```
# Install packages and test if they can be loaded
# with library(<package>)

install.packages("reshape2")
install.packages("microbenchmark")
install.packages("RColorBrewer")
install.packages("Rcpp")
install.packages("data.table")

# Set working directory to location of scripts
setwd(<location of SpeedUpR scripts>)
```

Josephine Daub

R-Ladies meetup

December 11, 2018

How to speed up your R code?

Josephine Daub

R-Ladies meetup

December 11, 2018

My experience with R and other languages

- Master Computational Science UVA:
  - course Bioinformatics 2: R
  - master project: C++
  - Perl, C, Mathematica, Matlab, Python, …
- PhD project: mostly in R
- 1st Postdoc: bash, R
- Current postdoc: R, bash, Python

For children. Against cancer.

Princess MÁXIMA
center for pediatric oncology

Speeding up your R code

- Timing & **Profiling:** Which part of my code is slow?
- **Simple tricks** to speed up, using: memory allocation, defaults, matrices instead of dataframes, which/ifelse
- **Avoiding for-loops**: vectorization, matrix manipulation, apply, build-in R functions
- **Convert** parts of your code (for-loops) to c++ with **rcpp**
- Alternatives to the (slow) **aggregate** function: datatables, sapply
- Not today: **parallelization**

Our study: Genetic Interactions in Childhood Cancer

- Genetic interaction/epistasis:

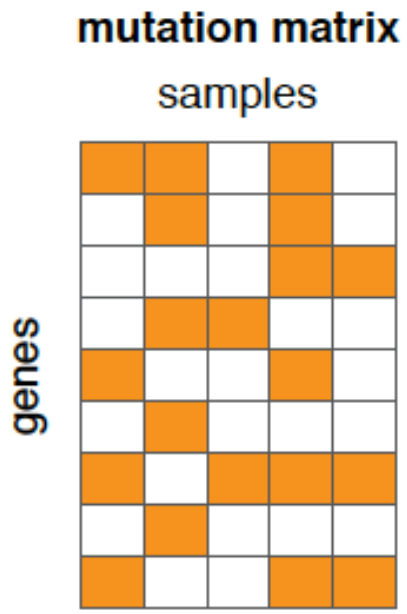
  The **phenotype (outcome)** of combining two mutations is **unexpected** given the phenotype of the single mutations

- **Genetic interactions** between mutated genes can promote or hinder cancer progression

- Find **pairs of mutated genes** that **co-occur** more (or less) often than expected given the frequency of the individual mutated genes

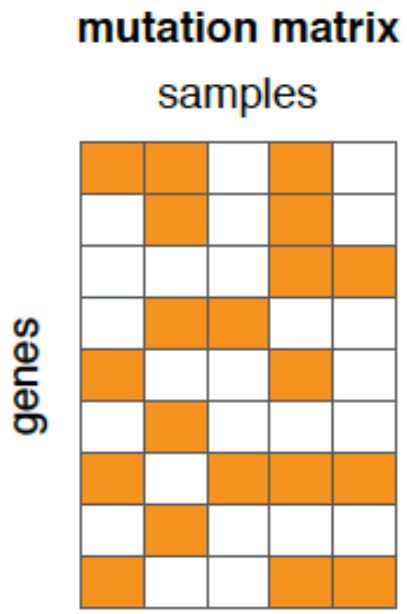
Methods to detect genetic interactions

- Create **Sample - Gene matrix**



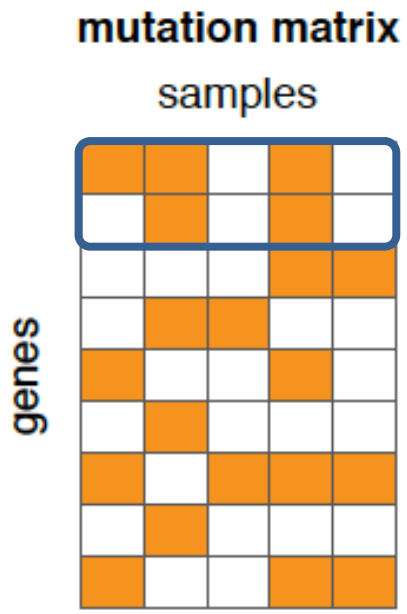
Methods to detect genetic interactions

- Create **Sample - Gene matrix**
- **Count** per gene pair the number of **co-occurring** alterations



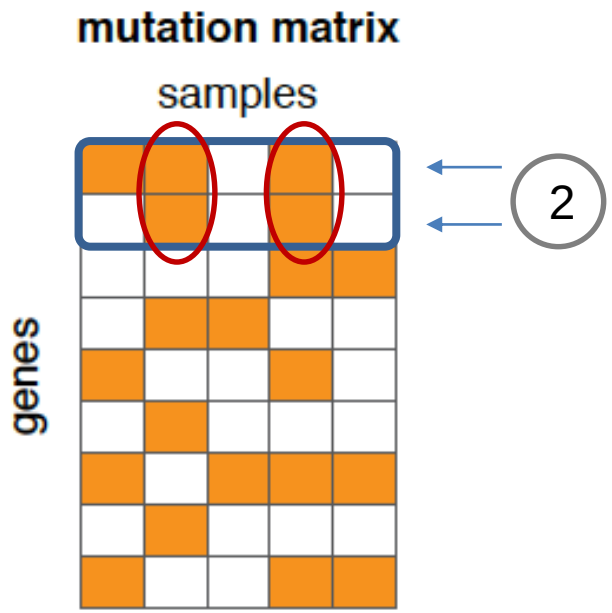
Methods to detect genetic interactions

- Create **Sample - Gene matrix**
- **Count** per gene pair the number of **co-occurring** alterations



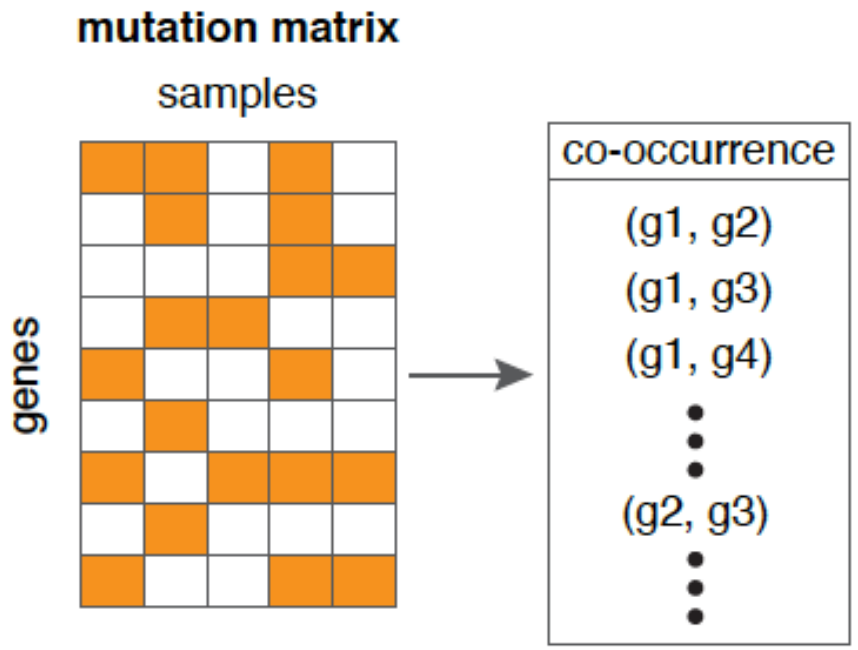
Methods to detect genetic interactions

- Create **Sample - Gene matrix**
- **Count** per gene pair the number of **co-occurring** alterations



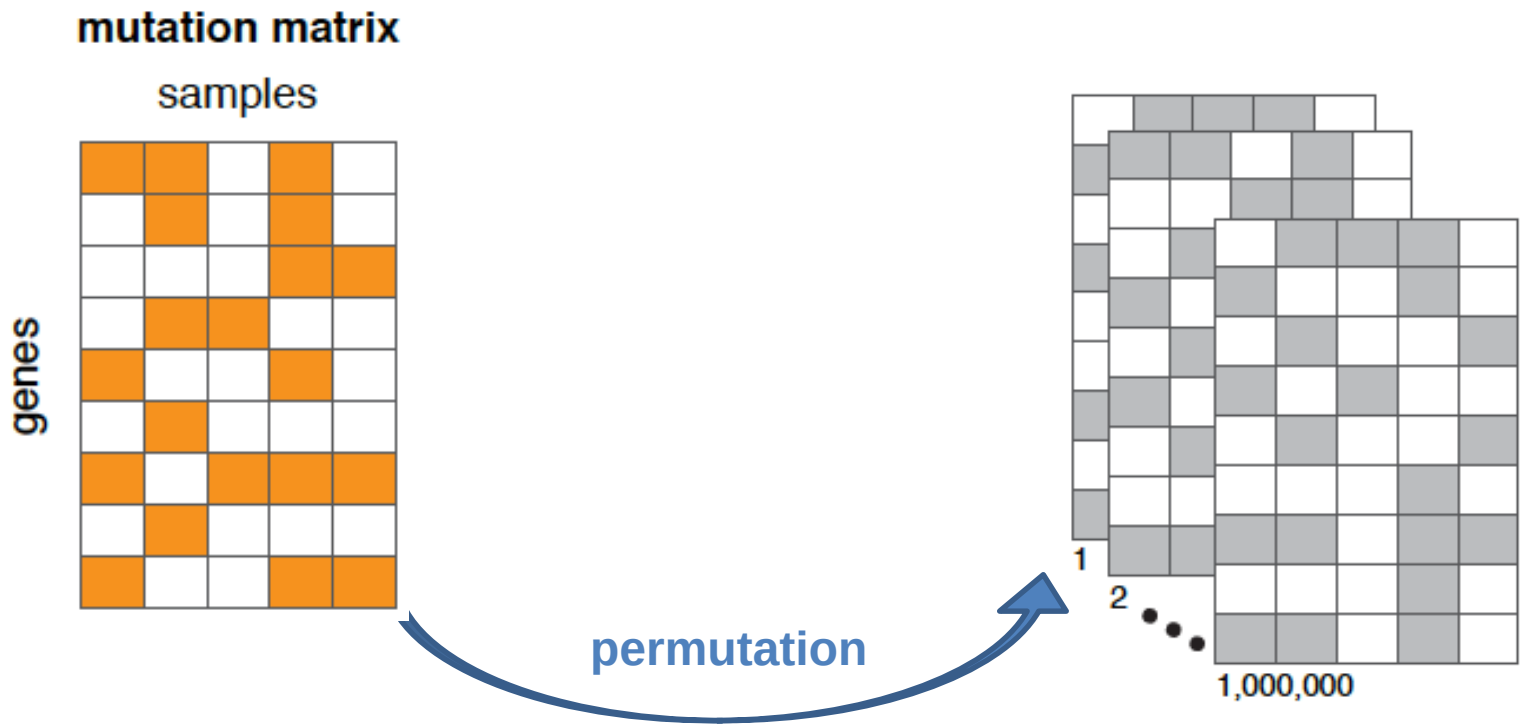
Methods to detect genetic interactions

- Create **Sample - Gene matrix**
- **Count** per gene pair the number of **co-occurring** alterations



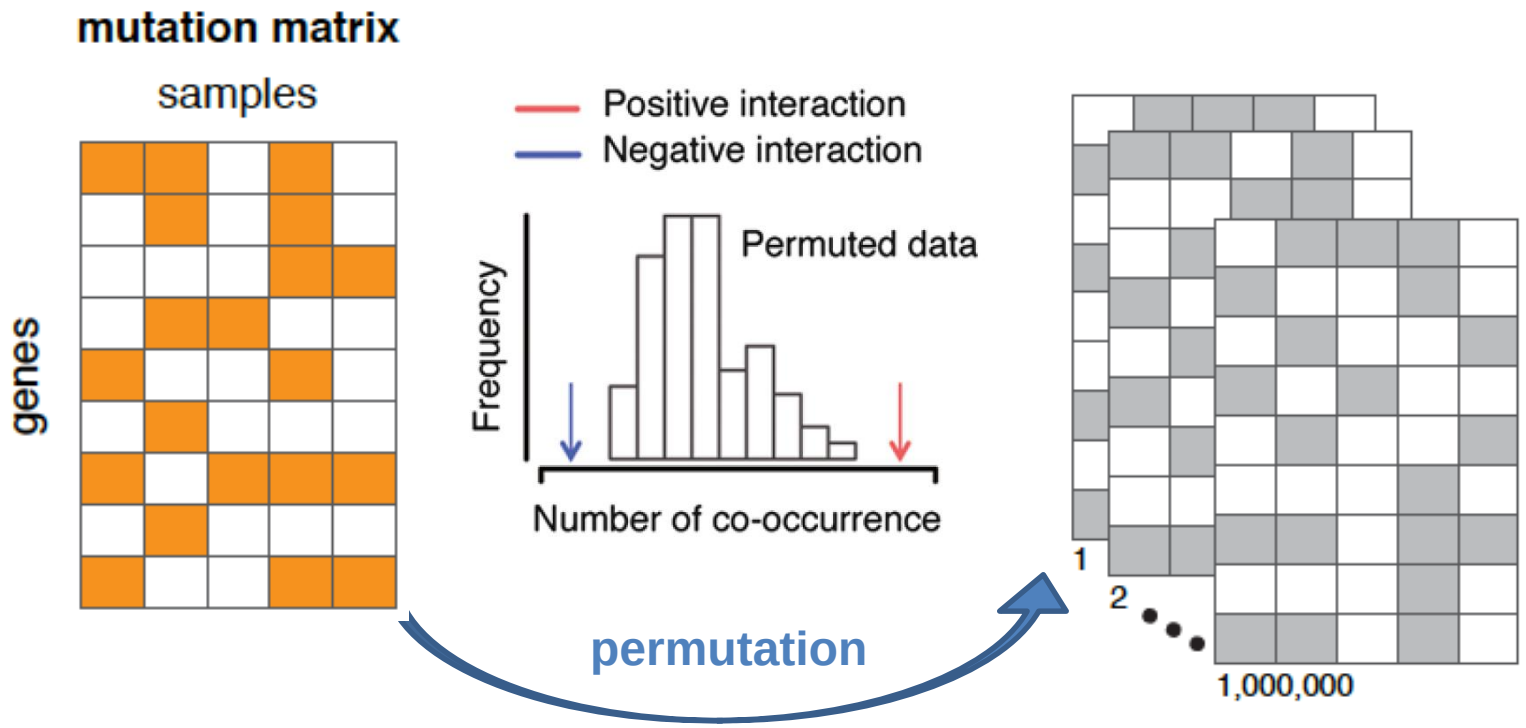
Methods to detect genetic interactions

- Create **Sample - Gene matrix**
- **Count** per gene pair the number of **co-occurring** alterations
- **Permute** the **matrix** to create **null distribution**, but keep margins fixed



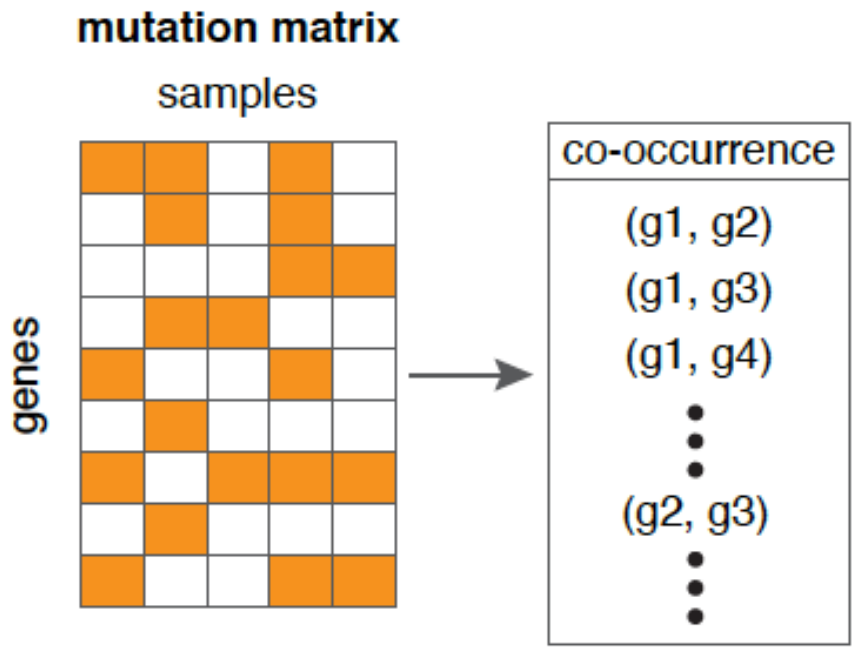
Methods to detect genetic interactions

- Create **Sample - Gene matrix**
- **Count** per gene pair the number of **co-occurring** alterations
- **Permute** the **matrix** to create **null distribution**, but keep margins fixed



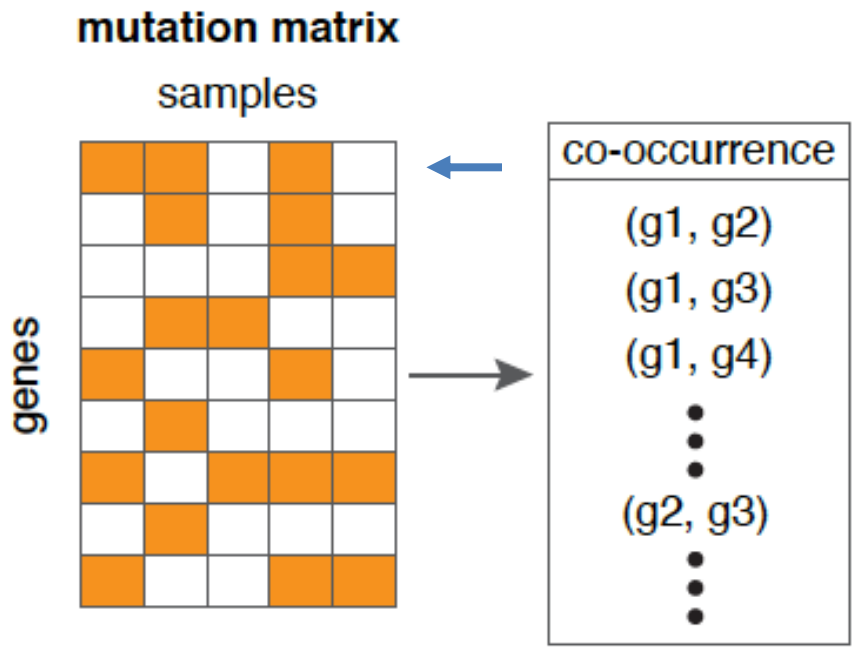
- Compare observed counts with null distribution to **infer significance** (p-value)

Implementation in R



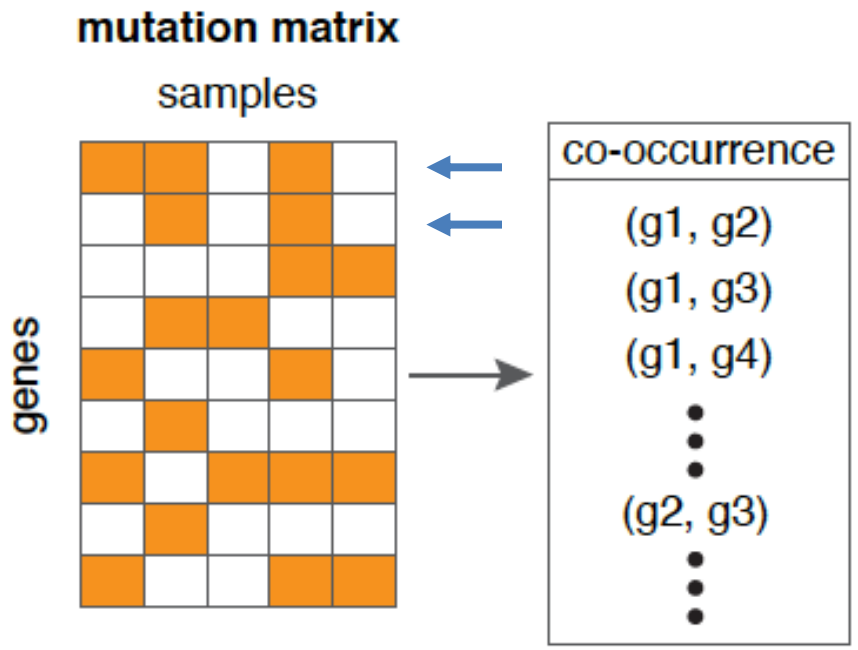
Implementation in R

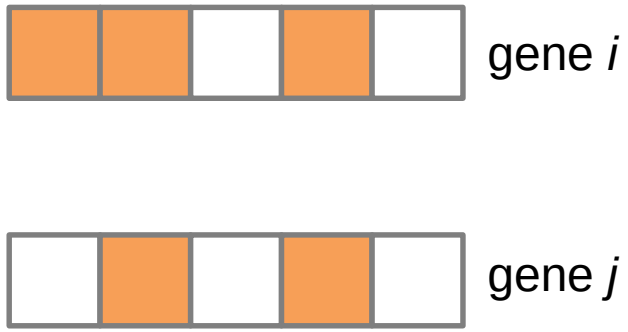
```
for each gene i
```



Implementation in R

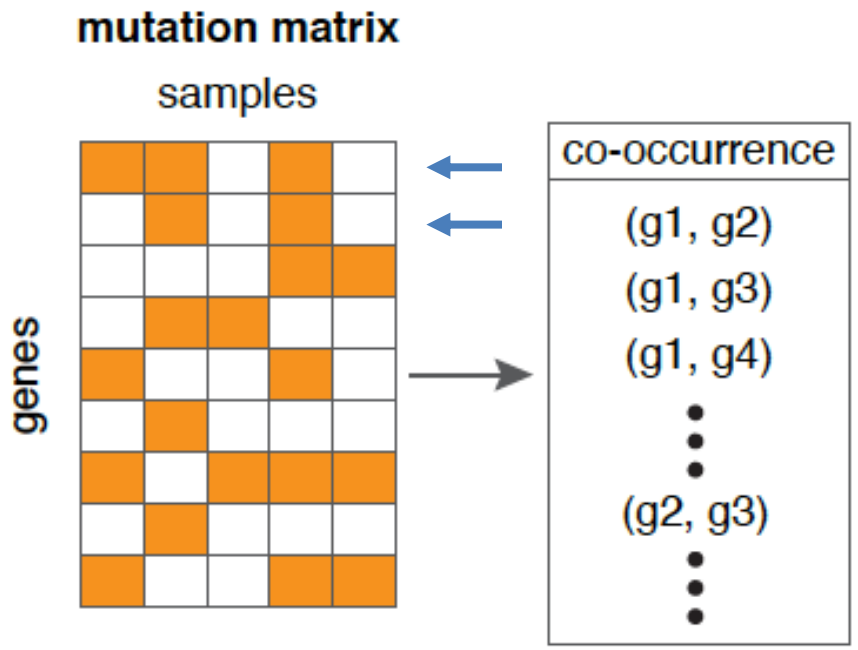
```
for each gene i
    for each gene j ≠ i
```

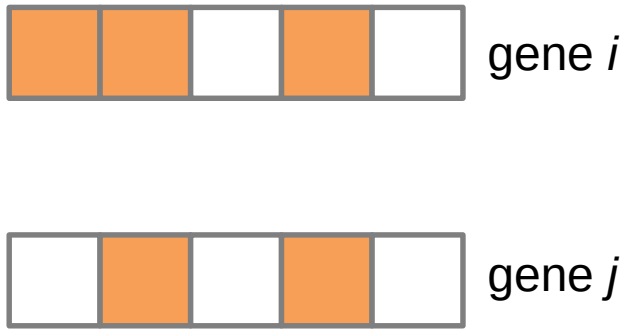




Implementation in R

```
for each gene i
    for each gene j ≠ i
        co_count_{i,j} = 0
```

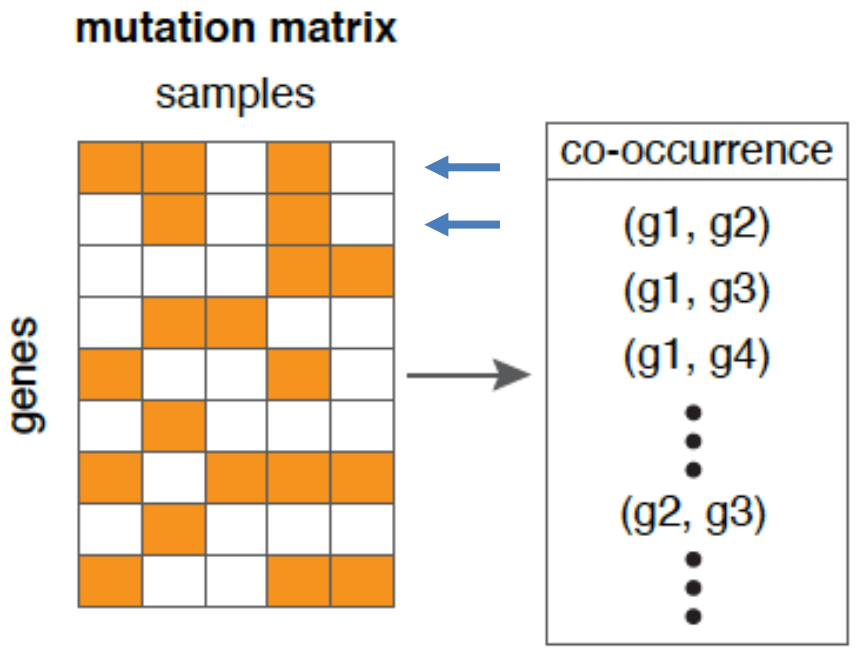


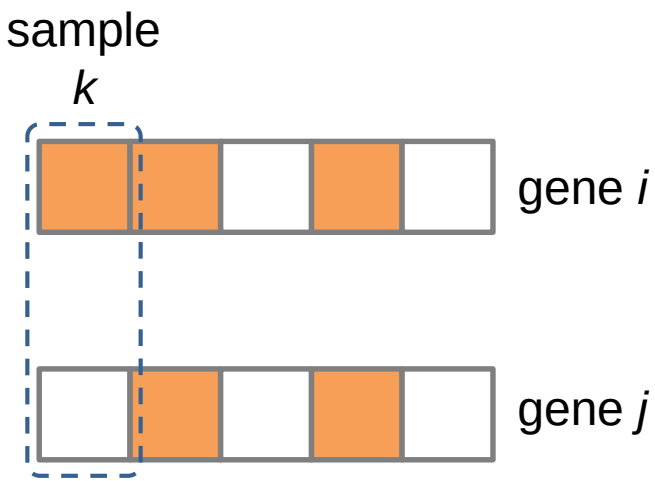


co_count
0

Implementation in R

```
for each gene i
    for each gene j ≠ i
        co_count_{i,j} = 0
        for each sample k
```

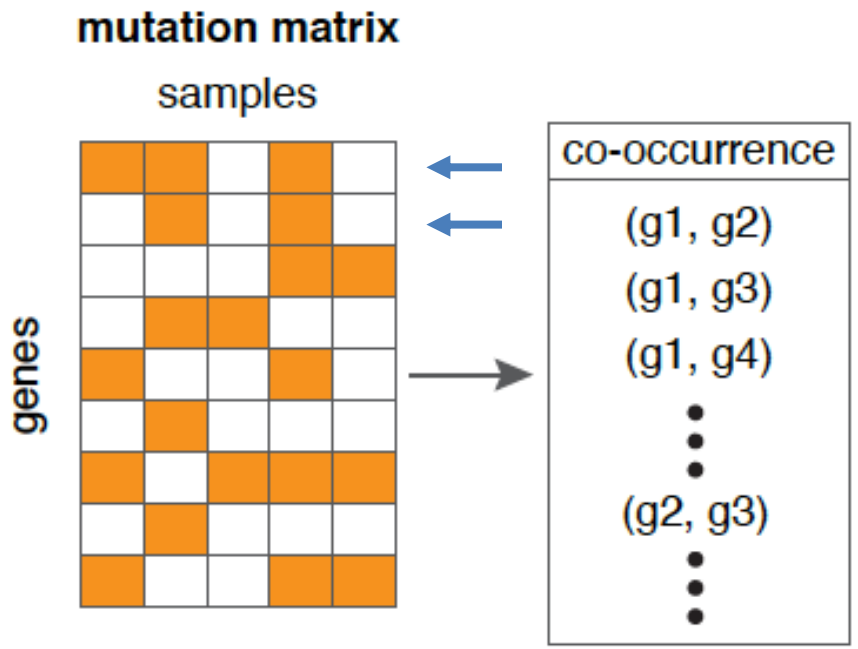


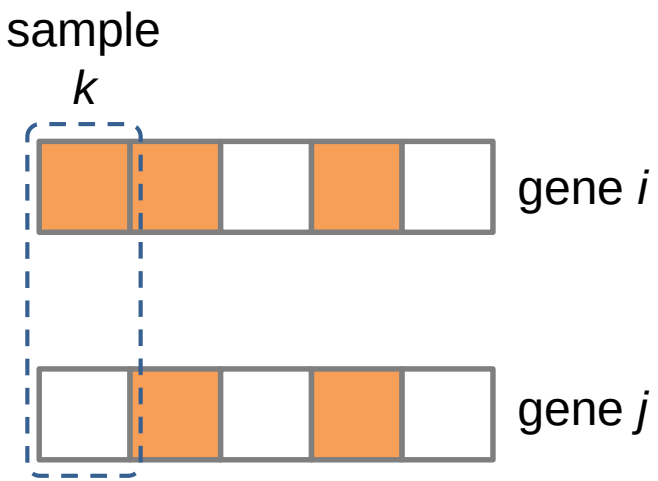


co_count
0

Implementation in R

```
for each gene i
    for each gene j ≠ i
        co_count_{i,j} = 0
        for each sample k
            if mtx_{i,k}==1 & mtx_{j,k}==1 then
```

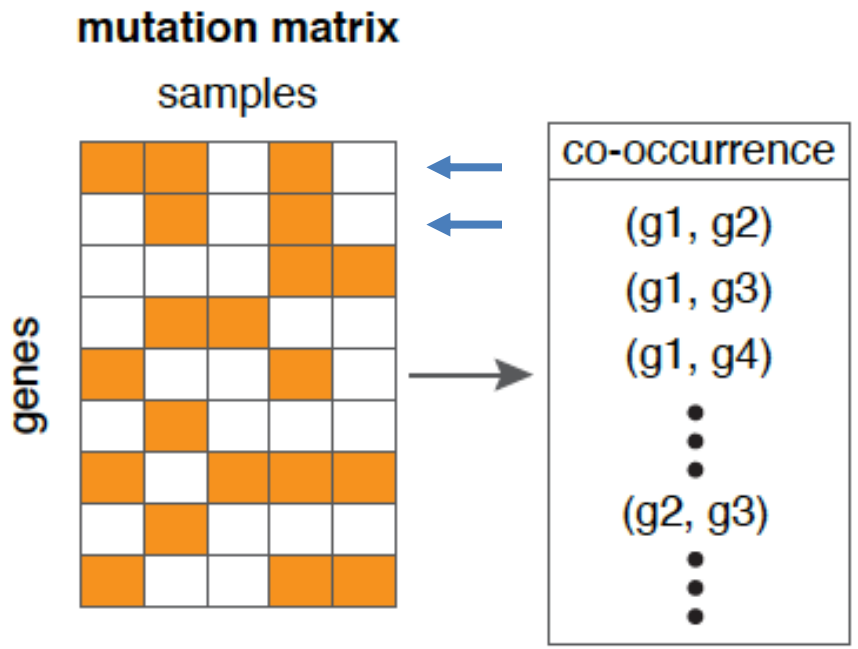


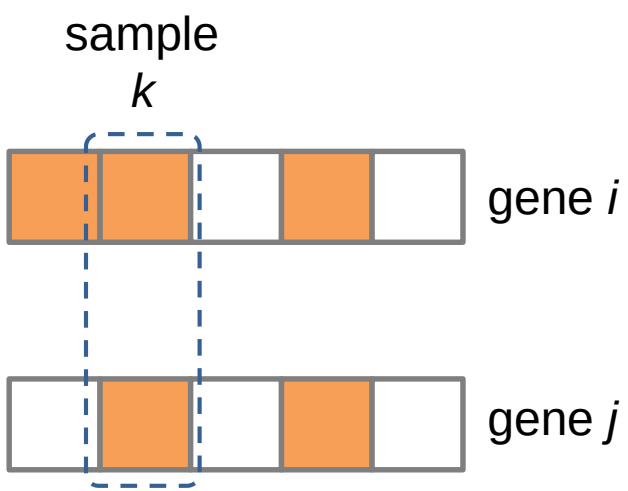


co_count
0

Implementation in R

```
for each gene i
    for each gene j ≠ i
        co_count_{i,j} = 0
        for each sample k
            if mtx_{i,k}==1 & mtx_{j,k}==1 then
```

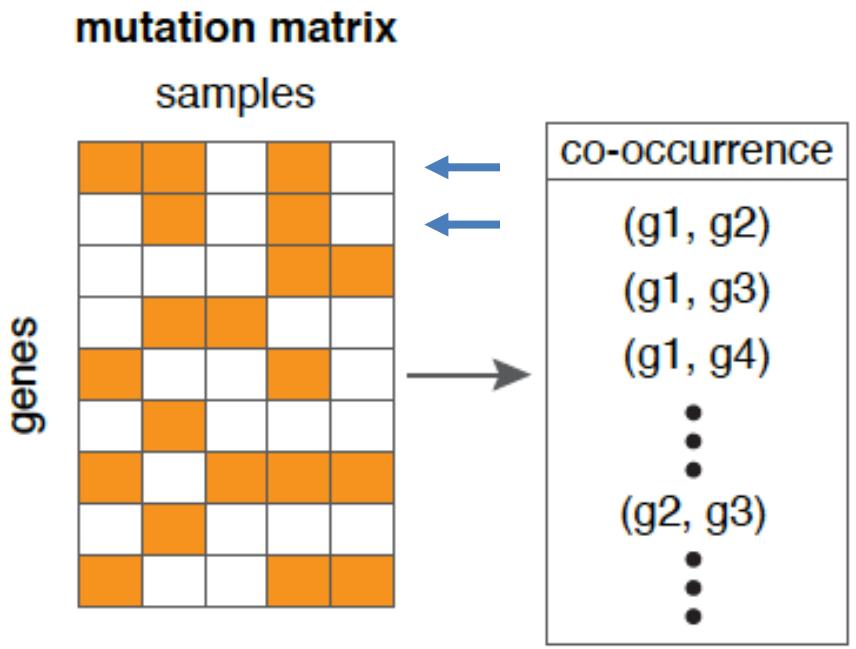


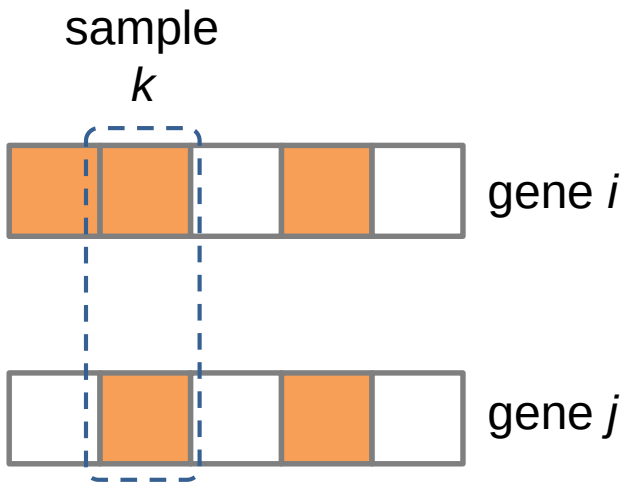


co_count
0

Implementation in R

```
for each gene i
    for each gene j ≠ i
        co_count_{i,j} = 0
        for each sample k
            if mtx_{i,k}==1 & mtx_{j,k}==1 then
                co_count_{i,j} = co_count_{i,j} + 1
```

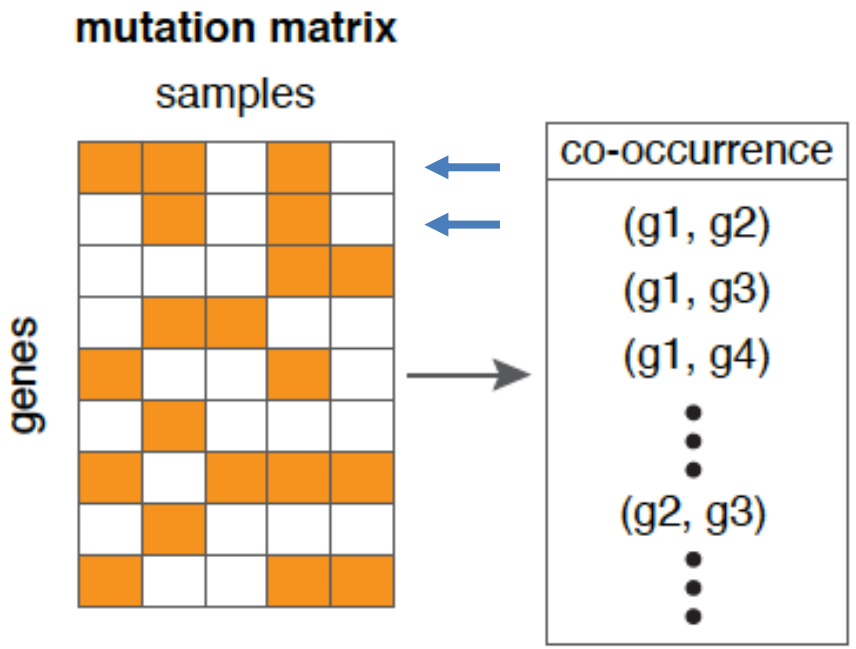


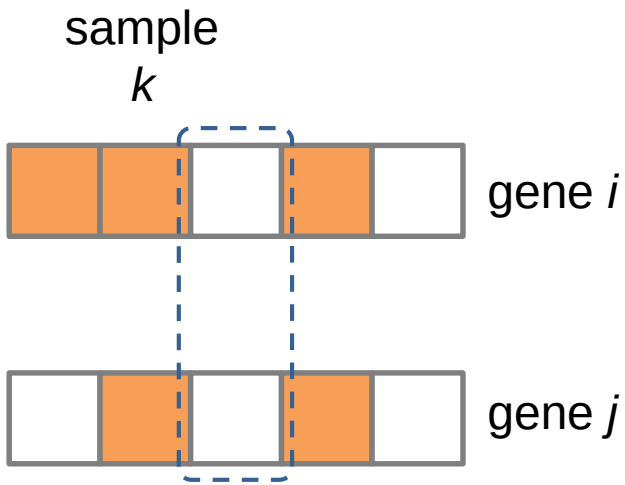


co_count
1

Implementation in R

```
for each gene i
    for each gene j ≠ i
        co_count_{i,j} = 0
        for each sample k
            if mtx_{i,k}==1 & mtx_{j,k}==1 then
                co_count_{i,j} = co_count_{i,j} + 1
```

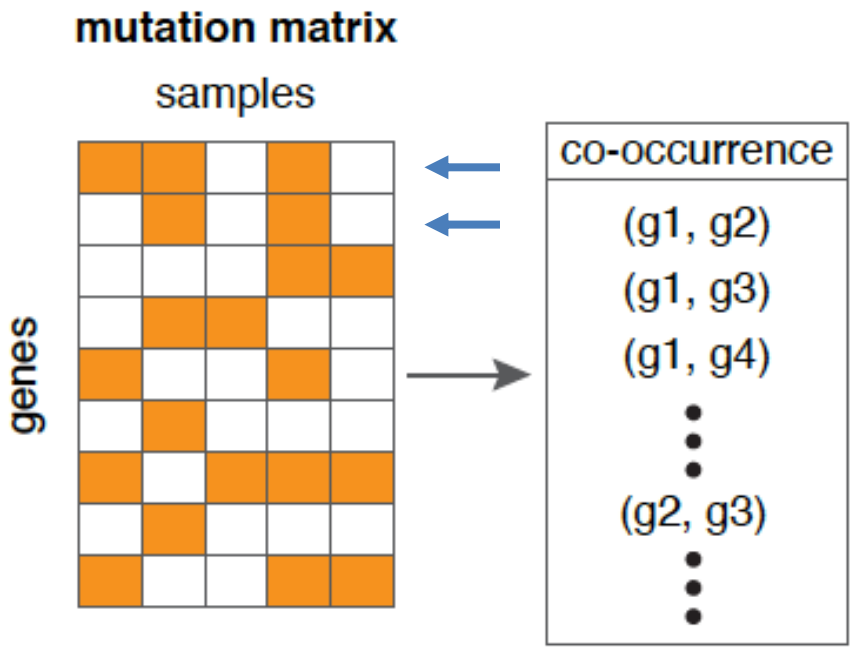


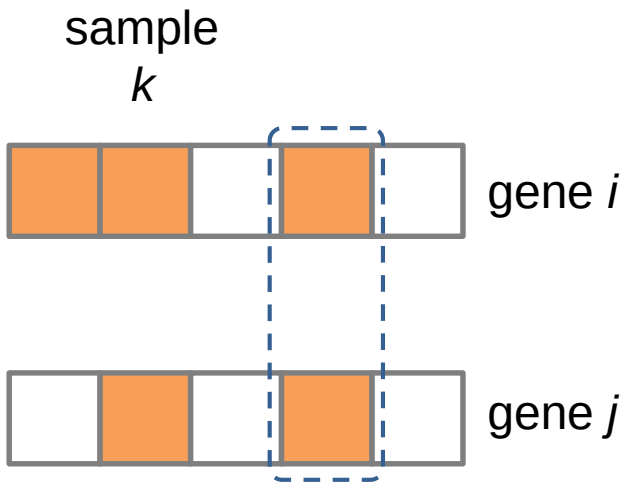


co_count
1

Implementation in R

```
for each gene i
    for each gene j ≠ i
        co_count_{i,j} = 0
        for each sample k
            if mtx_{i,k}==1 & mtx_{j,k}==1 then
                co_count_{i,j} = co_count_{i,j} + 1
```

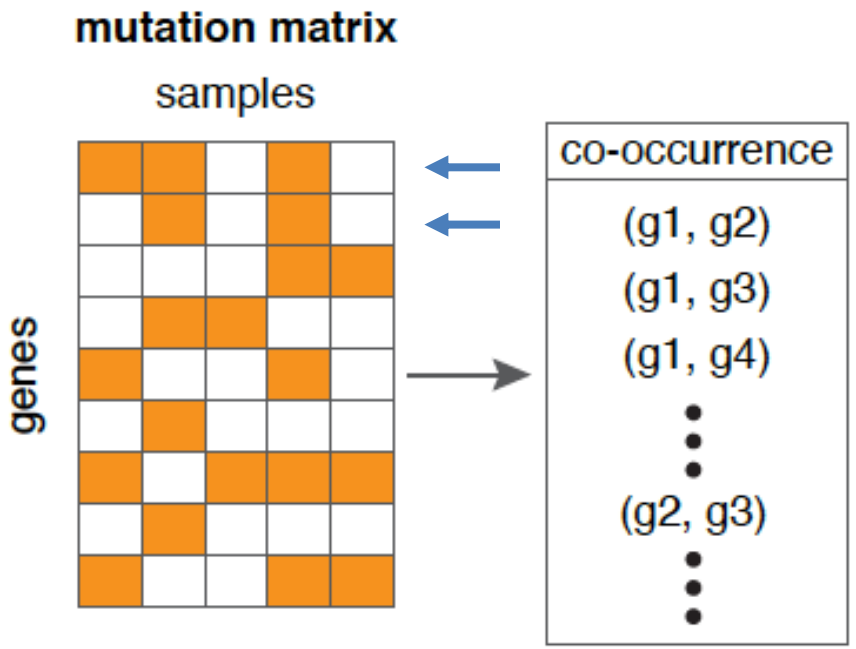


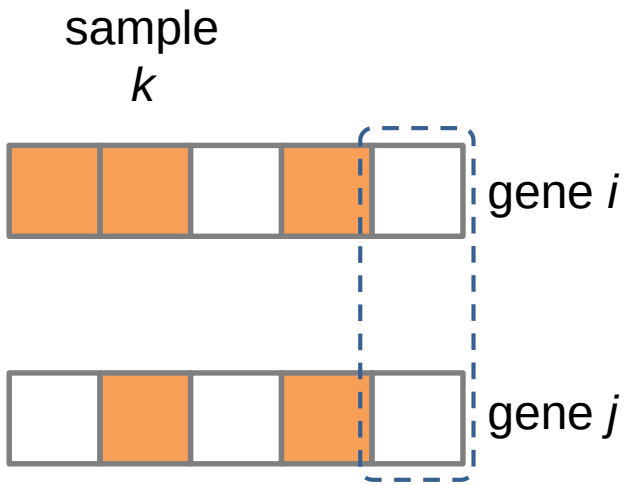


co_count
2

Implementation in R

```
for each gene i
    for each gene j ≠ i
        co_count_{i,j} = 0
        for each sample k
            if mtx_{i,k}==1 & mtx_{j,k}==1 then
                co_count_{i,j} = co_count_{i,j} + 1
```

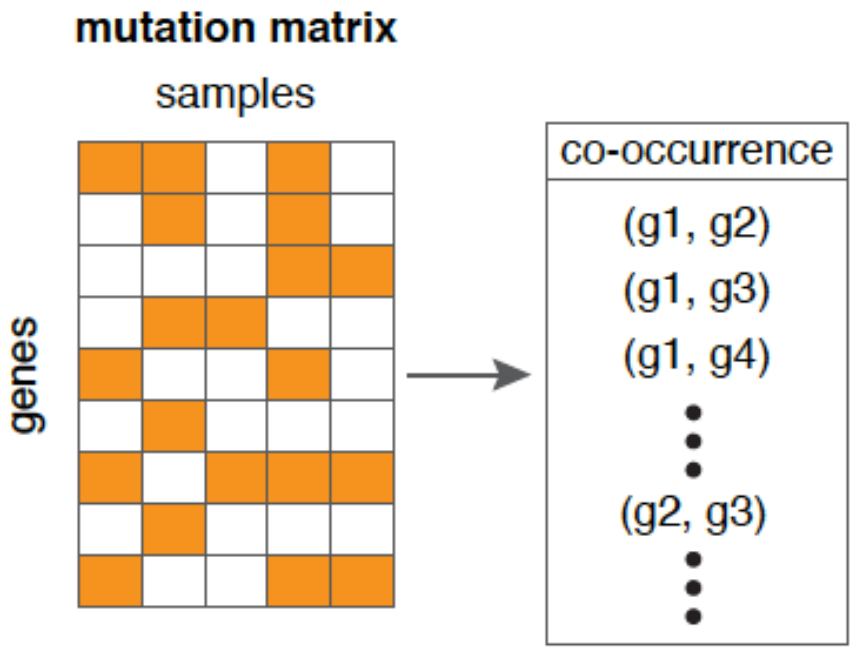


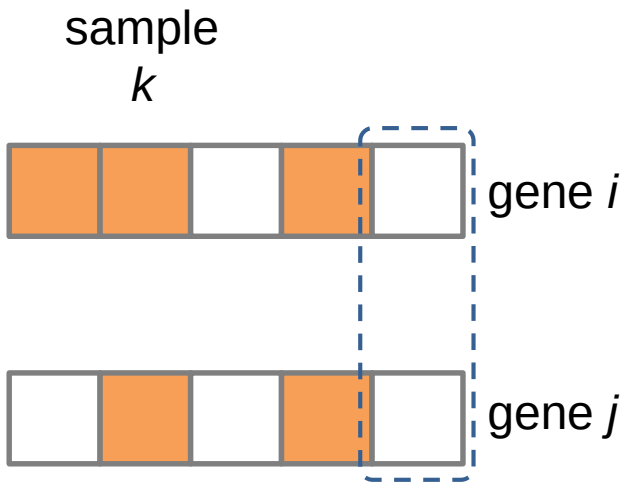


co_count
2

Implementation in R

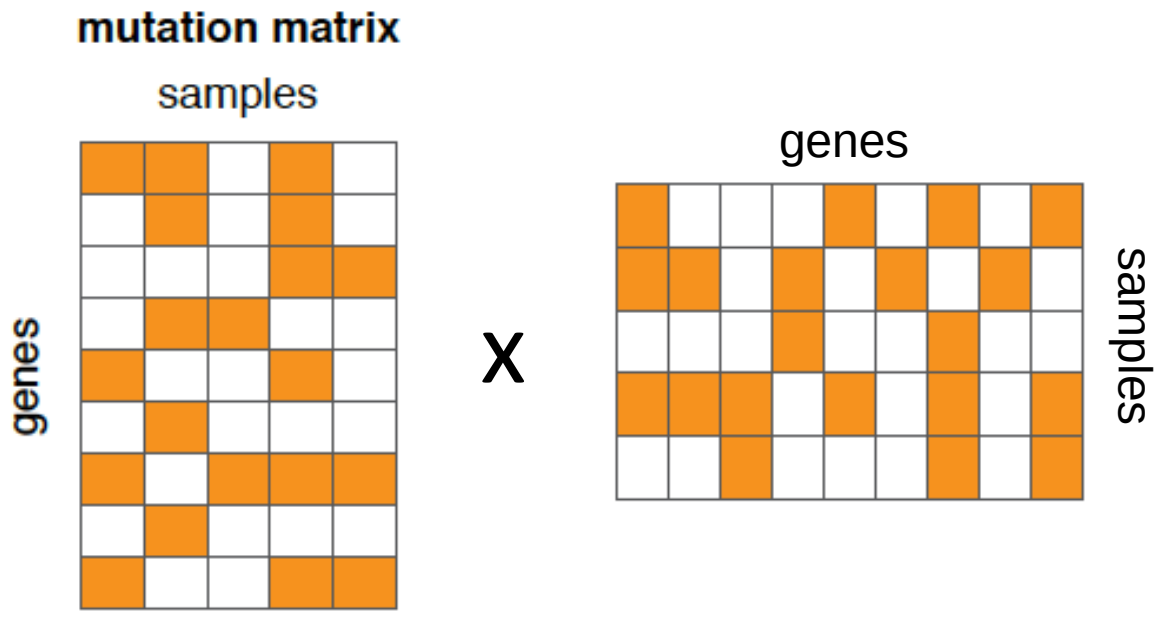
```
for each gene i
    for each gene j ≠ i
        co_count_{i,j} = 0
        for each sample k
            if mtx_{i,k}==1 & mtx_{j,k}==1 then
                co_count_{i,j} = co_count_{i,j} + 1
```



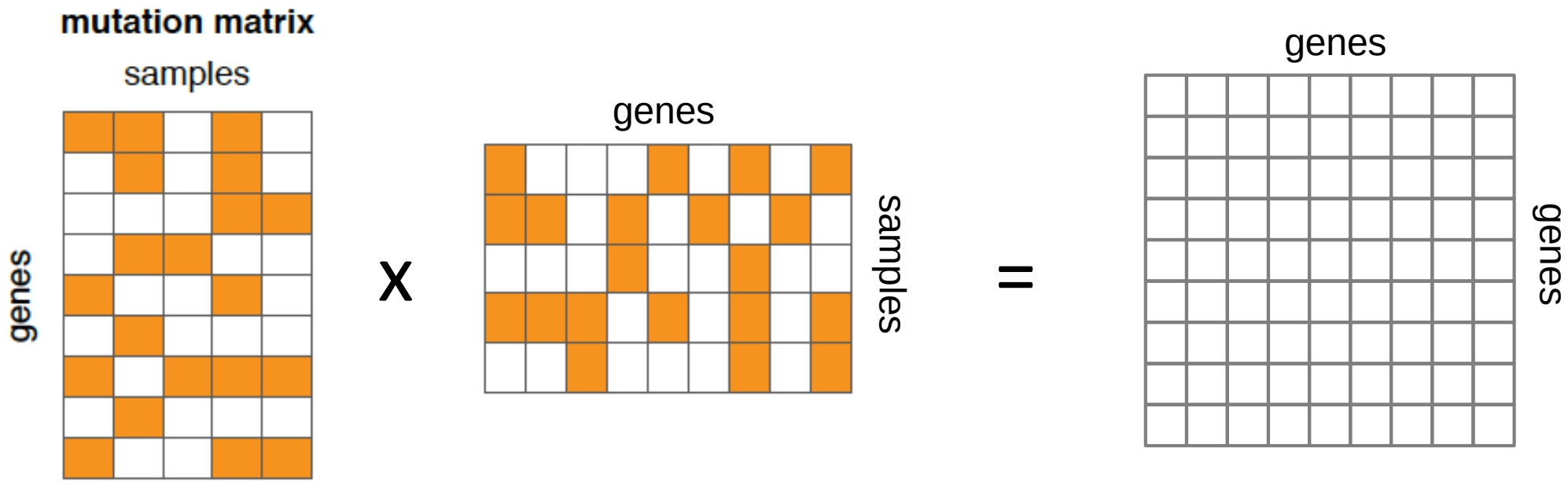


co_count
2

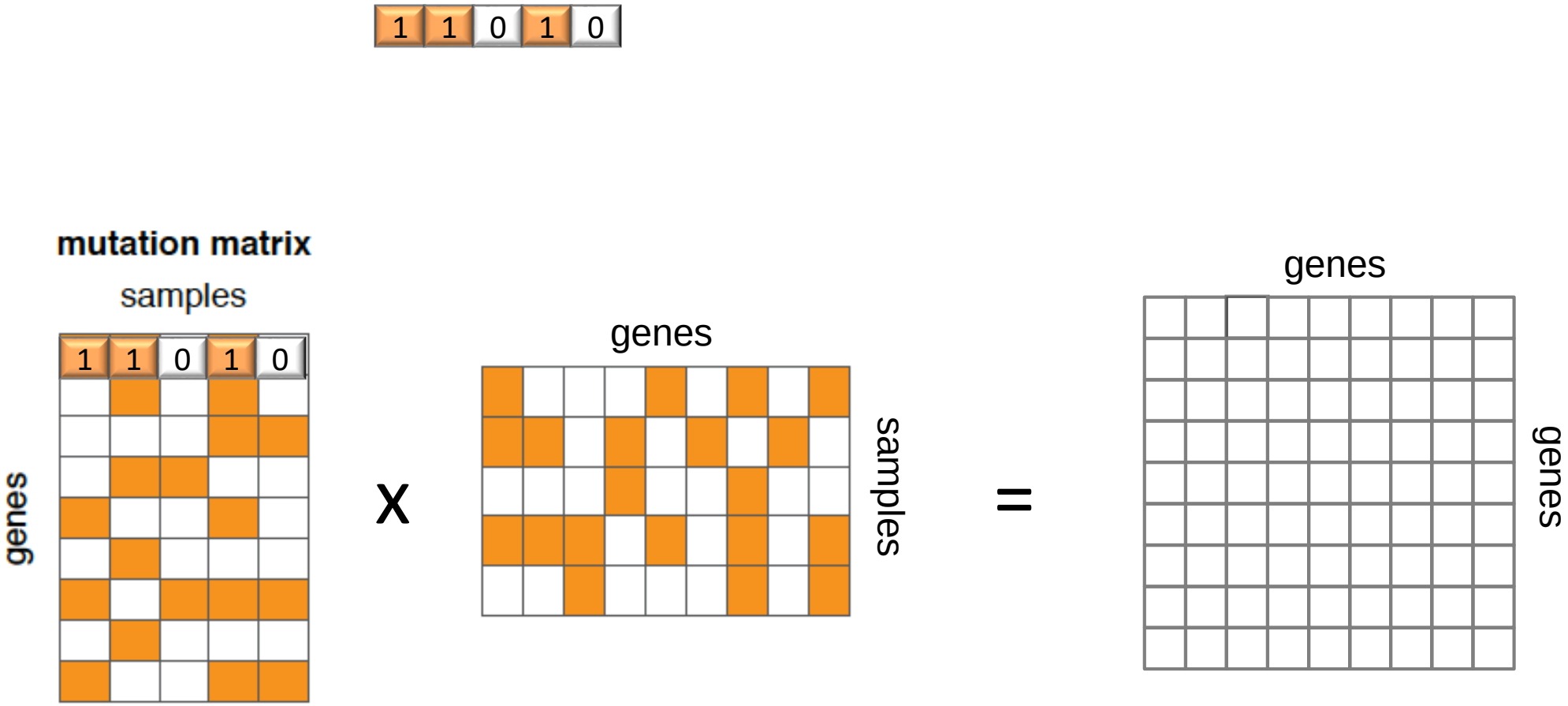
Matrix multiplication



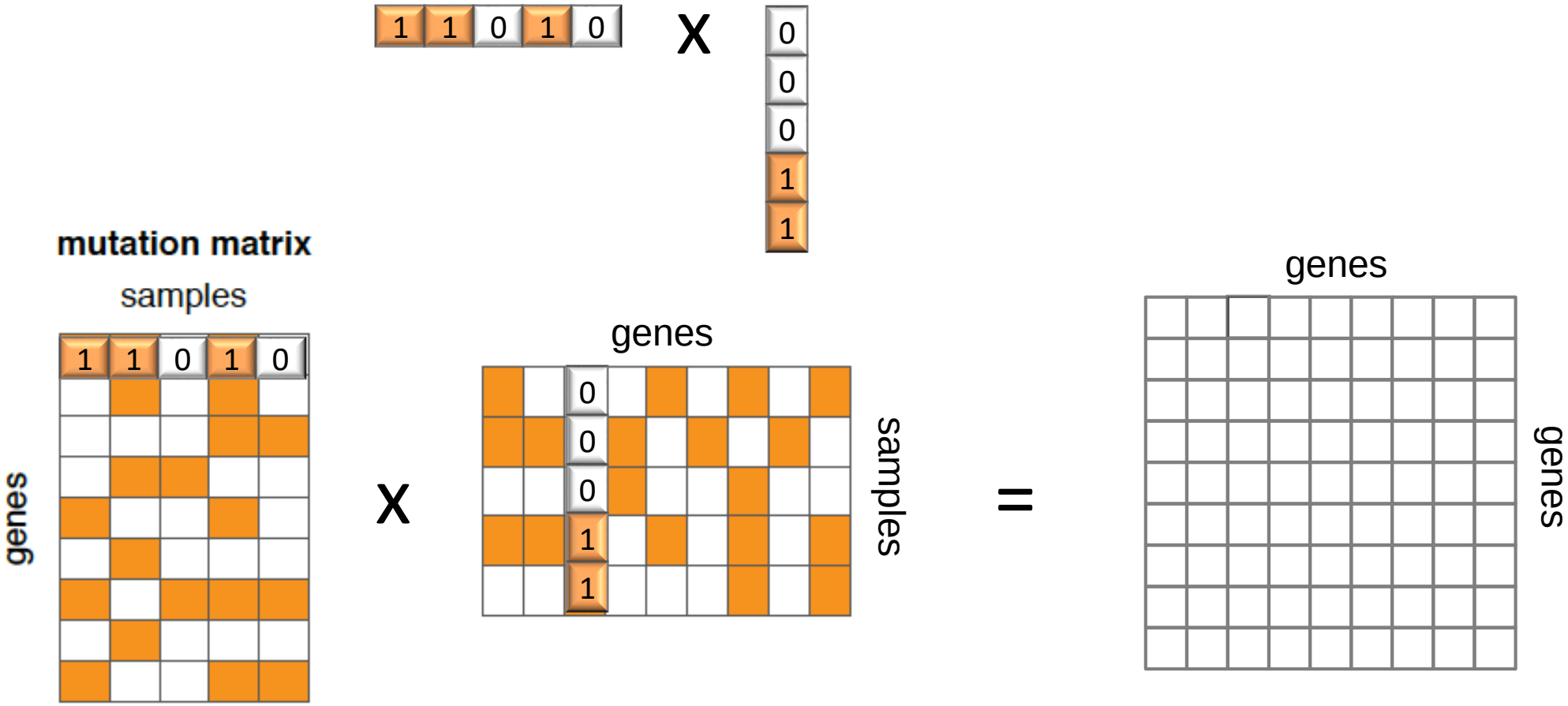
Matrix multiplication



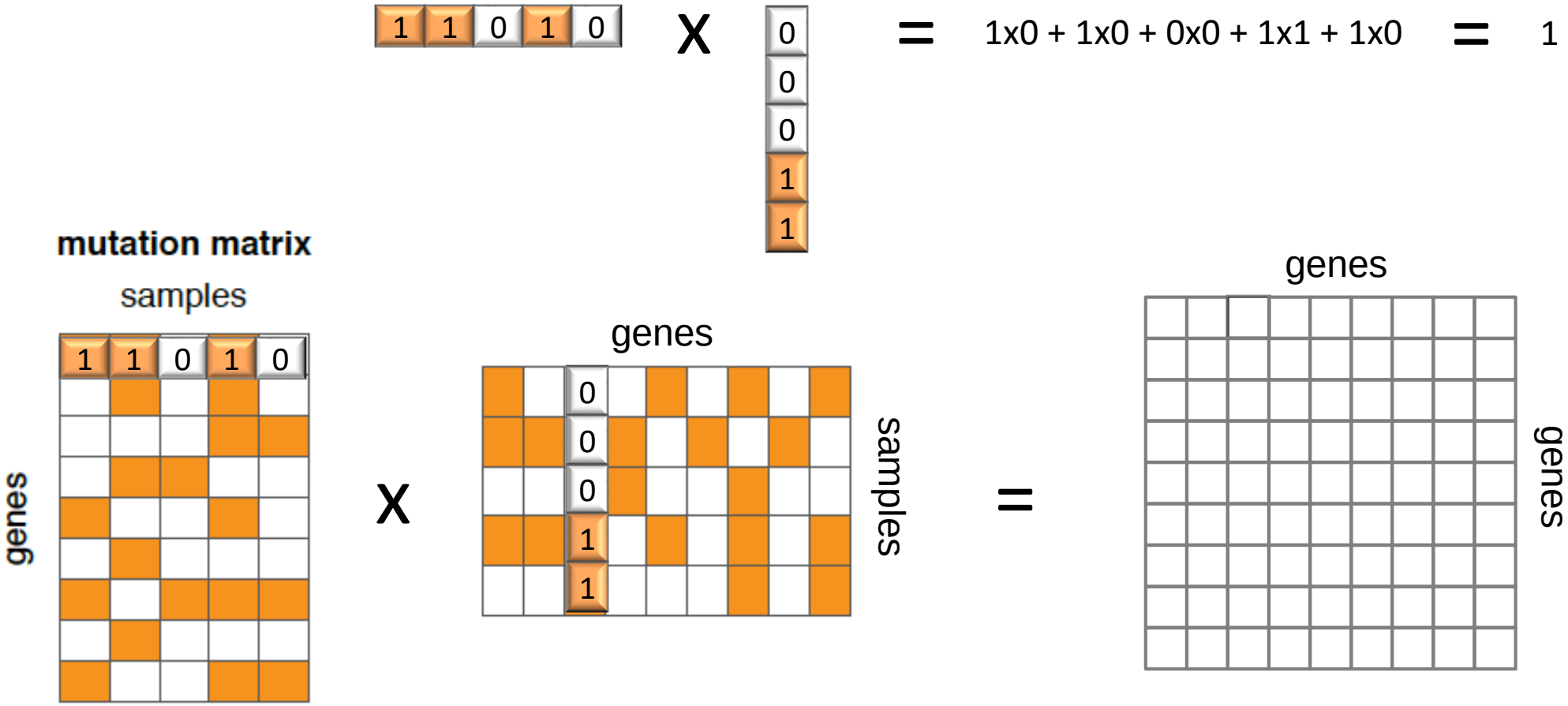
Matrix multiplication



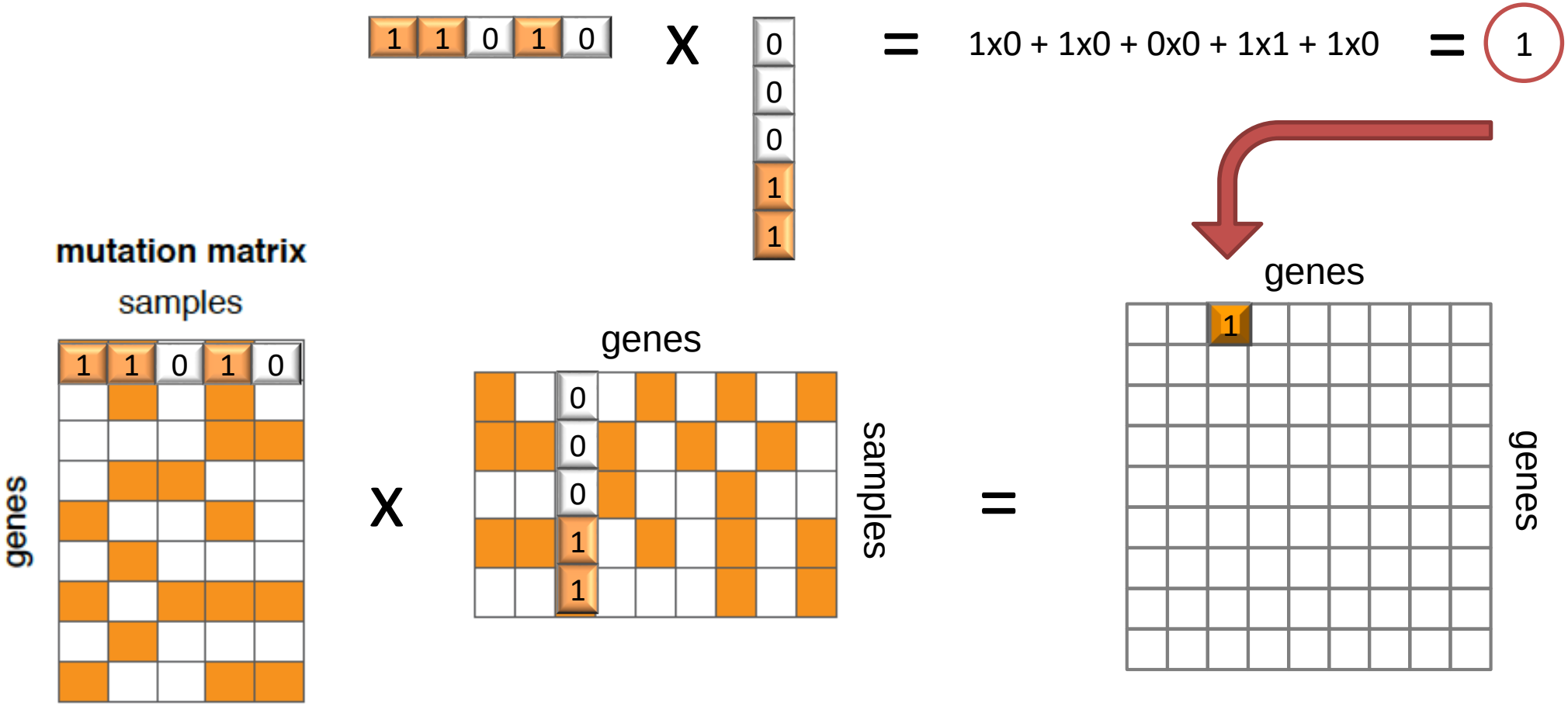
Matrix multiplication



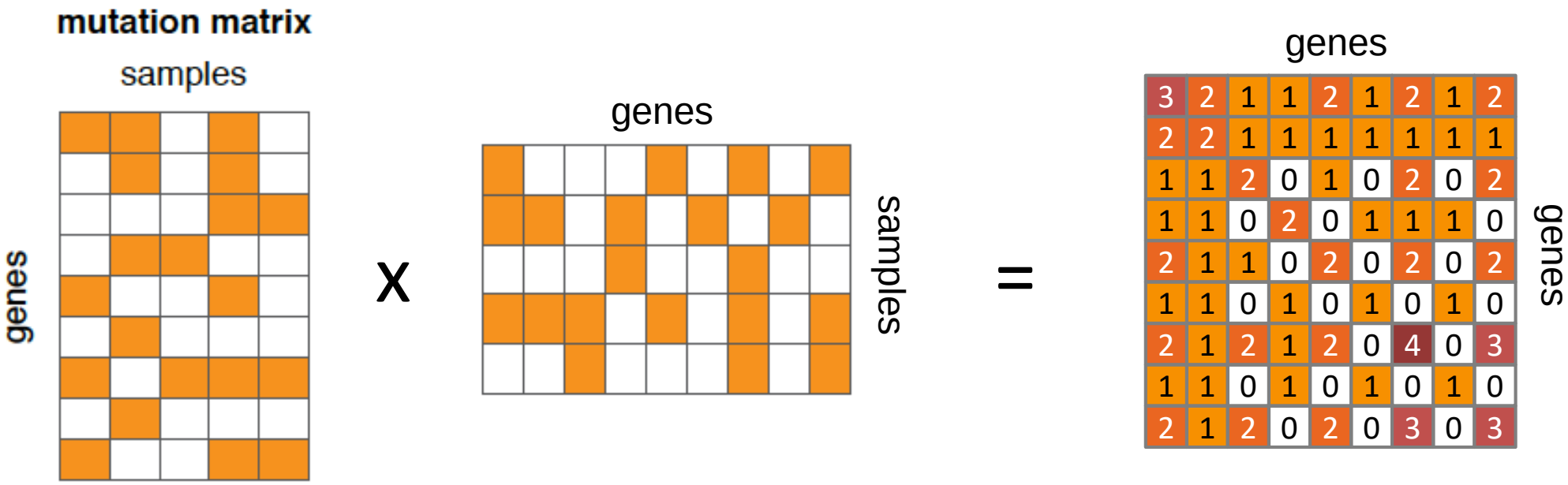
Matrix multiplication



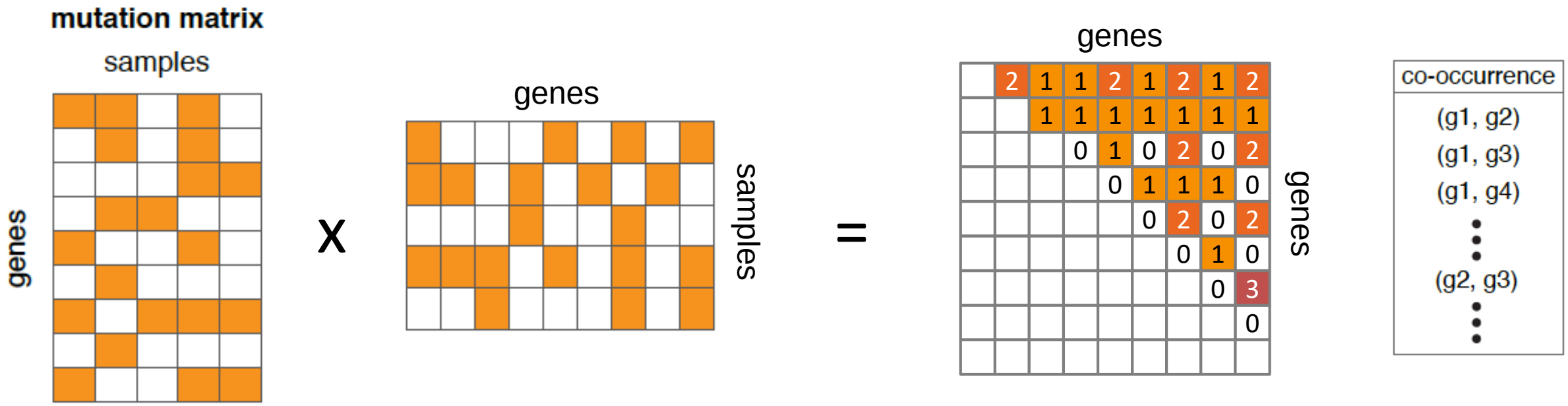
Matrix multiplication



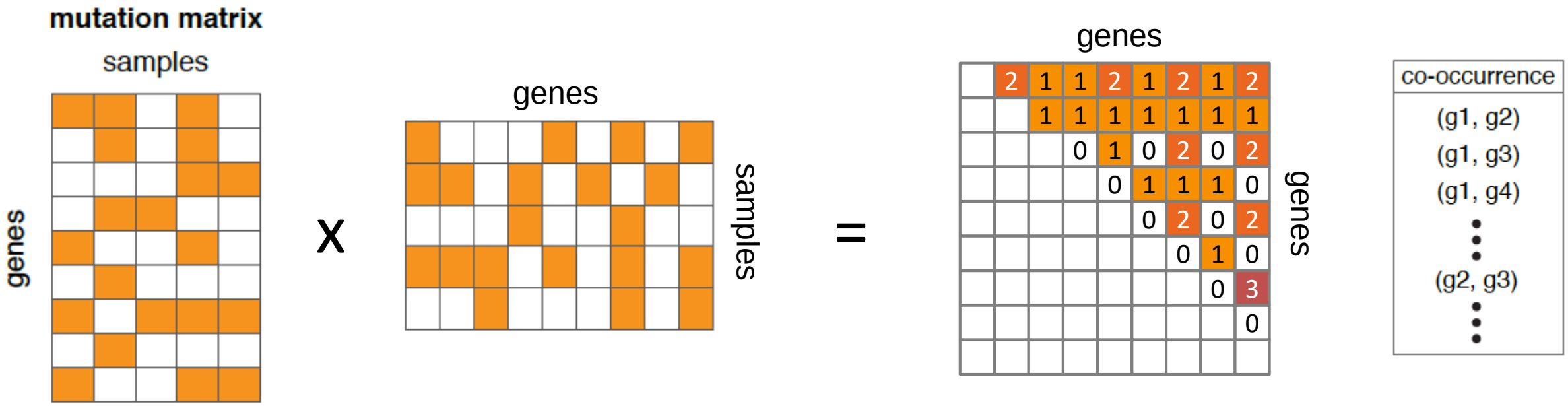
Matrix multiplication



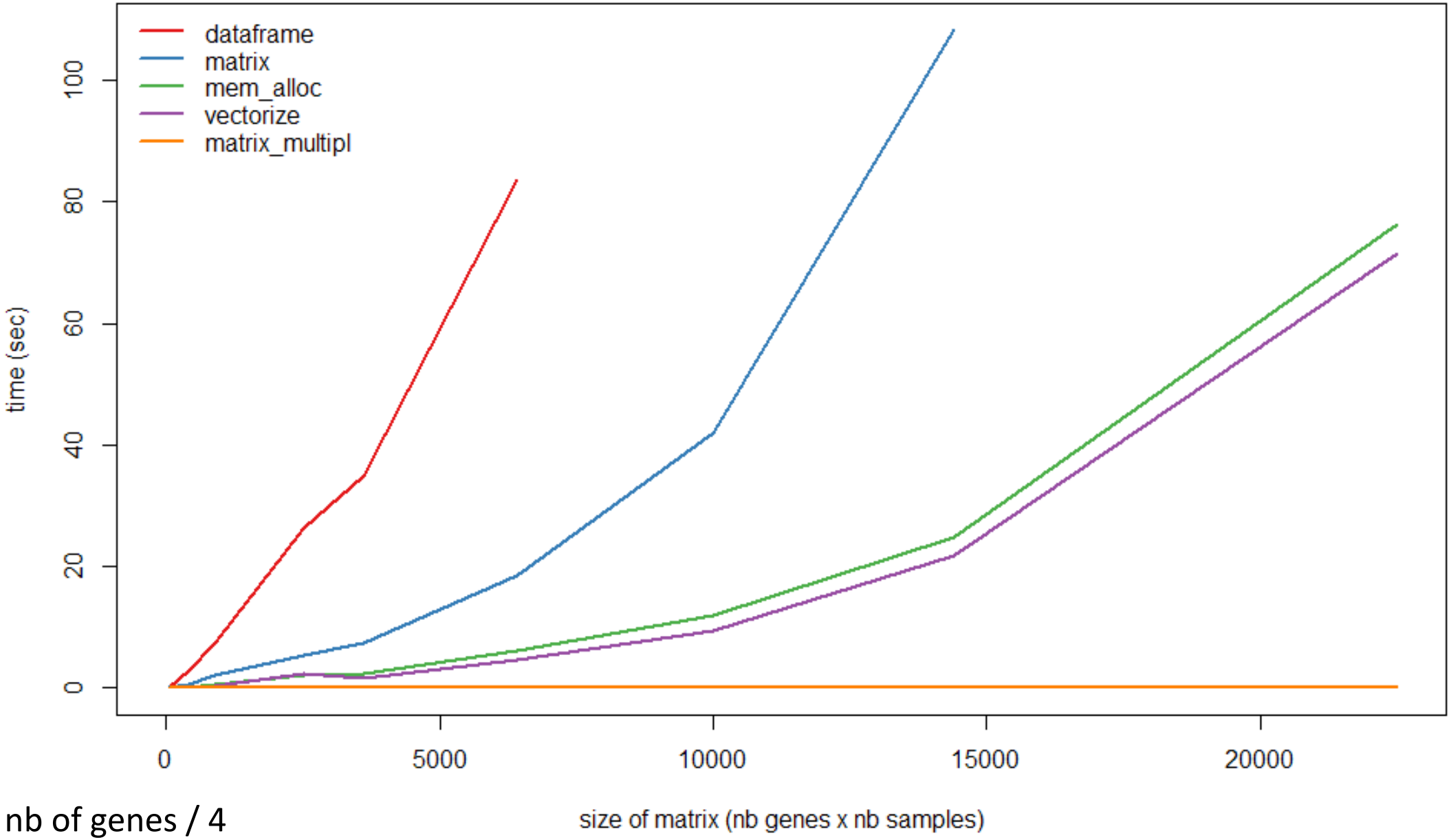
Matrix multiplication



Matrix multiplication

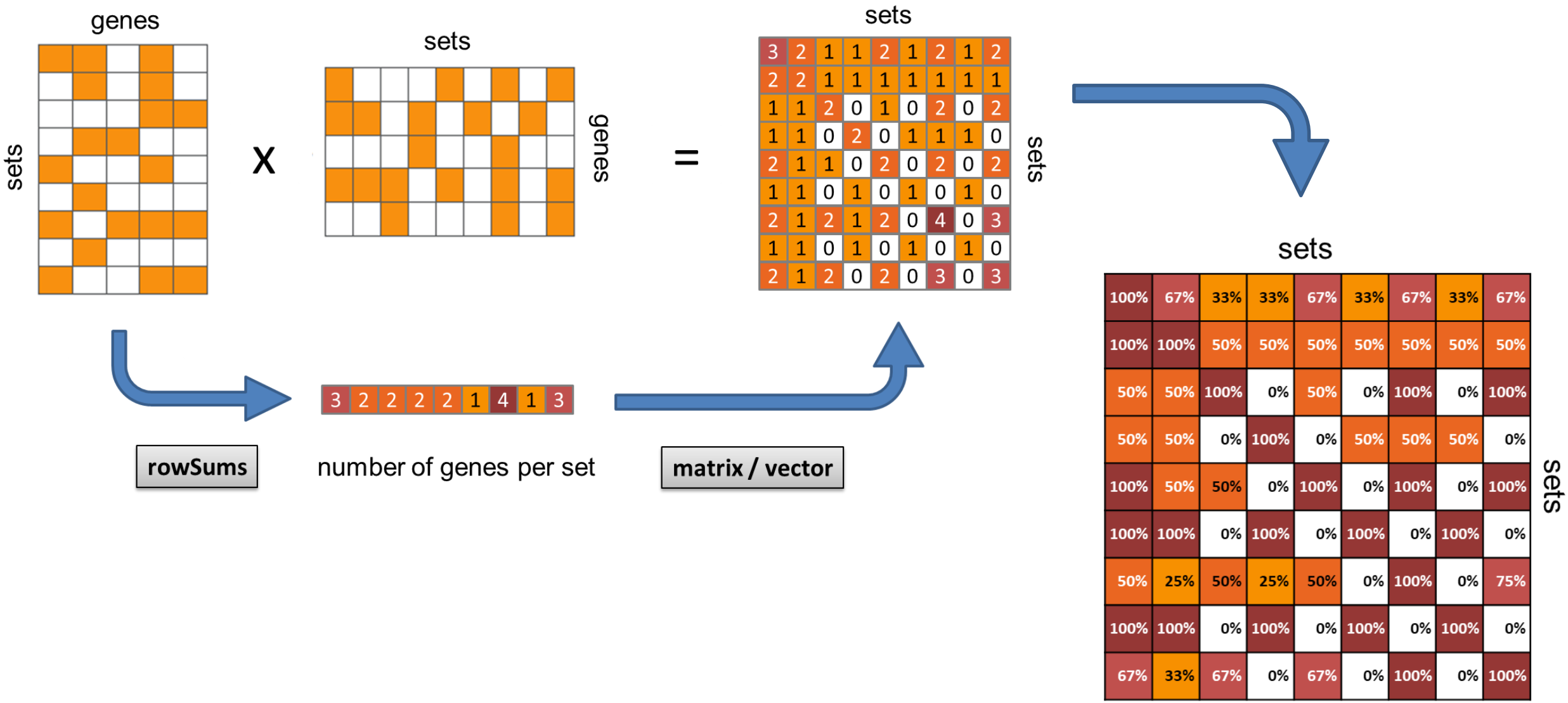


Benchmarking



nb of samples = nb of genes / 4

Similarity matrix



Example 2: which rows have sum > 4?

col1	col2	col3	col4
1.598	-0.325	1	9.647
1.110	2.491	5	1.712
1.298	-0.352	4	1.566
1.824	1.116	2	1.207
0.943	1.148	4	2.776
1.214	0.190	4	4.054
1.259	-2.696	2	0.046
0.214	0.050	1	0.621

output
greater
greater
greater
greater
greater
greater
less_or_equal
less_or_equal

from: www.r-bloggers.com/strategies-to-speedup-r-code

Example 2: which rows have sum > 4?

col1	col2	col3	col4
1.598	-0.325	1	9.647
1.110	2.491	5	1.712
1.298	-0.352	4	1.566
1.824	1.116	2	1.207
0.943	1.148	4	2.776
1.214	0.190	4	4.054
1.259	-2.696	2	0.046
0.214	0.050	1	0.621

```
for each row i
    if sum(col1_i,col2_i,col3_i,col4_i)>4
        output_i = "greater"
    else
        output_i = "less_or_equal"
```

from: www.r-bloggers.com/strategies-to-speedup-r-code

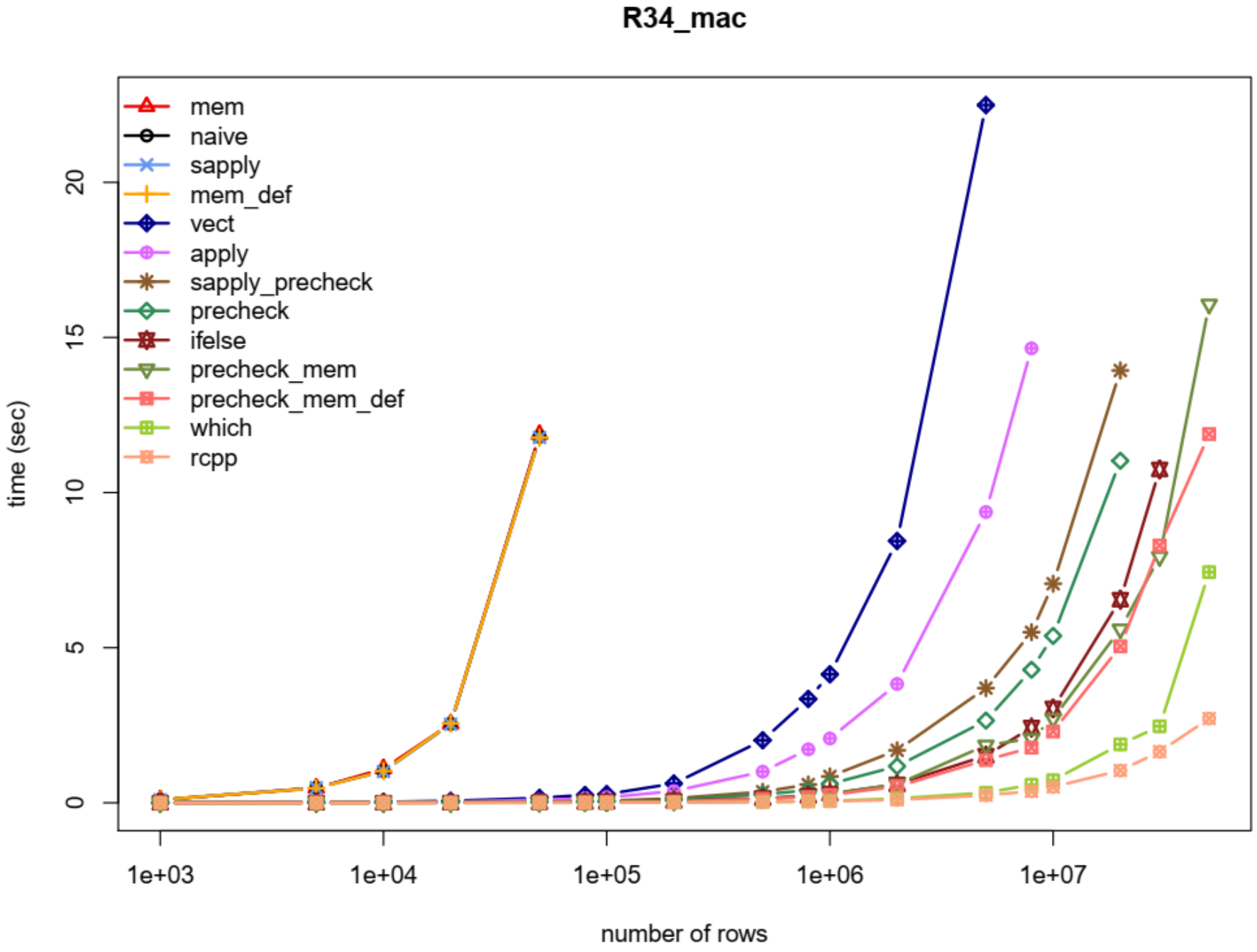
Example 2: which rows have sum > 4?

col1	col2	col3	col4
1.598	-0.325	1	9.647
1.110	2.491	5	1.712
1.298	-0.352	4	1.566
1.824	1.116	2	1.207
0.943	1.148	4	2.776
1.214	0.190	4	4.054
1.259	-2.696	2	0.046
0.214	0.050	1	0.621

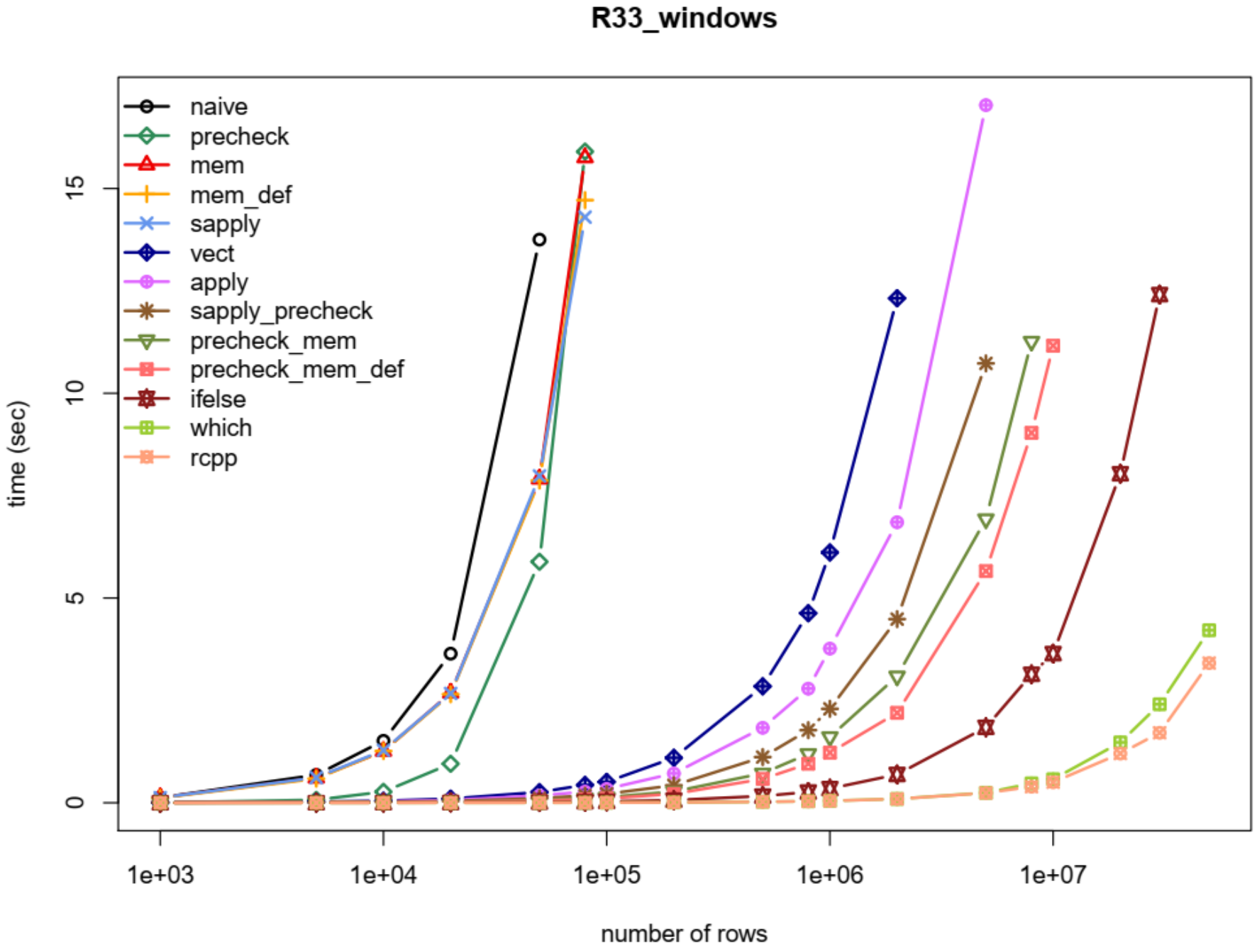
```
for each row i
    if sum(col1_i,col2_i,col3_i,col4_i)>4
        output_i = "greater"
    else
        output_i = "less_or_equal"
```

from: www.r-bloggers.com/strategies-to-speedup-r-code

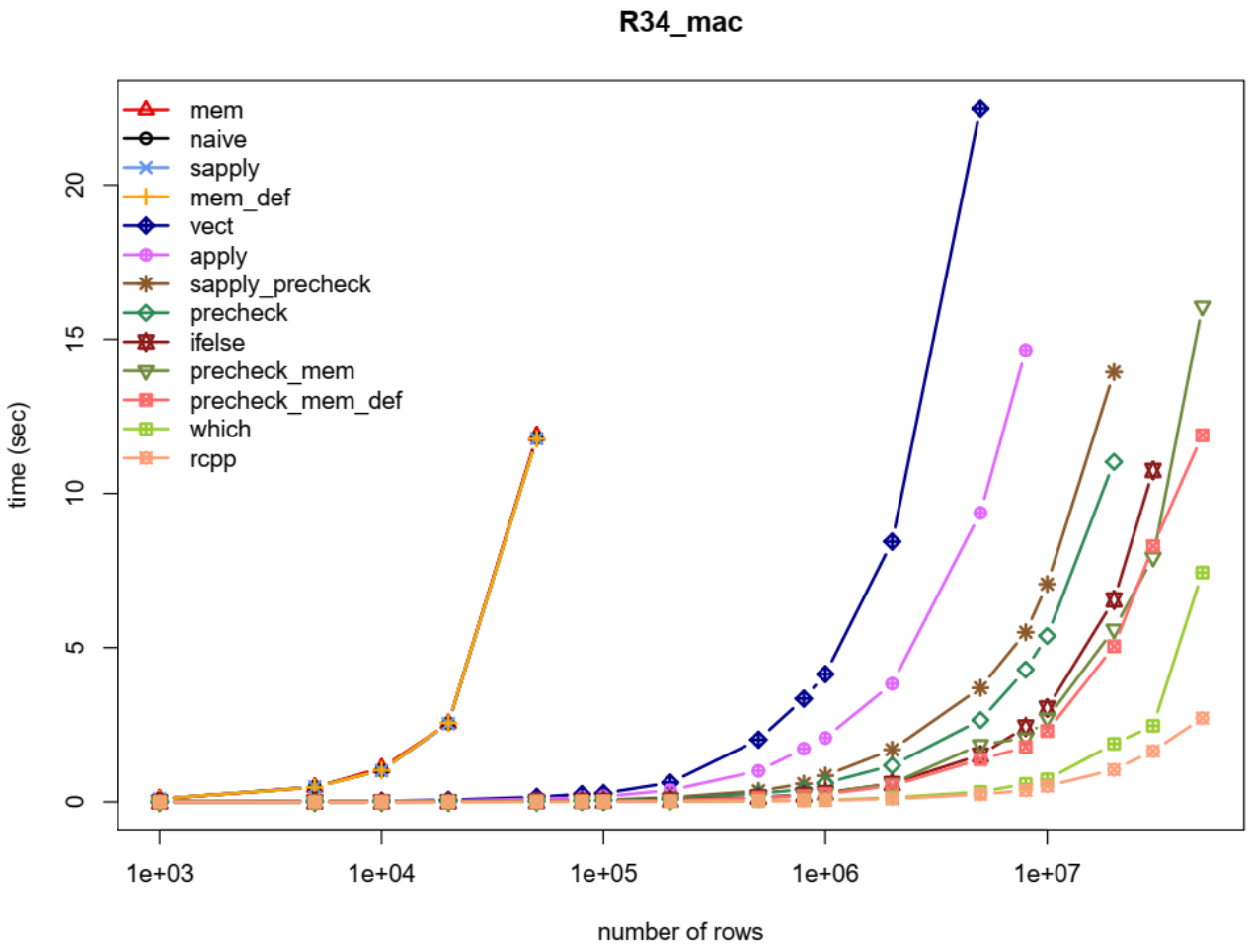
Benchmarking

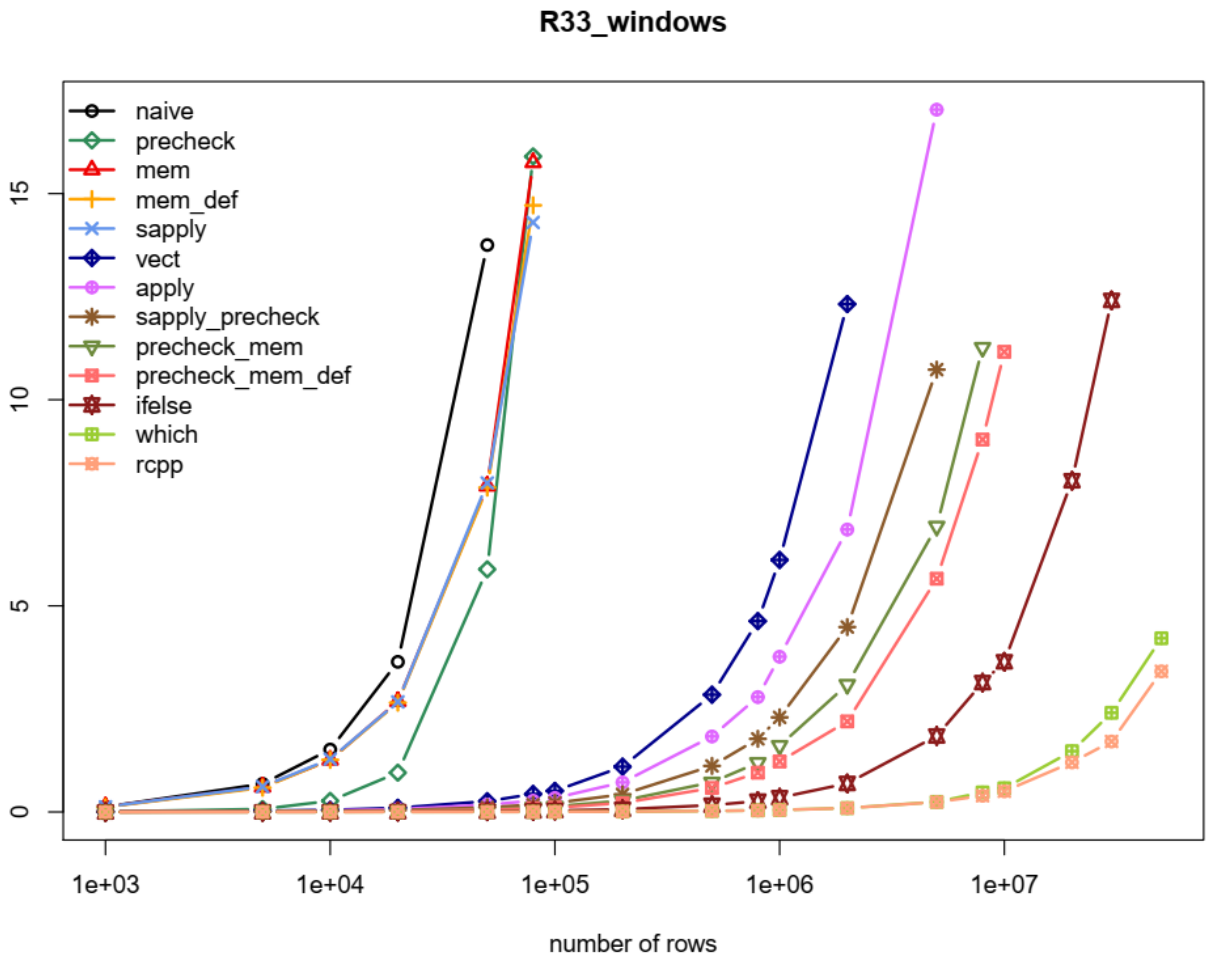


Benchmarking



Benchmarking





Example 3: aggregate values

group	col1	col2	col3	col4
1	1.598	-0.325	1	9.647
1	1.110	2.491	5	1.712
1	1.298	-0.352	4	1.566
2	1.824	1.116	2	1.207
2	0.943	1.148	4	2.776
3	1.214	0.190	4	4.054
3	1.259	-2.696	2	0.046
3	0.214	0.050	1	0.621

group	mean col1	max col4
1	1.335	2.491
2	1.384	1.116
3	0.896	0.190

Example 3: aggregate values

group	col1	col2	col3	col4
1	1.598	-0.325	1	9.647
1	1.110	2.491	5	1.712
1	1.298	-0.352	4	1.566
2	1.824	1.116	2	1.207
2	0.943	1.148	4	2.776
3	1.214	0.190	4	4.054
3	1.259	-2.696	2	0.046
3	0.214	0.050	1	0.621

group	mean col1	max col4
1	1.335	2.491
2	1.384	1.116
3	0.896	0.190