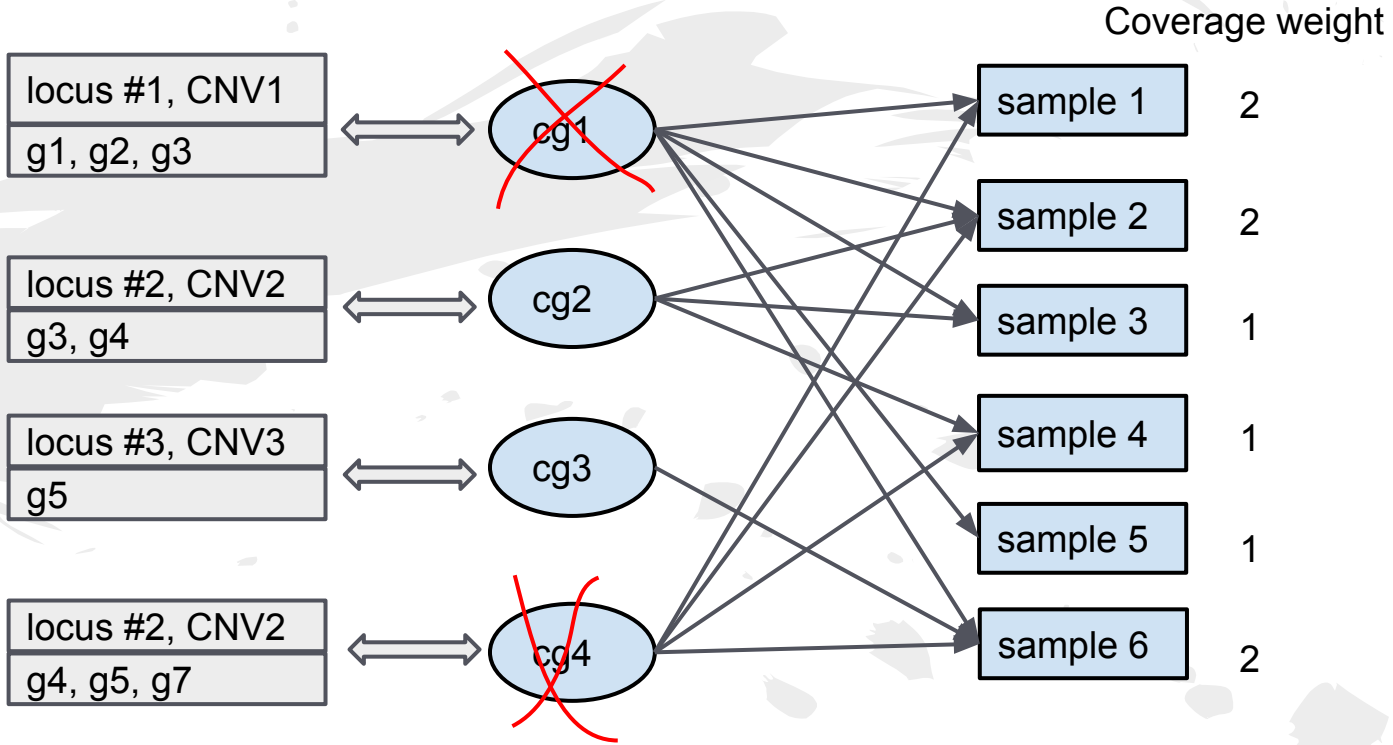


Take cg1 because it has total weight 5 (edges count)

Selecting final causal genes

We stop if coverage weight is more or equal to W. That must be true for all samples except B.

Let's consider:
W = 2, B = 2

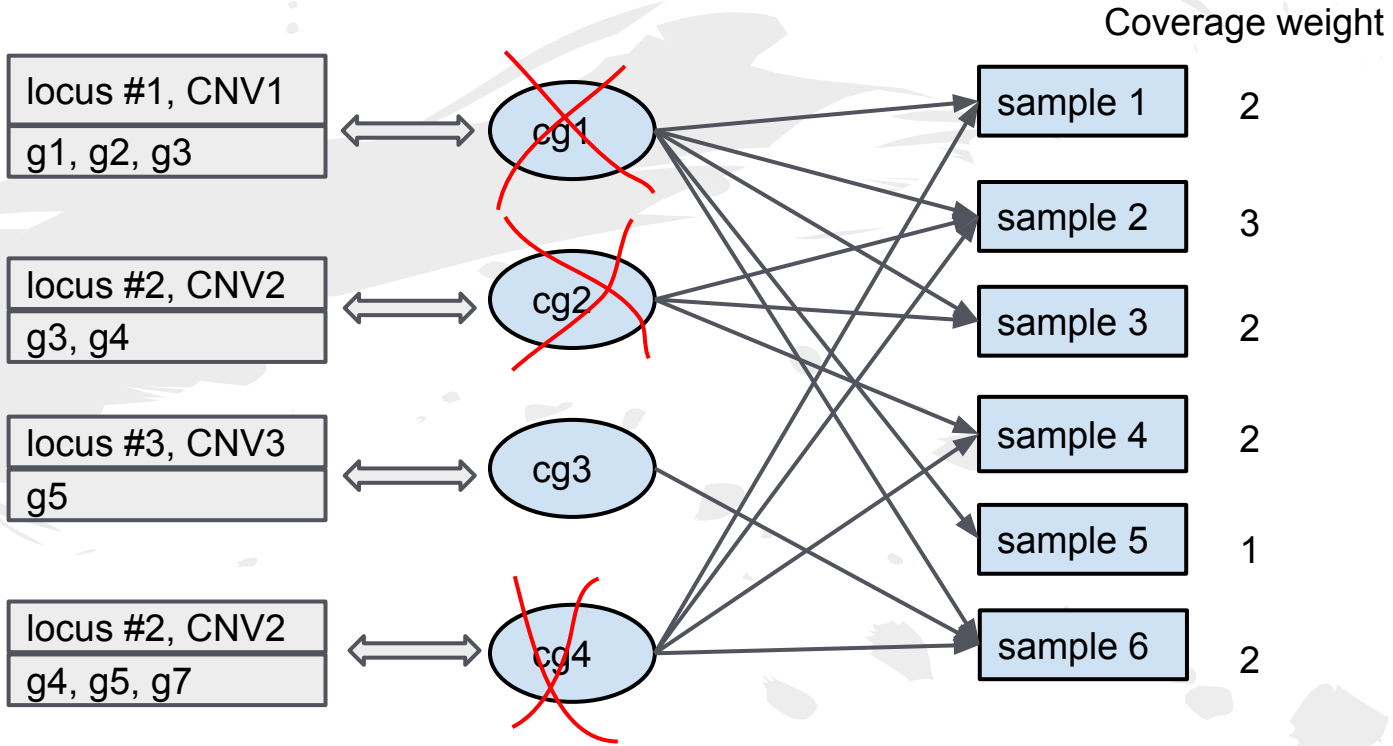


Take cg4 because it has total weight 4 (edges count)

Selecting final causal genes

We stop if coverage weight is more or equal to W. That must be true for all samples except B.

Let's consider:
W = 2, B = 2



Take cg2 because it has total weight 3 (edges count).
Hooray! We've just got: {cg1, cg2, cg4}

Implementation details

A lot of parsing data

Work with statistics

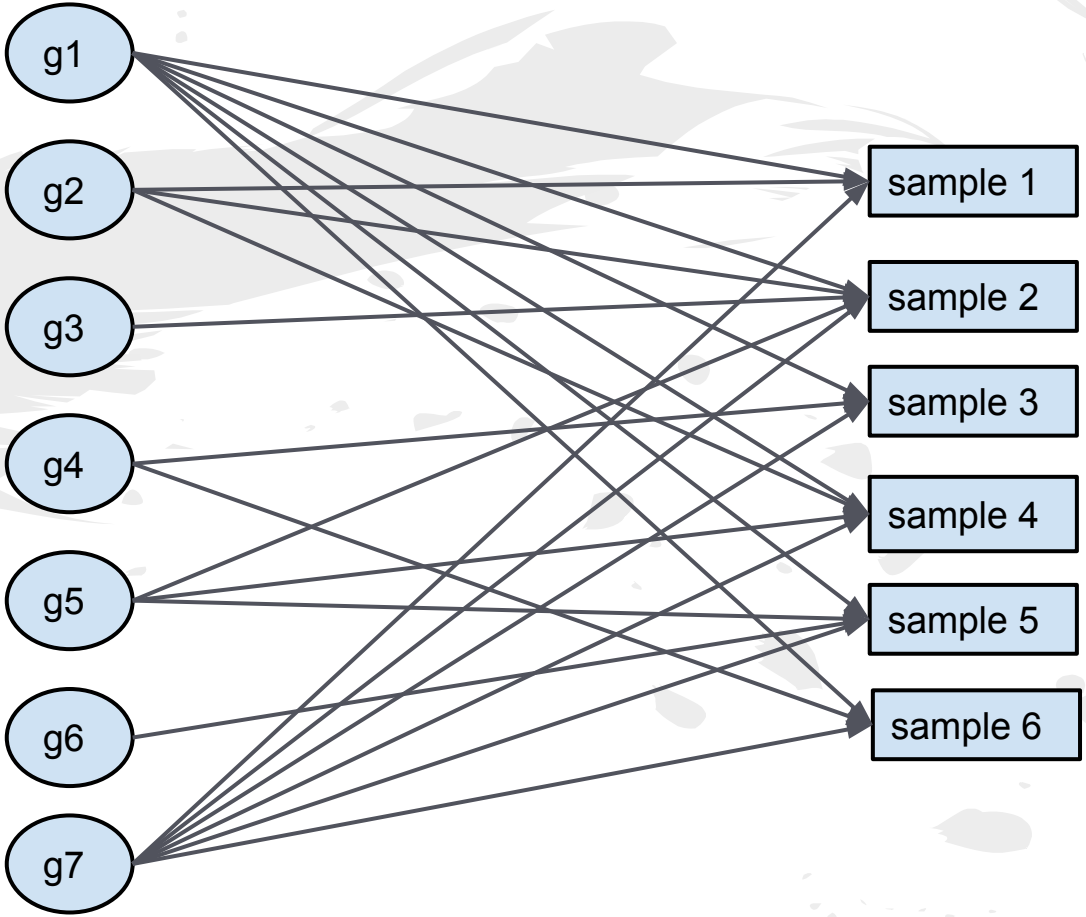
Python + numpy + networkx?

Work with graph-like structures

Avoiding one bottleneck

We stop if coverage is more or equal to A. That must be true for all samples except B.

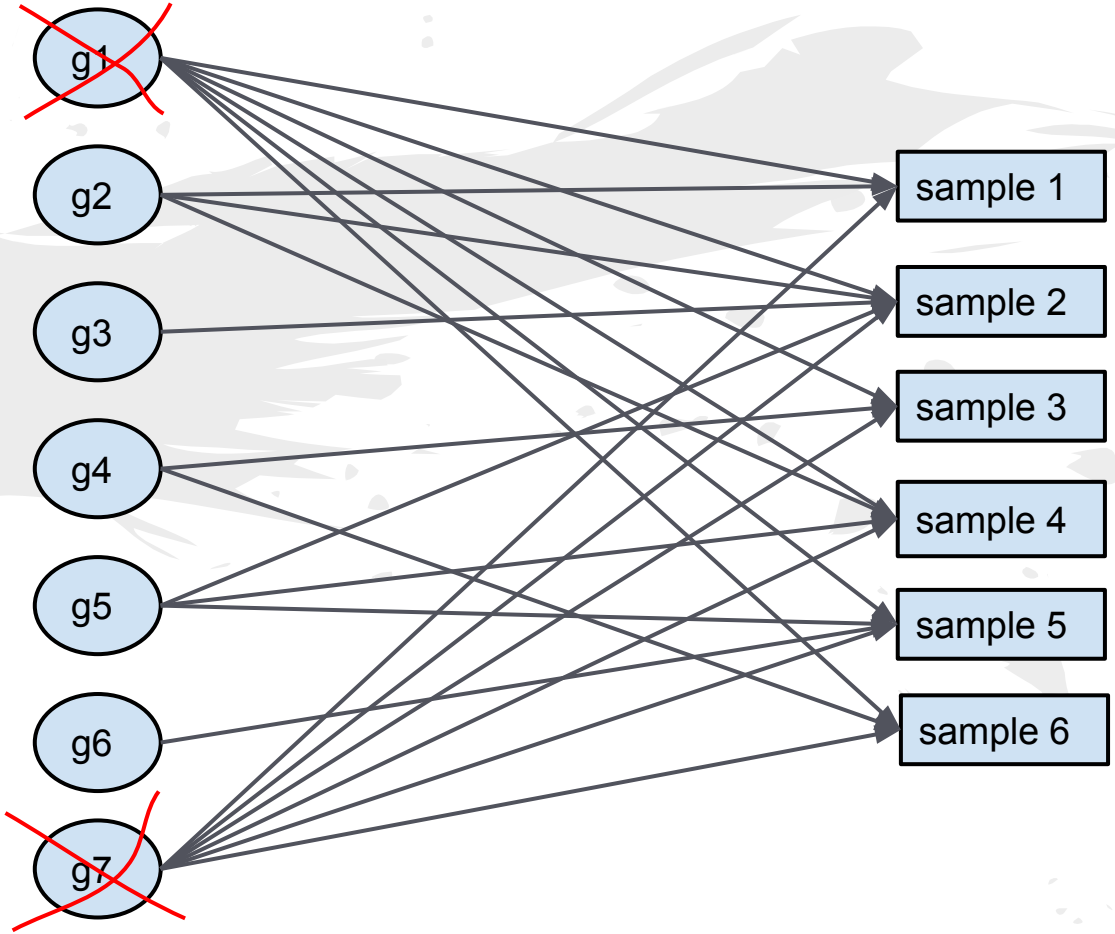
Let's consider:
A = 2, B = 2



Avoiding one bottleneck

We stop if coverage is more or equal to A. That must be true for all samples except B.

Let's consider:
A = 2, B = 2



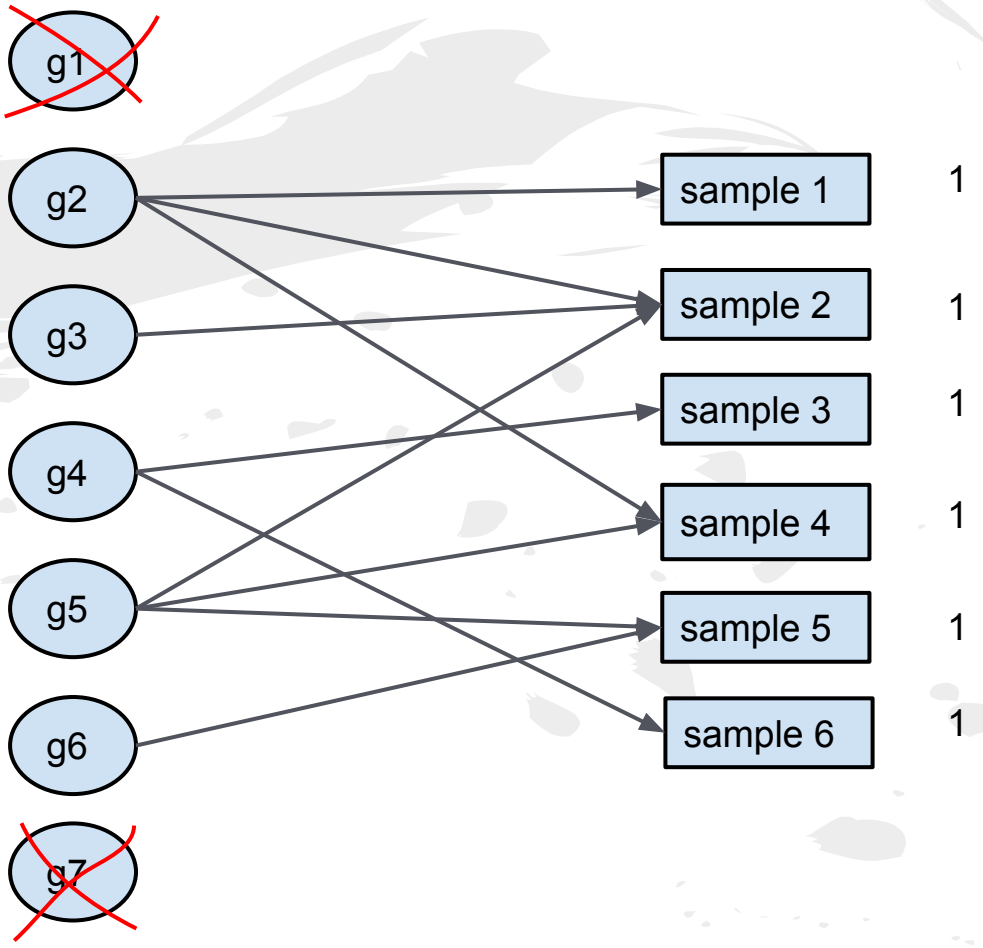
OK. We've just got {g1, g7}.

Hey!
What is about g2 and g4???

Avoiding one bottleneck

Possible solution:

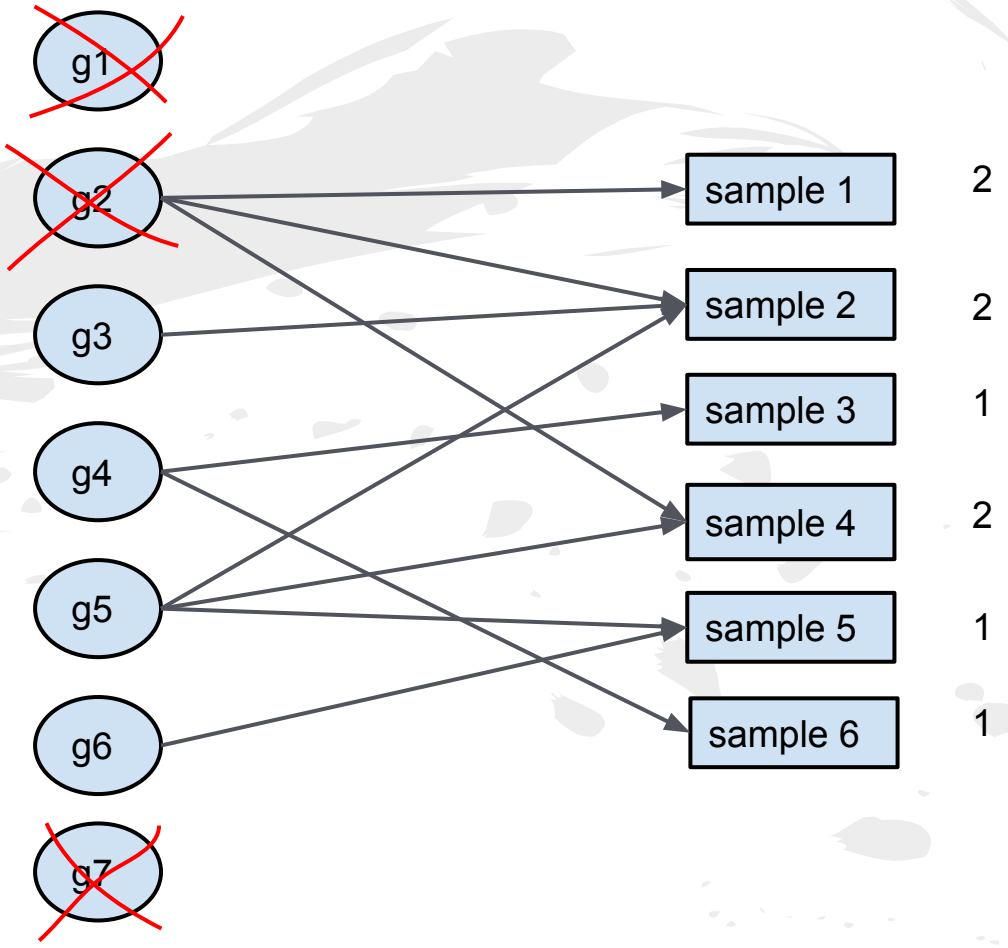
- take those genes that have twice more connections that all other genes;
- consider significant genes as exactly one gene;
- run usual algorithm without these significant genes;



Avoiding one bottleneck

We stop if coverage is more or equal to A. That must be true for all samples except B.

Let's consider:
A = 2, B = 2



Avoiding one bottleneck

We stop if coverage is more or equal to A. That must be true for all samples except B.

Let's consider:
A = 2, B = 2

We've just got {g1, g7, g2, g5}.

Improvements to the existing tool

- Target genes selection algorithm enhancement;
- Refactoring of source code;
- User-friendliness;
- Source code lines reduction;
- Github repo:

  https://github.com/RaiaN/netQTL

Questions?